

Energy-Efficient Dataflow Design for Monolithic 3D Systolic Arrays with Resistive RAM

Prachi Shukla*, Mohammadamin Hajikhodaverdian* Vasilis F. Pavlidis†, Emre Salman‡, and Ayse K. Coskun*

* Boston University - (prachis, aminhaji, acoskun)@bu.edu

† University of Manchester - vasileios.pavlidis@manchester.ac.uk

‡ Stony Brook University - emre.salman@stonybrook.edu

Abstract—Systolic arrays are commonly used for running deep neural networks (DNNs) at the edge, where latency and energy efficiency requirements are stringent. Monolithic 3D (MONO3D) is an emerging 3D integration technology that offers ultra-high vertical interconnect density among processing and memory layers. The bandwidth benefits provided by MONO3D can help meet the growing latency and energy efficiency demands for DNNs. This paper presents a novel implementation for weight stationary (WS) dataflow in MONO3D systolic arrays, called WS-MONO3D. WS-MONO3D utilizes multiple resistive RAM layers and SRAM with high-density vertical interconnects to multicast inputs and performs high-bandwidth weight pre-loading while maintaining the same order of multiply-and-accumulate operations as in native WS dataflow. Consequently, WS-MONO3D eliminates input and weight forwarding cycles, and, thus, provides up to a 40% reduction in energy-delay-product (EDP) over the native WS implementation in 2D with iso-configuration. The paper also demonstrates the impact of temperature on energy efficiency benefits in WS-MONO3D.

Index Terms—Monolithic 3D, deep neural networks, systolic arrays, dataflow, energy efficiency, temperature.

I. INTRODUCTION

Deep neural networks (DNNs) serve a number of applications, such as health monitoring, security, image recognition, and others. Owing to expanding adoption of DNNs in edge and mobile devices, it is essential for DNN accelerators to achieve low latency and high energy efficiency [1], [2]. Among various accelerator architectures, systolic arrays are among the top choices for accelerating DNN inference at the edge (e.g., [3]). Systolic array (see Fig. 1 for a sample) is a 2D grid of processing elements (PEs) that perform multiply-and-accumulate (MAC) operations. Data flows from the input feature map (IFMAP) and filter SRAMs into the array, moving through the PEs. Each PE performs a MAC operation, stores the result, and passes it to the next PE. Finally, the PEs in the bottom row write the output feature map (OFMAP) back to SRAM. Systolic arrays are also characterized by a dataflow that defines how the IFMAP, filter weights, and OFMAP are mapped onto the systolic array to minimize data movement and maximize data reuse for energy efficiency. For example, weight stationary (WS) is a commonly adopted dataflow in systolic arrays, in which weights are first pre-loaded into the PE array [4], [5], followed by forwarding of the IFMAP to generate the OFMAP. The straightforward design and dataflow of a systolic array makes it a popular architecture for DNN acceleration in edge devices [4], [6].

While prior works have mainly focused on 2D accelerators for DNN acceleration [2], [7]–[10], 2D scaling is reaching its limits [11], [12]. Monolithic 3D (MONO3D) has emerged as a highly promising integration technology to mitigate interconnect-related challenges and provide a significant boost in performance and efficiency, while also reducing the chip footprint as compared to 2D ICs [12]. In MONO3D technology, multiple thin device layers are fabricated sequentially, separated by a thin inter-layer dielectric (ILD), and connected using ultra-thin vertical interconnects, called monolithic inter-tier vias (MIV), providing high integration density [13], [14]. MONO3D also supports high bandwidth integration with emerging memory technologies such as resistive RAM (RRAM), which is a high-density technology that can store DNN weights on-chip, eliminating off-chip memory access and achieving energy efficiency [15], [16]. This reduces latency and improves energy efficiency for various DNNs [17]–[19], including those at the edge, while enabling high-bandwidth and high-density edge devices [2].

This work presents a new WS implementation in MONO3D systolic arrays, **WS-MONO3D**, to reduce latency and improve energy efficiency. Prior works on MONO3D systolic arrays (e.g., [18], [19]) have not considered one or more of the following: (i) the high bandwidth available through MIVs, (ii) high-density RRAM to achieve energy-efficient DNN acceleration, or (iii) on-/off-chip data movement energy, which is a significant fraction of system energy. To the best of our knowledge, this is the first work to improve WS dataflow by utilizing MIVs for a high-bandwidth interface between monolithically stacked RRAM and systolic arrays to address

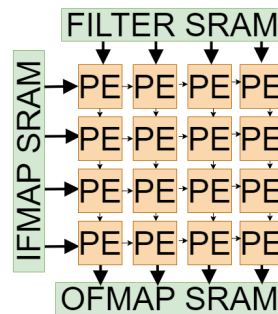


Fig. 1: A systolic array: 4×4 PE array with on-chip SRAMs.

latency and energy efficiency concerns. We utilize multiple layers of RRAM to store DNN weights on-chip and eliminate expensive off-chip DRAM accesses. We also demonstrate that our dataflow design reduces inference latency and improves energy efficiency even with fewer layers of RRAM. Our specific contributions are summarized as follows:

- We present WS-MONO3D, a new WS implementation for MONO3D systolic arrays that utilizes high-density MIVs to achieve latency and energy efficiency benefits over 2D. WS-MONO3D multicasts IFMAP and eliminates IFMAP forwarding. It also enables parallel pre-loading of weights into the PE array, thus, eliminating time-consuming weight forwarding cycles.
- Our design uses high-density and high-bandwidth MONO3D RRAM integration to store weights on the chip, reduce DRAM accesses for weights during DNN execution, and enable high bandwidth data transfer between RRAM and PE array using MIVs.
- We develop cross-layer architecture- and circuit-level models for three-, four-, and six-tier MONO3D systolic array configurations, each comprising a PE array, SRAMs for IFMAP and OFMAP, and RRAM layers for storing weights on the chip. We use these models to provide comparison across these designs as well as 2D counterparts.

Compared to a WS dataflow implemented on 2D systolic arrays, WS-MONO3D on a six-tier MONO3D system with four RRAM tiers provides up to 47% and 40% reduction in latency and energy-delay-product (EDP), respectively, for various DNNs for edge applications. The rest of the paper starts with a discussion of related work in Sec. II. We introduce WS-MONO3D in Sec. III and present its evaluation in Sec. IV. Finally, we conclude in Sec. V.

II. BACKGROUND AND RELATED WORK

This section provides background on WS dataflow and RRAM, followed by related work on MONO3D systolic arrays.

WS dataflow. WS is commonly used as an energy efficient data mapping for DNN acceleration in systolic arrays [20]. In WS¹ (or, native WS), weights are first pre-loaded into the PE array from a Filter SRAM through the top edge PEs, then passed to the PE below every cycle, as shown in Fig. 1 [21]. After weight pre-loading, IFMAPs are read from the left edge PEs and forwarded to PEs on the right every cycle. PEs generate partial sums (psums) and pass them to the PEs below. The PEs on the bottom row write outputs back to the OFMAP SRAM. Each column in the PE array computes an independent OFMAP channel. Often, there is an insufficient number of PEs to map the whole computation. In such cases, computation is sliced into folds (F) [21]. Consequently, the compute cycles in WS can be broken down as follows,

$$C_{WS} = \sum_i (w_i + I_i + O_i), \quad (1)$$

¹WS and native WS are used interchangeably in this paper.

where C_{WS} is the compute cycles in WS, $1 \leq i \leq F$ folds, w_i is the number of cycles spent in pre-loading weights, and I_i is the number of cycles to forward IFMAP from left to right until it reaches all of the PEs pre-loaded with weights. O_i includes total compute cycles when all the pre-loaded PEs generate psums (i.e., maximum throughput) and the total cycles spent forwarding psums from top to bottom. The primary benefit of WS is the reduction of data movement for weights, which are reused multiple times during inference. WS minimizes the bandwidth and energy consumption associated with transferring weights across the systolic array. This paper uses WS as a reference due to its benefits with respect to hardware utilization and energy efficiency [22]–[24]. We introduce a new MONO3D-specific WS that further reduces data movement, thereby decreasing energy consumption.

Resistive RAM. Emerging non-volatile memory technologies are potential replacements for conventional memories, such as SRAMs and DRAMs due to their high density, better scalability, and lower leakage power. RRAM is a high-density CMOS-compatible emerging non-volatile memory with low read latency/energy. Due to these characteristics, RRAMs are also getting popular in edge DNN accelerators for storing weights on-chip [2], [25]. However, RRAM has write endurance issues, mostly related to multi-level cells, which are not considered in this work [26]. Also note that in this work, RRAM is used as a read-only memory. Furthermore, RRAM can be fabricated with MONO3D technology [25].

MONO3D DNN Accelerators. Existing research in MONO3D DNN accelerators focuses on aspects, such as weight/activation sparsity, SRAM partition, process variability, compute-in-memory, or temperature-aware chiplet sizing for multi-DNN workloads [18], [27], [28]. However, none of the prior efforts has exploited the ultra-high bandwidth in MONO3D technology to improve dataflows for high efficiency. The closest work by Joseph et al. [19] distributes output stationary (OS) dataflow in 3D systolic arrays. Their area, energy, and performance models include only the PE array without considering the impact due to on-chip SRAMs, DRAMs, or interconnects, thus, making the evaluation incomplete. In this work, we optimize WS for MONO3D systolic arrays and evaluate its benefits over 2D using a cross-layer architecture and circuit-level models. Our proposed WS-MONO3D uses MIVs in MONO3D to reduce the cycles for forwarding IFMAP and weights to the systolic array during inference, resulting in lower latency and energy consumption compared to native WS in 2D.

III. WS-MONO3D

Our goal in this work is to design a new WS-based data mapping scheme for MONO3D, leveraging high-bandwidth MIVs to reduce data movement during DNN inference. In this section, we first provide an overview of our proposed architecture-level design decisions and describe improvements in WS-MONO3D. Then, we present the WS-MONO3D chip stack in detail. Finally, we detail our cross-layer architecture-

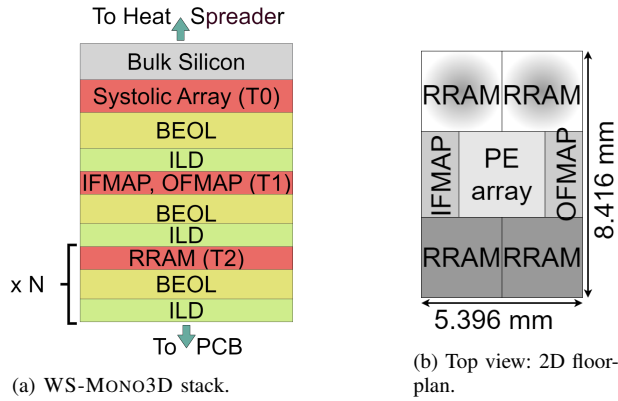


Fig. 2: (a) A flip-chip MONO3D chip stack with N RRAM tiers for storing weights. Each tier is $2.816 \times 2.816 \text{ mm}^2$, (b) Top view of (a)’s 2D counterpart, where the top two RRAMs can be omitted when we reduce the size.

and circuit-level models for performance, power, temperature, and area to evaluate WS-MONO3D.

A. Architecture-level Design Decisions

We make three main architectural design decisions to utilize MIVs and improve the spatio-temporal WS characteristics in MONO3D: (i) A_1 : MONO3D integration of SRAMs and RRAMs for high-bandwidth interface with the PE array; (ii) A_2 : in every WS fold, reduce the number of weight preload cycles to one by reading all the weights into the PE array through MIVs; (iii) A_3 : in every WS fold, reduce the IFMAP forwarding cycles to 1 cycle by multicasting the IFMAPs to all the PEs in their respective rows through MIVs. As a result of these decisions, Eq. (1) reduces to $C_{WS-Mono3D} = \sum_{i=1}^F (1 + 1 + O_i)$. Note that, while A_3 utilizes MIVs in WS-MONO3D for high-bandwidth with negligible area overhead, a similar high-bandwidth interface in a 2D design would need extra resources to support multicast and distribute input data across the systolic arrays. The extra resources will lead to an increase in footprint, further increasing latency and data movement overhead and, thus, is not studied in this work.

B. WS-MONO3D Chip Stack

Fig. 2a, illustrates the flip-chip MONO3D stack with a systolic array in tier 0 (T_0), IFMAP and OFMAP SRAMs in tier 1 (T_1) and N tiers of RRAM ($T_2, T_3, \dots, T(N+1)$), each with its own back-end-of-line (BEOL). T_0 with the systolic array is closest to the heat spreader because it has highest power consumption (P_c) among all tiers. In this work, we investigate three different configurations based on the chip stack to demonstrate that WS-MONO3D works well on different configurations with various tiers. To demonstrate the benefits of WS-MONO3D, we choose a fixed-size 256×256 systolic array with 2 MB IFMAP SRAM, 2MB OFMAP SRAM with varying RRAM sizes: 32 MB for six-tier ($N=4$), 16 MB for four-tier ($N=2$), and 8 MB for three-tier ($N=1$) configurations. Our goal is to demonstrate that WS-MONO3D

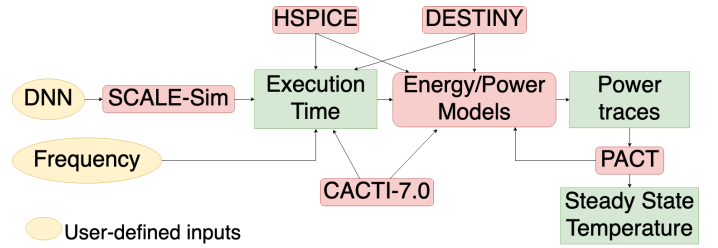


Fig. 3: Evaluation framework for WS-MONO3D

outperforms native WS in 2D in terms of efficiency and performance even with a few tiers, so we fix the size of the systolic array and SRAMs and only vary the MONO3D RRAM tiers. However, these configurations can be changed based on user specifications.

Additionally, using four layers of RRAMs (32 MB) to store DNN weights on the chip in our six-tier configuration eliminates the need for off-chip data access to DRAMs, reducing DRAM leakage power and energy consumption. We select these configurations with the objective to (i) have sufficient on-chip memory capacity to reduce and eliminate (when possible) off-chip DRAM accesses during DNN execution, (ii) minimize RRAM endurance concerns by including sufficient capacity to store all weights, when possible, without overwriting any cell during DNN execution, and (iii) minimize area mismatch between tiers (including MIV overhead). While MONO3D integration is an emerging technology with potential challenges in manufacturability, where recent works have made considerable progress in addressing these issues [29]. For the purpose of our work, we assume that MONO3D manufacturability is not a limiting factor. Furthermore, recent works have also explored multiple-layer MONO3D architectures in designing caches and DNN accelerators, demonstrating their potential [19], [30].

C. Architecture- and Circuit-level Cross-layer Models

Fig. 3 shows our cross-layer modeling framework to evaluate WS-MONO3D. Our framework integrates several modeling tools to estimate a configuration’s performance, power consumption, and thermal characteristics. We utilize architecture-level models (SCALE-Sim [21], CACTI-7.0 [31], and DESTINY [32]) to evaluate the area, performance, and power for DNN inference on MONO3D systolic arrays with WS-MONO3D dataflow. Circuit-level models (via HSPICE) help us estimate delay and power for MIVs and inter-PE communication. Additionally, we estimate steady-state temperatures using a compact thermal simulator (PACT [33]), accounting for the interaction between temperature and leakage power.

With this approach, we next provide a detailed modeling methodology for WS-MONO3D. For performance evaluation, we model WS-MONO3D in SCALE-Sim. SCALE-Sim is a CNN simulator for systolic arrays that models a stall-free inference. It models double-buffered on-chip memory to hide the DRAM cycles during DNN execution. We generate per-fold counters to determine the weight preloading cycles (w_i) and IFMAP forwarding cycles (I_i), and then calculate $C_{WS-Mono3D}$ for each DNN. For every convolutional ($Conv$)

layer, in addition, to compute cycles, SCALE-Sim outputs non-overlapping DRAM cycles that contribute towards total execution cycles. Since we model sufficient on-chip SRAMs to store the inputs and outputs during a *Conv* layer execution, we only add the non-overlapping DRAM cycles of the first *Conv* layer for reading inputs and the last *Conv* layer for writing outputs to calculate the total execution cycles. For the two *Conv* layers, we add additional cycles due to RRAM and SRAM read/write latencies plus routing delay to reach the on-chip memory tiers from the bottom tier $T(N + 1)$ in Fig. 2a. To calculate the routing delay, we first estimate the lateral and vertical distances using Manhattan Distance (MD), a commonly used approach [34], and then calculate the delay using HSPICE. Finally, we calculate the total DNN execution time using a user-defined frequency. We use SCALE-Sim’s default DRAM bandwidth of 10 B per cycle.

We use DESTINY and CACTI-7.0 to model RRAM and SRAMs, respectively, and generate their area, latency, and power consumption (P_c). Each SRAM is 2 MB with 16 B word length and 16 banks. We determine RRAM dynamic read/write energy and leakage using DESTINY. Since DESTINY models only a single bank, we assume each RRAM tier comprises 64 banks, each of 128 KB capacity and 256 B word length. Each SRAM/RRAM bank can be accessed in parallel, and has one read and one write port, each with dedicated MIVs, with one MIV per bit. Having 1 MIV/bit is reasonable because MIVs are on a nanometer scale and incur minimal area overhead. For instance, T_5 in Fig. 2a has 64 RRAM banks. Since each bank is 128 KB with 256 B word length, we insert $2 \times 2,057$ MIVs per bank (9 MIVs for address and 2048 MIVs for data, each for both read and write ports). Thus, there are 263,296 MIVs going from T_5 to T_4 . Per the MIV dimensions used in this work (i.e., 50 nm radius) [35], each MIV area is $7.85e-9 \text{ mm}^2$, and thus, T_5 incurs a via overhead of $2e-3 \text{ mm}^2$. For interconnect power modeling, inter-RRAM routing P_c is assigned to the tier’s BEOL. In CACTI/DESTINY, by default, the address/data bus of each SRAM/RRAM bank is on the edge due to the peripheral logic commonly placed along edges in 2D designs. For RRAM tiers in WS-MONO3D, due to the high RRAM bandwidth enabled by ultra-dense MIVs, we model the data bus arriving at the center of an RRAM bank instead to eliminate H-tree horizontal routing, a routing mechanism popularly used in 2D for routing data between the I/O port and the center of the bank [32]. This leads to a reduction in energy and read latency. Furthermore, since RRAMs have dedicated tiers in WS-MONO3D, we model all lateral RRAM routing in their corresponding BEOLs, a feature already provided by DESTINY. These RRAM architecture decisions eliminate the area overhead due to the high-bandwidth MIV interface and reduce RRAM read latency. To model these RRAM architectural changes, we update DESTINY’s RRAM model by setting the edge-to-center delay and power to zero for the data bus.

To estimate the routing P_c between RRAMs/SRAMs and the systolic array, we use MD to calculate vertical distance through the MIVs and lateral distance to reach PEs in the

systolic array tier, followed by HSPICE simulations. The routing P_c due to lateral wirelengths is added to the systolic array BEOL, while the MIV power is added to SRAM/RRAM tiers’ BEOL. In addition, we use HSPICE and array utilization to calculate the inter-MAC IFMAP, weight, and OFMAP forwarding P_c . We add the IFMAP, weight, and OFMAP forwarding P_c to the PE array power in WS in 2D. In contrast, since our proposed dataflow design eliminates IFMAP and weight forwarding P_c , we add only OFMAP forwarding P_c in WS-MONO3D. Note that physical design is out of our scope although there is ongoing research on MONO3D PDKs. Finally, for steady-state temperature estimation, we build a thermal model for our MONO3D system in PACT. For accurately determining leakage, we run DESTINY/CACTI iteratively with PACT-generated temperatures until convergence, i.e., the temperature difference between consecutive runs $< 1^\circ\text{C}$. In our analysis at 22 nm, a change of 1°C has a negligible impact on leakage. Thus, a smaller convergence criterion may be chosen but will not impact the thermal and power estimation results and instead result in longer simulations.

IV. EVALUATION

This section describes the experimental setup in WS-MONO3D and discusses its benefits with respect to 2D WS.

A. Experimental Setup

We perform our analysis at 22 nm CMOS technology node to demonstrate WS-MONO3D benefits. We use a representative MAC unit with area, energy, and frequency determined from a recent work [18]: $121 \mu\text{m}^2$, 0.26 pJ per 8-bit integer MAC operation, 1 GHz. Tier 0 is 500 nm thick, while the height of upper tiers is determined by gate pitch, i.e., $8 \times \frac{\text{technology node}}{2} \approx 85 \text{ nm}$ [36]. The length of a MIV is 270 nm since it passes through the ILD (100 nm), upper tier (85 nm), and the dielectric between the tier and metal layer (85 nm). We also use representative values for MIV’s diameter, pitch, area, resistance, and capacitance [37]. Using HSPICE, we obtain (i) MIV delay and energy values of 8.6 ps and 0.02 fJ, (ii) inter-MAC delay and energy values of 14 ps and 0.08 fJ, which are the delay and energy between neighboring PEs, respectively.

To evaluate WS-MONO3D benefits, we choose three configurations, as mentioned in the previous section: (i) a six-tier configuration with 32 MB RRAM, (ii) four-tier configuration with 16 MB RRAM, and (iii) three-tier configuration with 8 MB RRAM. Each configuration has a 256×256 systolic array with 2 MB IFMAP SRAM, 2 MB OFMAP SRAM, comprising the first two tiers. We target six high-accuracy DNNs commonly deployed at the edge: ResNet-18, ResNet-32, ResNet-50, MobiLeNet-V2, EfficientNet-B0, and GoogLeNet. Since the topologies of these DNNs vary from one another, their execution leads to varying performance, power, and thermal profiles. We evaluate WS-MONO3D for DNN inference, using the six DNNs at three frequency levels: 500 MHz, 700 MHz, and 1 GHz. While 1 GHz is from a representative recent work on mobile systolic arrays [2], the other two frequency levels

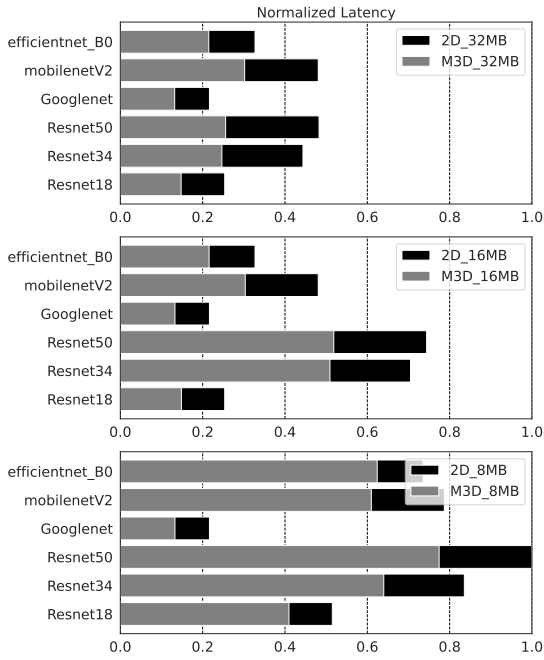


Fig. 4: Normalized latency (ms) comparison.

are chosen to demonstrate MONO3D impact on inference per second per Watt per mm^2 and temperature at different frequency levels, mimicking dynamic frequency scaling mechanisms that are commonplace in modern mobile products. We use a batch size of 1 for inference [38]. To compare WS-MONO3D to native WS, we model a 2D 256×256 systolic array with iso-capacity SRAM and RRAM that implements WS dataflow. A top view of the 2D floorplan is shown in Fig. 2b. All forwarding energies and power within the PE array are added to the 2D WS setup. Chip area (sum of area of all tiers) for 6-, 4-, and 3-tier configurations are 47.58 mm^2 , 31.72 mm^2 , and 23.79 mm^2 , respectively, while the chip footprint is $2.816 \times 2.816 \text{ mm}^2$ for all three. The equivalent 2D setups at iso-configuration with 6-, 4-, and 3-tier configurations have an area of $8.416 \times 5.396 \text{ mm}^2$, $5.616 \times 5.396 \text{ mm}^2$, and $4.216 \times 5.396 \text{ mm}^2$, respectively. In 2D, footprint and area are the same. To model the absence of heat sinks on edge devices, we reduce heat sink thickness to 1 nm. The heat spreader thickness is set to 1 mm, and 45°C is the ambient temperature [18], [39]. We also use two thermal budgets, 75°C and 85°C , to evaluate the thermal effects on WS-MONO3D. To model a low-cooling capability, we use a poor convection resistance ($1.3 \text{ W}/^\circ\text{C}$) [40].

B. Results

We compare WS-MONO3D to 2D WS for three different RRAM sizes at iso-frequency to evaluate inference latency and energy efficiency benefits. We also demonstrate that thermal awareness plays an important role in the design of systolic arrays implementing WS-MONO3D. Fig. 4 shows the latencies normalized to the maximum latency observed among all DNNs in this experiment, i.e., *ResNet-50* executing on the 2D WS configuration with 8MB RRAM. WS-MONO3D achieves

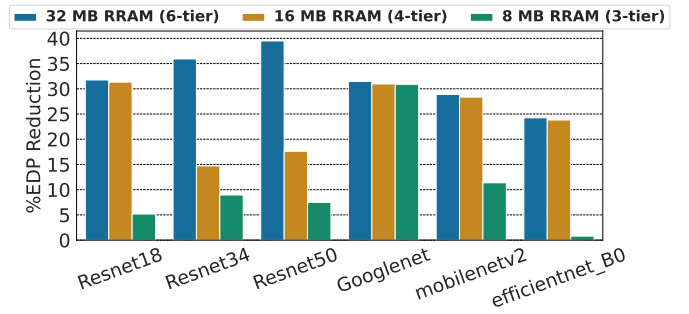


Fig. 5: EDP benefits in WS-MONO3D w.r.t. 2D WS at 1 GHz.

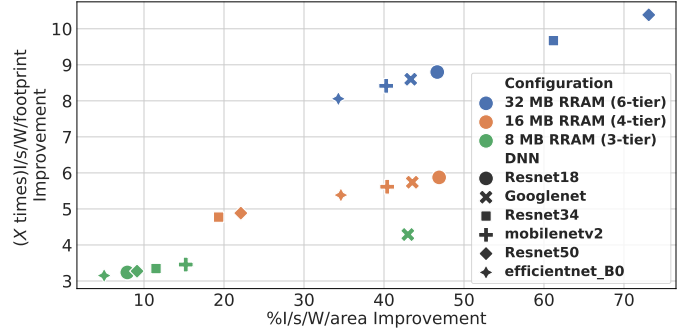
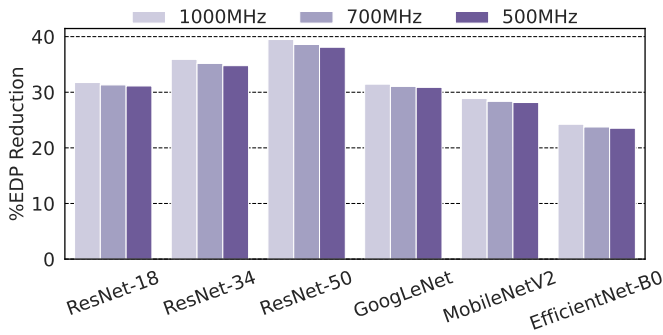
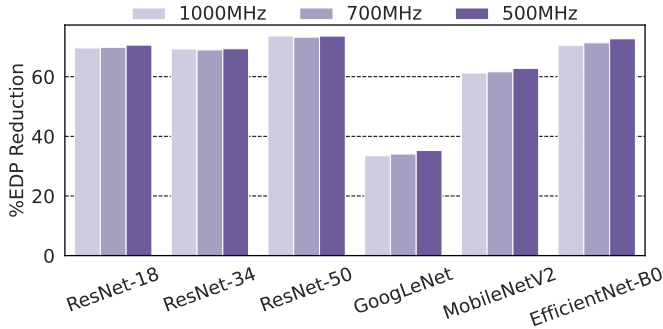


Fig. 6: WS-MONO3D $I/s/W/area$ and $I/s/W/footprint$ comparison w.r.t. 2D WS.

a latency reduction of up to 47% (avg. 41%) for 6-tier, 30% (avg. 20.1%) for 4-tier, and 22% (avg. 14.98%) for 3-tier due to reduction in compute cycles from IFMAP multicast and parallel weight preloading. Furthermore, the figure shows that WS-MONO3D outperforms 2D WS in all three configurations across all DNNs. Fig. 5 compares EDP improvements of WS-MONO3D configurations to a 2D configuration of equivalent RRAM size (32 MB for comparing with 6-tiers, 16 MB for 6-tiers, and 8 MB for comparing with 3-tiers configurations), operating at a frequency of 1 GHz with native WS. We also include DRAM energy in EDP. We observe that WS-MONO3D results in up to 12% (avg. 9%) higher chip power than 2D WS. This is primarily because more RRAM banks are active for the parallel preloading of weights using MIVs. However, across all configurations, WS-MONO3D exhibits enhanced efficiency in terms of EDP compared to 2D, with a maximum and average reduction, respectively, of 40% and 32% for 6-tier, 30% and 23% for 4-tier, and 30% and 11% for 3-tier. The reason that 6-tier outperforms 3-tier and 4-tier with respect to latency and EDP is that it has sufficient RRAM capacity that eliminates off-chip DRAM accesses for all DNNs. The similar benefits observed in different WS-MONO3D configurations for the same DNN are due to having sufficient RRAM to store all weights. Fig. 6 shows inference per second per Watt per area ($I/s/W/area$) improvement over native WS in 2D for the three configurations at 1 GHz (see Sec. IV-A for area values). WS-MONO3D achieves up to 74% improvement for the 6-tier configuration. The improvements are primarily due to the latency benefits in WS-MONO3D



(a) EDP reduction in WS-MONO3D 6-tier w.r.t. 2D WS with 32 MB RRAM.



(b) EDP reduction in WS-MONO3D 6-tier w.r.t. 2D WS with 16 MB SRAM.

Fig. 7: WS-MONO3D 6-tier EDP comparison w.r.t. 2D WS with (a) only RRAM, and (b) only SRAM.

since MONO3D and 2D areas are similar at iso-configuration. In addition, since footprint efficiency is critical due to limited package area, we also measure $I/s/W/footprint$ in Fig. 6. With respect to 2D WS, we see a significant improvement up to $10\times$ (avg.: $9\times$) for the 6-tier configuration, out of which the footprint savings alone is $\approx 6\times$ due to vertical integration. Building on the previous discussion, we focus on our six-tier configuration, which has yielded the most significant benefits in WS-MONO3D. Fig. 7a demonstrates the EDP benefits of WS-MONO3D at different frequencies. Additionally, we compare 6-tier WS-MONO3D configuration to a design with the same area as our 2D configuration but using SRAM instead of RRAM for weights, which reduces the weight storage to 16 MB. Fig. 7b shows the EDP improvement with respect to WS in 2D design with only SRAM, which achieves up to 60% EDP reduction. Comparing these results in Fig. 7, we observe that our proposed method outperforms conventional 2D architectures using SRAM or RRAM. Interestingly, WS-MONO3D EDP benefits are greatest in *ResNet-50* executing on the six-tier configuration. This is due to two reasons. First, WS-MONO3D provides more significant benefits in *Conv* layers than fully-connected (FC) layers. FC layers are matrix-vector multiplication, where only the first row in a systolic array is utilized. Consequently, WS-MONO3D provides improvement only due to the multicasting of the inputs. In contrast, *Conv* layers are matrix-matrix multiplication and can benefit from both multicasting and parallel pre-loading of weights. Second, WS-MONO3D benefits increase with a greater number of

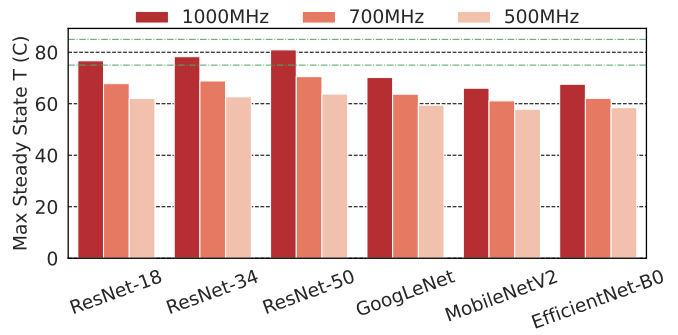


Fig. 8: WS-MONO3D 6-tier steady-state temperatures.

DNN channels. A greater number of channels means more cycles are spent in left-to-right input forwarding in 2D WS and, hence, more benefits can be achieved from input multicasting in WS-MONO3D. Since, out of all the DNNs investigated in this paper, *ResNet-50* has the maximum number of *Conv* layers (i.e., 48) with the number of channels ranging from 64 to 2048, WS-MONO3D benefits are the highest.

We also obtain steady state temperatures, as in Fig. 8 for 6-tier configuration, and evaluate WS-MONO3D at various thermal constraints. A relaxed constraint of 85°C allows DNN execution at all three frequencies. However, under tighter constraints, e.g., 75°C , the average latency and EDP benefits reduce to 29% and 18%, respectively. This is because while *ResNet-18*, *34*, and *50* execute at 1000 MHz in the 2D systolic array with WS dataflow, the strict thermal budget allows 700 MHz (not 1000 MHz) in WS-MONO3D to avoid thermal violations. Thus, temperature impacts WS-MONO3D benefits. Furthermore, by reducing the number of layers, we can reduce the temperature of the design. We observe a temperature decrease of 1°C when transitioning from a 6-tier to a 4-tier configuration and up to 3°C when moving from a 6-tier to a 3-tier configuration. For instance, in the case of *ResNet-50*, at a frequency of 1000 MHz in the 6-tier configuration, the architecture operates at 80.92°C . In contrast, this temperature drops to 77.96°C in the 3-tier configuration.

V. CONCLUSIONS

This paper presented WS-MONO3D, a novel implementation of WS dataflow in MONO3D systolic arrays to improve energy efficiency of DNNs at the edge. WS-MONO3D utilizes the ultra-high MIV bandwidth in MONO3D technology and eliminates cycles spent in pre-loading weights and forwarding IFMAPs to minimize latency and improve energy efficiency. We investigated three different configurations based on MONO3D chip stack that reduce and eliminate (e.g., in 6-tier design for all DNNs) the off-chip DRAM accesses for re-fetching weights, which provides energy efficiency benefits. Compared to WS in 2D, WS-MONO3D provides up to 47% reduced latency and 40% lower EDP at a relaxed temperature constraint of 85°C for the 6-tier configuration. WS-MONO3D also achieves up to 74% improvement in $I/s/W/area$ and $10\times$ improvement in $I/s/W/footprint$ over WS in 2D.

REFERENCES

- [1] T. Chen *et al.*, “DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning,” *ACM SIGARCH Computer Architecture News*, vol. 42, no. 1, pp. 269–284, 2014.
- [2] H. Li, M. Bhargav, P. N. Whatmough, and H.-S. Philip Wong, “On-chip memory technology design space explorations for mobile deep neural network accelerators,” in *ACM/IEEE DAC*, 2019, pp. 1–6.
- [3] E. TPU, “Edge tensor processing units (tpu) performance benchmarks,” *EdgeTPU*, 2022. [Online]. Available: <https://coral.ai/docs/edgetpu/benchmarks/>
- [4] N. P. Jouppi *et al.*, “In-datacenter performance analysis of a tensor processing unit,” in *ACM/IEEE Proc. of ISCA*, 2017, pp. 1–12.
- [5] H. T. Kung, B. McDanel, S. Q. Zhang, C. T. Wang, J. Cai, C. Y. Chen, V. C. Y. Chang, M. F. Chen, J. Y. C. Sun, and D. Yu, “Systolic building block for logic-on-logic 3d-ic implementations of convolutional neural networks,” in *IEEE ISCAS*. IEEE, 2019, pp. 1–5.
- [6] H.-T. Kung, “Why systolic architectures?” *Computer*, no. 1, pp. 37–46, 1982.
- [7] B. Asgari, R. Hadidi, H. Kim, and S. Yalamanchili, “Lodestar: Creating locally-dense CNNs for efficient inference on systolic arrays,” in *IEEE/ACM Design Automation Conference (DAC’19)*, 2019, pp. 1–2.
- [8] M. Zhu, T. Zhang, Z. Gu, and Y. Xie, “Sparse tensor core: Algorithm and hardware co-design for vector-wise sparse neural networks on modern gpus,” in *IEEE/ACM Proc. of International Symposium on Microarchitecture*, 2019, pp. 359–371.
- [9] H. T. Kung, B. McDanel, S. Q. Zhang, X. Dong, and C. C. Chen, “Maestro: A memory-on-logic architecture for coordinated parallel use of many systolic arrays,” in *IEEE ASAP*, vol. 2160, 2019, pp. 42–50.
- [10] Z.-G. Liu, P. N. Whatmough, and M. Mattina, “Systolic Tensor Array: An efficient structured-sparse GEMM accelerator for mobile CNN inference,” *IEEE CAL*, vol. 19, no. 1, pp. 34–37, 2020.
- [11] S. Naffziger *et al.*, “Pioneering chiplet technology and design for the AMD EPYC™ and Ryzen™ processor families: Industrial product,” in *ACM/IEEE ISCA*, 2021, pp. 57–70.
- [12] V. F. Pavlidis, I. Savidis, and E. G. Friedman, *Three-dimensional integrated circuit design*. Newnes, 2017.
- [13] M. Vinet *et al.*, “Monolithic 3d integration: A powerful alternative to classical 2d scaling,” in *IEEE S3S*, 2014, pp. 1–3.
- [14] L. Brunet *et al.*, “First demonstration of a CMOS over CMOS 3D VLSI CoolCube™ integration on 300mm wafers,” in *IEEE Symposium on VLSI Tech.*, 2016, pp. 1–2.
- [15] K. Dhananjay, P. Shukla, V. F. Pavlidis, A. Coskun, and E. Salman, “Monolithic 3D integrated circuits: Recent trends and future prospects,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2021.
- [16] Z. Li *et al.*, “RRAM-DNN: An RRAM and model-compression empowered all-weights-on-chip DNN accelerator,” *IEEE JSSC*, vol. 56, no. 4, pp. 1105–1115, 2020.
- [17] P. Shukla, V. F. Pavlidis, E. Salman, and A. K. Coskun, “Tread-m3d: Temperature-aware dnn accelerators for monolithic 3-d mobile systems,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4350–4363, 2023.
- [18] P. Shukla, S. S. Nemtzow, V. F. Pavlidis, E. Salman, and A. K. Coskun, “Temperature-aware optimization of monolithic 3d deep neural network accelerators,” in *Proc. of ASP-DAC*, 2021, pp. 709–714.
- [19] J. M. Joseph *et al.*, “Architecture, dataflow and physical design implications of 3d-ics for dnn-accelerators,” in *IEEE ISQED*, 2021, pp. 60–66.
- [20] C. Peltekis, D. Filippas, G. Dimitrakopoulos, C. Nicopoulos, and D. Pnevmatikatos, “Arrayflex: A systolic array architecture with configurable transparent pipelining,” in *IEEE DATE*, 2023, pp. 1–6.
- [21] A. Samajdar *et al.*, “A systematic methodology for characterizing scalability of DNN accelerators using scale-sim,” in *IEEE ISPASS*, 2020, pp. 58–68.
- [22] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016.
- [23] C. Wang, Z. Wang, S. Li, Y. Zhang, H. Shen, and K. Huang, “Ews: An energy-efficient cnn accelerator with enhanced weight stationary dataflow,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 7, pp. 3478–3482, 2024.
- [24] R. Xu, S. Ma, Y. Wang, and Y. Guo, “Cmsa: Configurable multi-directional systolic array for convolutional neural networks,” in *2020 IEEE 38th International Conference on Computer Design (ICCD)*, 2020, pp. 494–497.
- [25] M. M. Sabry Aly *et al.*, “The n3xt approach to energy-efficient abundant-data computing,” *IEEE*, vol. 107, no. 1, pp. 19–48, 2018.
- [26] S. R. Lee *et al.*, “Multi-level switching of triple-layered taos rram with excellent reliability for storage class memory,” in *2012 Symposium on VLSI Technology (VLSIT)*. IEEE, 2012, pp. 71–72.
- [27] P. Shukla, D. Aguren, T. Burd, A. K. Coskun, and J. Kalamatianos, “Temperature-aware sizing of multi-chip module accelerators for multi-dnn workloads,” in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023, pp. 1–6.
- [28] F. Chen, L. Song, H. Li, and Y. Chen, “Marvel: A vertical resistive accelerator for low-power deep learning inference in monolithic 3d,” in *IEEE DATE*, 2021, pp. 1240–1245.
- [29] X. Liu, W. Wen, X. Qian, H. Li, and Y. Chen, “Neu-noc: A high-efficient interconnection network for accelerated neuromorphic systems,” in *ASP-DAC*, 2018, pp. 141–146.
- [30] B. K. Joardar, J. R. Doppa, P. P. Pande, H. Li, and K. Chakrabarty, “Accured: High accuracy training of cnns on rram/gpu heterogeneous 3-d architecture,” *IEEE TCAD*, vol. 40, no. 5, pp. 971–984, 2021.
- [31] S. Thoziyoor *et al.*, “CACTI 6.5,” *hpl.hp.com*, 2009.
- [32] M. Poremba, S. Mittal, D. Li, J. S. Vetter, and Y. Xie, “Destiny: A tool for modeling emerging 3d nvm and edram caches,” in *IEEE DATE*, 2015, pp. 1543–1546.
- [33] Z. Yuan, P. Shukla, S. Chetoui, S. Nemtzow, S. Reda, and A. K. Coskun, “Pact: An extensible parallel thermal simulator for emerging integration and cooling technologies,” *IEEE TCAD*, vol. 41, no. 4, pp. 1048–1061, 2021.
- [34] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, “Express virtual channels: towards the ideal interconnection fabric,” *SIGARCH Comput. Archit. News*, vol. 35, no. 2, p. 150–161, jun 2007.
- [35] A. Guler and N. K. Jha, “Hybrid monolithic 3D IC floorplanner,” *IEEE TVLSI*, vol. 26, no. 10, pp. 1868–1880, 2018.
- [36] D. Bhattacharya and N. K. Jha, “Ultra-high density monolithic 3-d finfet sram with enhanced read stability,” *IEEE TCAS I: Regular Papers*, vol. 63, no. 8, pp. 1176–1187, 2016.
- [37] S. K. Samal, D. Nayak, M. Ichihashi, S. Banna, and S. K. Lim, “Monolithic 3d ic vs. tsv-based 3d ic in 14nm finfet technology,” in *IEEE S3S*, 2016, pp. 1–2.
- [38] H. Kwon, L. Lai, T. Krishna, and V. Chandra, “Heterogeneous dataflow accelerators for multi-DNN workloads,” in *IEEE HPCA*, 2021, pp. 71–83.
- [39] K. Wei *et al.*, “Thermal analysis and junction temperature estimation under different ambient temperatures considering convection thermal coupling between power devices,” *Applied Sciences*, vol. 13, no. 8, p. 5209, 2023.
- [40] K. Skadron *et al.*, “Temperature-aware microarchitecture,” *ACM SIGARCH Computer Architecture News*, vol. 31, no. 2, pp. 2–13, 2003.