MOCCA: A Process Variation Tolerant Systolic DNN Accelerator using CNFETs in Monolithic 3D

Samuel J. Engers
Indiana University Bloomington
sengers@iu.edu

Cheng Chu
Indiana University Bloomington
chu6@iu.edu

Dawen Xu
Seehi Microelectronics Co., Ltd
xudawen@seehi.com

Ying Wang
Chinese Academy of Sciences
wangying2009@ict.ac.cn

Fan Chen
Indiana University Bloomington
fc7@iu.edu

ABSTRACT

Hardware accelerators based on systolic arrays have become the dominant method for efficient processing of deep neural networks (DNNs). Although such designs provide significant performance improvement compared to its contemporary CPUs or GPUs, their power efficiency and area efficiency are greatly limited by the large computing array and on-chip memory. In this work, we demonstrate that we can further improve the efficiency of systolic accelerators using emerging carbon nanotube field-effect transistors (CNFETs) by stacking the computing logic and on-chip memory on multiple layers and utilizing monolithic 3D (M3D) vias for low-latency communication. We comprehensively explore the design space and present MOCCA, the first process variation tolerable CNFET-based systolic DNN accelerator. We validate MOCCA against previous 2D accelerators on state-of-the-arts DNN models. On average, MOCCA achieves the same throughput with 6.12× and 2.12× improvement respectively on performance and power efficiency in a 2× reduced chip footprint.

CCS CONCEPTS

• **Hardware → Emerging technologies**.

KEYWORDS

CNFET, DNN, systolic array, process variation

ACM Reference Format:

Samuel J. Engers,, Cheng Chu,, Dawen Xu,, Ying Wang,, and Fan Chen, . 2022. MOCCA: A Process Variation Tolerant Systolic DNN Accelerator using CNFETs in Monolithic 3D. In *Proceedings of the Great Lakes Symposium on VLSI 2022 (GLSVLSI '22), June 6–8, 2022, Irvine, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3526241.3530380

1 INTRODUCTION

Domain-specific accelerators [1] have become prevalent for tackling the massive computing requirements in Deep Neural Networks (DNNs) processing through intensive data reuse (e.g., ~1000 operations per weight) and a large (e.g., >30 MB) software-managed on-chip memory. Despite the significant performance improvement, the power and area efficiency of such designs are greatly limited by the large computing array and on-chip memory. In a 4-chip Google TPU ASIC [1], the die power can reach to 40 *Watt*. At the same time, the computing array and on-chip SRAM together occupy ~70% of the chip area. The TPU ASIC is implemented in 28 *nm* process and clocked at 700 *MHz*. The power and area efficiency at more advanced technology nodes and/or higher frequencies will be further deteriorated due to the diminishing scaling returns of silicon filed-effect transistors (FETs).

Driven by the ever-increasing need for next-generation efficient electronic systems, various beyond CMOS technologies [2] are being explored in recent years. In particular, computing systems built from FETs fabricated with Carbon nanotubes (CNTs) have shown great promise to significantly improve power efficiency, due to its ~9× improved energy-delay product (EDP) compared with CMOS [3], scalability down to 3 *nm* and beyond [4], and superior intrinsic thermal properties [5]. Moreover, the Carbon Nanotube Field-Effect Transistors (CNFETs) circuits can be fabricated in the low-temperature back-end-of-line (BEOL) directly over silicon substrate [6], providing a unique opportunity to achieve area-efficient monolithic 3D (M3D) integrated nanosystems.

Despite being technically attractive, CNFETs suffer from substantial process variations [7] due to inherent imperfections in the current synthesis processes used to produce CNTs. Major imperfections include CNT density variations, mis-aligned CNTs, and metallic CNTs (*m*-CNTs), because they all directly affect the CNT-count in each device, resulting in increased device delay variation and system performance degradation. Fortunately, the process variation in CNFETs demonstrated a strong direction-dependant correlation [8], which can be exploited at circuit- and architectural-level to realize PV-tolerable designs. thereby improving system performance and power efficiency. In this work, we implement such CNFET-based accelerator with M3D and show that the integration of leading technologies will significantly advance future abundant-data computing systems. The key contributions of this paper are:

(1) We study the impact of CNT-count variations on the performance of systolic DNN accelerators. Based on the 2D feature of computing arrays, we propose to exploit the spatial correlation of CNT variations to improve the system perforance in CNFET-based DNN accelerators.
(2) We present a MAC layout for CNTFET MAC arrays by considering the process variation correlations. A *outlier skiping* scheme

GLSVLSI '22, June 6–8, 2022, Irvine, CA, USA

ACM ISBN 978-1-4503-9322-5/22/06…$15.00
https://doi.org/10.1145/3526241.3530380

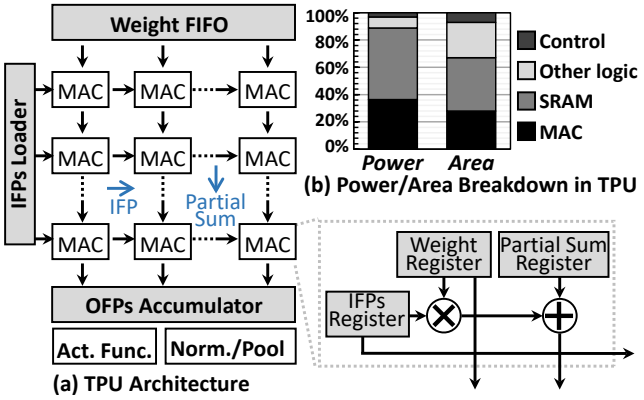

Figure 1: The Google TPU.

is proposed based on our investigation on the delay distributions, which can provide 1.6× performance improvement with negligible hardware overhead.

(3) We propose a finer-grained SRAM buffer organization. On-chip memory is realized in small banks on top of the computing layer and communicates directly with the lower computing layer via vertical inter-layer vias.

(4) We present MOCCA, the first CNFET-based systolic DNN accelerator, which achieves 6.12× improvement on performance, and 2.12 × improvement on power efficiency compared to the 2D TPU baseline design.

2 BACKGROUND AND MOTIVATION

2.1 Systolic DNN Accelerator

Systolic architecture. Figure 1 (a) shows the Google TPU [1] architecture, which will be used as the baseline systolic architecture in this paper. TPU consists of a 256×256 8-bit Multiply-and-Accumulate (MAC) units connected with 2D mesh. *Weights* are fetched from off-chip memory and stored into a FIFO, while *Input Feature Maps (IFPs)* are loaded to the *IFPs Loader* from a 24MB on-chip SRAM. Each MAC unit is equipped with a few registers for operand buffering. Coupled with such carefully-designed memory hierarchy, weights are preloaded into the MACs and remain stationary before being exhaustively used, IFPs are fed from the edges and streamed from left to right, while the generated partial sums are forwarded downstream in each cycle. The *Output Feature Maps (OFPs)* are accumulated at the *OFPs Accumulator*. The efficiency of such systolic array comes from communication only between nearest neighbours, which in turn provides high compute density, low-overhead input buffering, and simplified routing.

Power/area breakdown. Figure 1 (b) shows the area and power breakdown in TPU. The major bottleneck comes from the systolic MAC array and the 24MB on-chip SRAM, consuming respectively 36.4% and 52.5% of the power consumption. For a 4-chip, 28 *nm*, 700 *MHz* TPU, the total power consumption is 40 *Watt*, but 70% of which is CMOS leakage power. The reasons for such high static power dissipation are twofold. First, the leakage current in CMOS logic increase exponentially beyond 45 *nm*.making energy-efficient computing at highly-scaled process technologies very challenging. Second, the large systolic computing grid and on-chip memory incur a large area overhead, resulting in a proportional increase in the static power of the chip.

2.2 CNFET & Monolithic 3D system

Device. A CNFET shares the same device structure with MOSFET, the difference is that it utilizes CNTs instead of bulk silicon as the channel material, CNTs have a 1∼2 *nm* diameter hollow cylindrical structure with remarkable electrical and thermal properties [6]. Typically, multiple parallel semiconducting CNTs (*s*-CNTs) are grown using chemical synthesis and transferred to a substrate. CNT regions under the the *source* and *drain* are heavily doped, while the region of CNTs under the *gate* are undoped and its conductivity is controlled by the gate voltage.

Process variations. CNFETs performance are quantized in terms of the CNT-count in each device. Since CNTs are grown using chemical synthesis and then transferred onto a substrate to form channels in CNFETs, it is extremely difficult to ensure uniform density (i.e., *CNT density variation*) and precise positioning of CNTs (i.e., *mis-aligned CNTs*) during the manufacturing process [8]. In addition, roughly 33% of the CNTs produced by typical CNT growth processes are metallic (*m*-CNT) [9], which can cause excessive leakage and even logic gate malfunction. Current *m*-CNT removal techniques [10] may aggravate the CNT density variations by inadvertently removing some functionally correct semiconducting CNTs (*s*-CNTs). All the aforementioned process variations compromise reliability of CNFETs and lead to increased device delay variations or incorrect logic functionality.

Monolithic 3D. (M3D) [11] integration allows multiple active device layers be sequentially fabricated on the same substrate with fine-grained, nano-scale inter-layer vias (ILVs). Such M3D systems provides orders of magnitude smaller area overhead than conventional through-silicon-vias (TSVs) based 3D designs [12] . The Carbon Nanotube Field-Effect Transistors (CNFETs) circuits emerge as a perfect candidate for implementing M3D systems as they can be reliably fabricated in the low-temperature back-end-of-line (BEOL).

3 MOCCA

3.1 Modeling of Process Variation

We use VARIUS [13] to generate CNT-count samples based on the statistic tool R and its package geoR. VARIUS adopts a multi-variate normal distribution with the spherical structure to model spatial correlations. Because the presence of *m*-CNTs, the minimum CNT-count in a practical CNFET is expected to be > 8.9. Therefore, we adopt the experimentally validated CNT-count model with $\mu = 9$ and $\sigma = 2.1$ [8]. The probability of *m*-CNT, removed *m*-CNT and *s*-CNT are obtained from [14]. These variations are normally reflected as the gate delay and driving capability. We vary the variation parameters in the Variation-Aware Nanosystem Design Kit [15], and incorporate them with the Stanford University Virtual Source CNFET Model [16] for device simulation. We leverage RTL-based simulations to model the impact of timing violations. We quantify the nominal delay/energy as the critical path delay and associated energy when there are no timing variations. We adopt the CNFET-based SRAM in [17]. Similar to previous work [18], we upsize control, and nonlinear units (i.e., all non-MAC logic) to ensure system reliability.

3.2 Design Space Exploration

Layout of a MAC array. Due to the spatial correlation in CNT-count, the layout of CNFET circuits has a dramatic impact on the system performance. Therefore, careful considerations should be

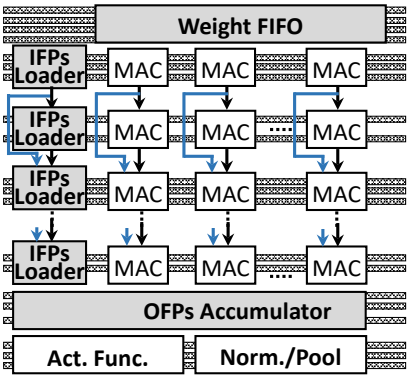

Figure 2: The proposed MAC array layout.

given to the layout design. Figure 2 shows the proposed layout of the computing units in MOCCA. Since *MAC Arrays* inherently exhibit a 2D organization, they inevitable would suffer from large variation at one dimension. For the *IFPs Loader*, we place them close to the corresponding MAC rows so that the data loading and data processing will match each other consistently. *Weight FIFO*, and vector computing units including *OFPs Accumulator*, *Activation Function unit*, and *Normalization/Pooling Unit* are implemented along the CNT growth direction. In general, our goal is to ensure minimum variation in the monolithic computing array. All analysis below are based on the MAC delay shown in Figure 2.

MAC array size v.s. power efficiency. Figure 3 demonstrated the trend in power efficiency (i.e., *FPs/Watt*) when workloads are run on MAC arrays of different sizes. All data are normalized to the power efficiency of the baseline 256×256 MAC array size. In a glance it seems that the optimal MAC array size in terms of power efficiency for all workloads lies in between 32×32 and 64×64 instead of 256×256 used in TPU, which is consistent with previous work [19]. The reason can be attributed to (1) the limited on-chip memory bandwidth fail to support the intensive memory requirement for large (e.g., >64×64) MAC arrays; and (2) the intrinsic sparsity [20] in DNNs yield low resource utilization but large power overhead when large computing array is applied.

MAC array size v.s. frequency. We also explored the highest frequency for MAC arrays. In general, as we reduce the array size, a finer-grained access latency could be obtained for each individual MAC array, and hence, the highest frequency of the MAC arrays increases. In contrast, the MAC frequency reduces significantly due to the severe process variation as the array dimension increases. As an example, experiment results show that an approximately ideal 3× improved latency [3] can be achieved with an 8×8 MAC size. We summarize the highest frequency achieved with differnet MAC array sizes in Table 1.

3.3 The MOCCA Architecture

Overview. Figure 4 shows the overview of the proposed MOCCA architecture. Similar to the baseline design, we place the MAC array

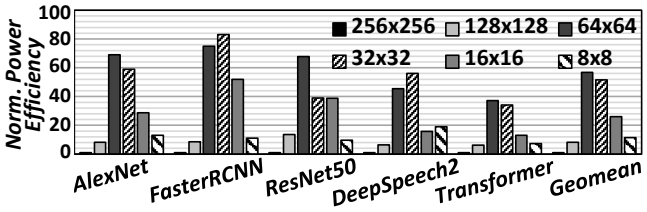

Figure 3: The comparison on power efficiency.

Table 1: The highest frequency in different MAC sizes.

Size	256×256	64×64	32×32	16×16	8×8
Frequency (GHz)	0.9	1.2	1.8	2.1	2.4

and SRAM buffer to two separate layers, namely *Computing Layer* and *Memory Layer*, respectively. The *interface circuits*, *Control Logic*, and other supporting circuits are all placed on the top *Memory Layer* to balance the chip footprint. For dense inter-layer connection, we adopt the fine-grained vertical integration through denser nano-scale inter-layer vias (ILVs) [21]. The overall MOCCA architecture obtains a uniformly distributed ~166.4mm^2 chip footprint.

Computing layer. As shown in Figure 2, each row of MACs are aligned with correlated CNFETs. To obtain better-than-worst-case performance, we implement the MAC array with a size of 32×32, which is the optimal array dimension in term of both power efficiency and MAC frequency as we demonstrated in Figure 3 and Table 1. All MAC arrays are connected via a centralized crossbar. Further more, we set out to improve performance by skipping the outlier rows in a MAC array. Specifically, we allow each MAC implement an extra wire to bypass their neighboring MAC as highlighted in the blue wire in Figure 2. Such *outlier skipping* incurs ~5% area overhead but significantly enhance the MAC operating frequency as we will show in Section 4.

Memory layer. We modify the CNFET-based register file model in [17] for SRAM simulation. According to our analysis, the access delay variation is worse than it in MAC array, which is consistent with the conclusion in [17]. Therefore, we divide the 24MB SRAM to six 4MB banks and allow each bank to be accessed at its own frequency. Each of the upper SRAM bank is directly connected to corresponding bottom MAC arrays for fast access. All the bank in the memory layer are also connected through a central crossbar.

Interconnection between layers. It is safe for us to assume that the chip can be fully tested to obtain the latency information after fabrication. MOCCA always matches the fast MAC with the fast SRAM bank. For instance, assuming MAC *A* and memory bank *B* are respectively the fast in each bank as highlighted with red in Figure 4, he direct ILV between the two layers can provide the fastest data delivery. If memory bank *C* is the fastest memory bank, based on the direct IVL between the central crossbars in the two layers, the data can still be delivered from *C* to *A* within the same cycle numbers.

Design overhead. All circuits components in MOCCA are implemented with CNFETs. The overall MOCCA architecture consumes averagely 16.3 *Watt* power and occupies 166.4mm^2 chip area.

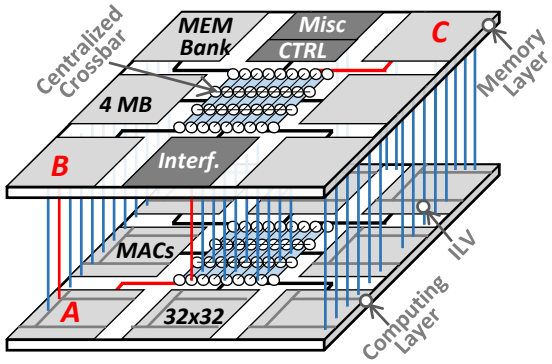

Figure 4: The MOCCA architecture.

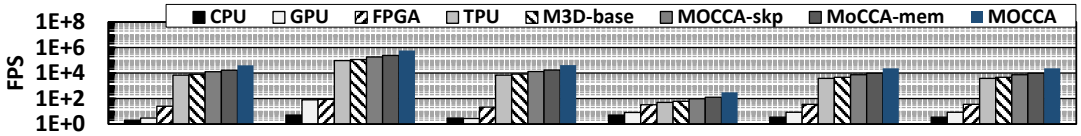

Figure 5: The performance of different designs.

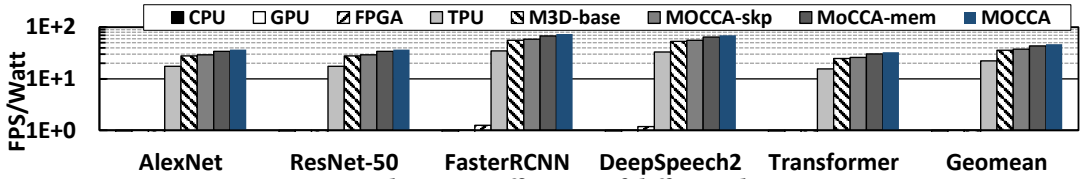

Figure 6: The power efficiency of different designs.

4 EVALUATION

Benchmark. We evaluate the MOCCA architecture using five state-of-the-art DNN models covering a wide range of applications including image classification, object detection, speech recognition, and natural language processing. LeNet is trained with MNIST for simple hand-written digits, while AlexNet, VGG16, ResNEt-18, and MobileNet are trained with ImageNet for complex classification tasks. All models are trained in Tensorflow. We quantized both the actications and weights of all CNNs with 8-bit.

Scheme. We selected Intel Xeon E5-2630 V3, 8-core CPU, an Nvidia Tesla P100 GPU, a Xilinx Virtex7 FPGA, and 2D ASIC chip Google TPU [1]. Google TPU comprises four chips, each of which can achieve larger throughput but consume more power. We denote the baseline CNFET TPU demonstrated in Figure 1 as *M3D-base*. To further evaluate the individual benefits of *outlier skipping* and *memory banking*, we provide their results respectively as *MOCCA-skp*, *MOCCA-mem*.

Simulation. To evaluate the performance and energy of TPU, we adopt the Scale-sim [19] simulator to captures the latency, energy, resource utilization, and access counts for various components in the architecture. We assume a low-power DRAM interface with 4 pJ/bit. The run times for CPU/GPU platform are measured by Tensorflow and the energy costs are measured on real hardware.

Results on performance. Figure 5 compares the performance among CPU, GPU, and different DNN accelerators. We use frame per seconds (FPS) as the metric for evaluation. The baseline *M3D-base* achieve a geomean 1.2× improvement compared to the 2D CMOS baseline, which is far below the projected 3× reduction in delay and 9× improvement in EDP. With our proposed *outlier skipping* in MAC arrays, we can see a 1.6× enhancement in performance. SRAM bank allows each CNFET-based SRAM bank run at its fast speed, the overall performance can be further improved by 2.1×. In all cases, MOCCA achieve the best FPS as it combines the benefits of optimal-sized MAC array and SRAM banking. A 6.12× improvement in FPS can be achieved compared to the M3D baseline design.

Results on power efficiency. Figure 6 exhibits the power efficiency for all the designs in terms of the DNN performance per Watt. In general, the FPS per Watt of different designs share the similar trend to their performance shown in Figure 5. For all the benchmarks, MOCCA demonstrated the highest power efficiency. Compared to the 2D CMOS TPU design, the baseline M3D TPU achieves 1.6× improvement, while *outlier skipping* and SRAM bank respectively achieve a 1.05× and 1.22× improvement. For all the evaluated benchmarks, MOCCA achieves a geomean of 1.32× and 2.5× improvement on performance per Watt compared to the M3D and 2D baseline.

5 CONCLUSION

In this paper,we study the impact of CNT process variations on the performance of systolic accelerators. Based on detailed design space exploration, we propose MOCCA, a CNFET-based DNN accelerators which features optimally-sized computing arrays and SRAM banks. With outlier skipping in MAC arrays and fine-grained inter-layer communication, we show that MOCCA achieves 6.12× improvement on performance, and 2.12× improvement on power efficiency compared to the 2D TPU baseline design.

REFERENCES

[1] N. P. Jouppi *et al.*, "In-Datacenter Performance Analysis of a Tensor Processing Unit," in *ISCA*, 2017.
[2] D. Woods, "Photonic Neural Networks," in *Nature Physics*, 2012.
[3] J. Deng *et al.*, "A compact SPICE Model for Carbon-Nanotube Field-Effect Transistors Including Nonidealities and Its Application—Part II: Full Device Model and Circuit Performance Benchmarking," *IEEE T-ED*, 2007.
[4] A. D. Franklin *et al.*, "Sub-10 nm Carbon Nanotube Transistor," *Nano letters*, 2012.
[5] C. Yu *et al.*, "Thermal Conductance and Thermopower of an Individual Single-Wall Carbon Nanotube," *Nano Letters*, 2005.
[6] M. M. Shulaker *et al.*, "Monolithic 3D Integration of Logic and Memory: Carbon Nanotube FETs, Resistive RAM, and Silicon FETs," in *IEDM*, 2014.
[7] S. Banerjee *et al.*, "Analysis of the Impact of Process Variations and Manufacturing Defects on the Performance of Carbon-Nanotube FETs," *IEEE TVLSIS*, 2020.
[8] J. Zhang *et al.*, "Carbon Nanotube Circuits in the Presence of Carbon Nanotube Density Variations," in *DAC*, 2009.
[9] J. Zhang *et al.*, "Probabilistic Analysis and Design of Metallic-Carbon-Nanotube-Tolerant Digital Logic Circuits," *TCAD*, 2009.
[10] G. Zhang *et al.*, "Selective Etching of Metallic Carbon Nanotubes by Gas-Phase Reaction," *Science*, 2006.
[11] M. M. S. Aly *et al.*, "The N3XT Approach to Energy-Efficient Abundant-Data Computing," in *Proceedings of the IEEE*, 2018.
[12] F. Chen *et al.*, "Marvel: A Vertical Resistive Accelerator for Low-Power Deep Learning Inference in Monolithic 3D," in *DATE*, 2021.
[13] S. Sarangi *et al.*, "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects," in *IEEE Trans. Semicond. Manuf.*, 2008.
[14] J. Zhang *et al.*, "Design Guidelines for Metallic-Carbon-Nanotube-Tolerant Digital Logic Circuits," in *DAC*, 2008.
[15] G. Hills, "Variation-Aware Nanosystem Design Kit (NDK)," 2015.
[16] C. . Lee *et al.*, "A Compact Virtual-Source Model for Carbon Nanotube FETs in the Sub-10-nm Regime—Part I: Intrinsic Elements," *T-ED*, 2015.
[17] T. Li *et al.*, "CNFET-based High Throughput Register File Architecture," in *ICCD*, 2016.
[18] G. Hills *et al.*, "TRIG: Hardware Accelerator for Inference-based Applications and Experimental Demonstration using Carbon Nanotube FETs," in *DAC*, 2018.
[19] A. Samajdar *et al.*, "SCALE-Sim: Systolic CNN Accelerator Simulator," *arXiv e-prints*, 2018.
[20] S. Han *et al.*, "Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding," in *ICLR*, 2016.
[21] J. Shi *et al.*, "A 14nm FinFET Transistor-Level 3D Partitioning Design to Enable High-Performance and Low-Cost Monolithic 3D IC," in *IEDM*, 2016.