Preprint of paper from VLSI-SoC 2024, Tangier, Morocco, October 2024.

Commercial Evaluation of Zero-Skipping MAC Design for Bit Sparsity Exploitation in DL Inference

Harideep Nair*§, Prabhu Vellaisamy*§, Tsung-Han Lin†, Perry Wang†, Shawn Blanton*, and John Paul Shen*
* Carnegie Mellon University
† MediaTek USA Inc.

***Abstract*—General Matrix Multiply (GEMM) units, consisting of multiply-accumulate (MAC) arrays, perform bulk of the computation in deep learning (DL). Recent work has proposed a novel MAC design, Bit-Pragmatic (PRA), capable of dynamically exploiting bit sparsity. This work presents *Oz*MAC (*Omit-zero*-MAC), a modified re-implementation of PRA, but extends beyond earlier works by performing rigorous post-synthesis evaluation against binary MAC design across multiple bitwidths and clock frequencies using TSMC N5 process node to assess commercial implementation potential. We demonstrate the existence of high bit sparsity in eight pretrained INT8 DL workloads and show that 8-bit *Oz*MAC improves all three metrics of area, power, and energy significantly by 21%, 70%, and 28%, respectively. Similar improvements are achieved when scaling data precisions (4, 8, 16 bits) and clock frequencies (0.5 GHz, 1 GHz, 1.5 GHz). For the 8-bit *Oz*MAC, scaling its frequency to normalize the throughput, it still achieves 30% improvement on both power and energy.**

***Index Terms*—Zero-skipping multiply-accumulate, commercial TSMC N5 evaluation, bit sparsity, deep learning inference**

I. Introduction and Background

General matrix multiply (GEMM) hardware, employing large arrays of multiply-accumulate (MAC) units, is the core compute fabric for modern deep learning accelerators (DLAs) [1]. A conventional bit-parallel MAC unit consists of a combinational array multiplier and an adder to accumulate the product, and a register to store the value. Any improvement on the MAC unit design is replicated many fold in the large MAC arrays, yielding potential for significant reduction in hardware complexity of DLAs [2]–[5]. Further, current industry standard for inference has moved from 32-bit floating-point (FP32) format to 16-bit floating-point (FP16) and 8-bit integer (INT8) formats. A recent study from IBM [6] summarizes the trend towards lower precision, highlighting imminent move towards 4-bit integer (INT4) and 2-bit integer (INT2) in the near future.

Recent work on Bit-Pragmatic (PRA) [3] has proposed a novel MAC design that leverages bit sparsity (i.e., the number of '0' bits within a binary value) to perform bit serial compute efficiently by skipping over zero bits using simple serial shift-and-add compute. In this work, we present a re-implementation of PRA with minor modifications for added hardware efficiency, called *Oz*MAC, but the main contribution of this work is the rigorous state-of-the-art evaluation of *Oz*MAC using commercial TSMC N5 (5nm) process node across multiple data precisions and clock frequencies, significantly extending beyond prior works utilizing TSMC 65nm technology and higher precision single-clock configurations.

Key contributions of our work are:

- Present *Oz*MAC based on Bit-Pragmatic (PRA), capable of exploiting dynamic bit sparsity by skipping over zero bits in binary values. This zero-skipping design in itself is not novel; the main focus of this work is its evaluation.
- Implement wide range of *Oz*MAC designs using commercial design tools and TSMC N5 process design kit.
- Evaluate power-performance-area (PPA) for various data precisions (4-bits, 8-bits, 16-bits) and clock frequencies (500 MHz, 1 GHz, 1.5 GHz), against binary MAC.
- Demonstrate high bit sparsity in eight DL models leading to significant power reduction and how this can be used to increase *Oz*MAC's throughput via frequency scaling.

The paper is organized as follows. *Oz*MAC microarchitecture is briefly summarized in Section II followed by hardware evaluation methodology in Section III. We present sparsity and corresponding PPA evaluation in Section IV, followed by bit-width and frequency scaling analysis in Sections V and VI respectively. Finally, Section VII presents key conclusions.

II. OzMAC Microarchitecture and Design

*Oz*MAC microarchitecture, derived from the Inner Product Unit within Bit-Pragmatic (PRA) [3], consists of three simple functional modules as shown in Fig. 1: 1) *Oz-encoder*, 2) *shifter*, and 3) *accumulator*. *Oz-encoder* is a Finite State Machine which keeps track of the current and next positions of '1' in the input bit pattern. Using this information, it outputs a one-hot encoded value capturing the bit positions of '1's every clock cycle for as many cycles as the number of '1's. For example, as illustrated in Fig. 1, the input '0101_2' is encoded as two one-hot values spanning two clock cycles: '0100_2' in the first cycle and '0001_2' in the next cycle. By doing this, it skipped over the two '0's and only incurs compute cycles for the '1's. The *Oz-encoded* input then goes to the shifter that determines the shift magnitude of the second input. The appropriately shifted second input is then added to the accumulator value. The minor modification to PRA employed in *Oz*MAC is the *Oz*-encoder which feeds a 1-hot representation of the shift value to the shifter in contrast to PRA's oneffset generator which outputs a binary shift value. Employing 1-hot input simplifies the shifter hardware at the expense of more input lines (negligible overhead compared to reduction of shifter gate complexity).

§Equal contribution

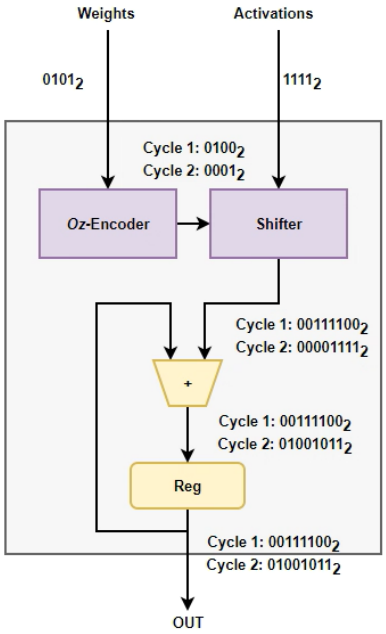


Fig. 1: *Oz*MAC (based on PRA [3]) with example compute.

III. Hardware Framework for Evaluation

We perform rigorous, industry-standard evaluation of the *Oz*MAC design to get accurate PPA and energy results and compare against a conventional bit-parallel bMAC. The technology library used for evaluation is the commercial TSMC N5 (5nm) process node, with Synopsys design tools employed for simulation, synthesis, and power calculations.

First, *Oz*MAC RTL design is created in SystemVerilog, with functional verification performed using Synopsys VCS. A synthetic dataset with 1000 sample weights and activation values is developed, with the values reflecting the sparsity levels of the DL benchmarks under consideration. This allows for the appropriate switching activity to be captured, as well as resultant average *Oz*MAC compute cycles to be reported by the means of a testbench. Next, lint check is performed on the SystemVerilog source files using Synopsys SpyGlass and then synthesis is performed to convert the RTL-level design into a gate-level netlist using Synopsys Design Compiler, sourcing TSMC N5 library files. Gate-level netlist simulation is then performed for verification and collection of the switching activity of the design in the form of a SAIF dump. The SAIF dump is then sourced along with the netlist to perform accurate power calculations using Synopsys PrimeTime PX.

IV. Sparsity Scaling Analysis

Like PRA, *Oz*MAC performs highly efficient shift-and-add operations and trades off latency for lower area and power. The "omit-zero" capability of *Oz*MAC is key to mitigating this latency overhead by exploiting dynamic bit sparsity in input data. In other words, higher bit sparsity (i.e., more zero bits in the input data) will result in shorter compute latency and thereby lower energy consumption.

TABLE I: Bit Sparsity and Cycle-Count Overhead for Pretrained Weights for Eight INT8 Quantized DL Benchmarks.

DL Benchmark	Average number of '1' bits (Actual cycle-count overhead)	Bit Sparsity Percentage
MobileNetV2	2.334	70.83%
MobileNetV3	1.711	78.61%
InceptionV3	2.430	69.62%
ShuffleNetV2	2.583	67.71%
GoogleNet	2.461	69.24%
ResNet18	2.398	70.02%
ResNet50	2.495	68.81%
ResNeXt101	2.289	71.39%

TABLE II: TSMC N5 PPA and Energy (averaged across eight DL benchmarks) for 8-bit OzMAC and bMAC at 500 MHz.

MAC Hardware	Area (μm^2)	Power (mW)	Latency (ns)	Energy (pJ)
bMAC	25.361	0.084	2	0.167
OzMAC	19.996	0.025	4.76	0.120
% Improvement	21.2	69.7	-	28.0

As illustrated in Table I, we use eight pretrained and quantized INT8 models, available as part of PyTorch's Torchvision library, that are widely used in state-of-the-art DL literature. Layer-by-layer analysis of the converged weights and activation values resulting from running ImageNet benchmark inputs illustrate the sparsities inherent in these benchmarks. For each model, we extract the average number of '0' bits in every 8-bit weight value across all the layers and calculate the bit sparsity as the percentage of '0' bits over the total number of bits. The DL models have close to 70% bit sparsity with MobileNetV3 having the highest sparsity of 78.61%. Table I also shows the number of '1' bits, which is equal to the compute latency in cycles. It can be seen that the effective compute latency for 8-bit *Oz*MAC owing to bit sparsity is between 1.7-2.5 cycles, much lower than the worst-case latency of 8 cycles. Comparing this to 1 cycle latency of bMAC, *Oz*MAC's power consumption must be 1.7-2.5x lower than that of bMAC to achieve similar energy efficiency.

Table II provides the die area, power, latency and energy consumption of *Oz*MAC and bMAC averaged across the eight DL benchmark models. Note that the power consumption values are obtained via PTPX using benchmark-specific test vectors that capture bit sparsity characteristics. The operating frequency for both designs is 500 MHz. An 8-bit conventional bMAC computes 1 MAC operation in 2 ns (1 cycle) while consuming about 25 μm^2 area, 84 μW power, and 167 fJ energy, whereas *Oz*MAC only consumes about 20 μm^2 area, 25 μW power, and 120 fJ energy while incurring 4.76 ns latency on average. Compared to conventional bMAC, this amounts to 21% less die area, 70% less power, and 28% less energy with 2.38x higher latency. This significant improvement in all three metrics can be attributed to three key factors: 1) simpler shift-and-add hardware with less area and leakage power footprint, 2) serial *Oz*-encoder that enables significant reduction in signal transitions at the input stage, thereby improving dynamic power, and 3) capability to exploit

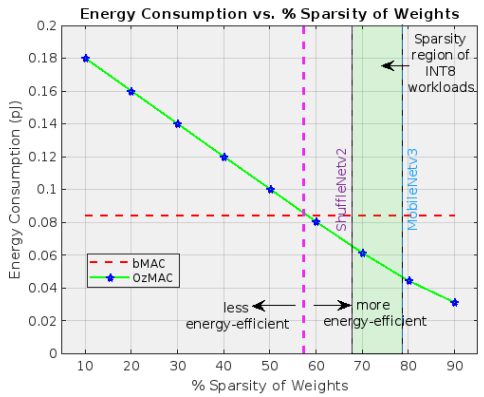


Fig. 2: Energy consumption vs. % bit-sparsity. Green-shaded region depicts the sparsity regions for Table I workloads.

the high bit sparsity present in the DL benchmarks (Table I).

Given the power consumption values from Table II, it can be seen that *Oz*MAC reduces power by 3.36x on average. This implies that for an 8-bit *Oz*MAC design, it can incur up to 3.36 clock cycles on latency overhead per MAC operation, before its energy consumption exceeds that of bMAC. We can calculate the minimum bit sparsity needed for *Oz*MAC to maintain superior energy efficiency as $1 - \frac{3.36}{8} = 58\%$. Fig. 2 plots the energy consumption across varying bit sparsity, to demonstrate this cross-over point at 58% sparsity. Interestingly, all eight DL benchmarks exhibit bit sparsity higher than the threshold of 58% as can be seen from Fig. 2. Significant reduction in power consumption, coupled with sparsity-induced latency reduction, allows *Oz*MAC to maintain superior energy efficiency over bMAC in spite of multi-cycle latency overhead. For throughput-sensitive applications, the higher latency of *Oz*MAC can be addressed via frequency scaling, as will be demonstrated later in Section VI-B.

Key Takeaway: *Oz*MAC achieves significant reduction in area, power and energy relative to bMAC for typical DL workloads, by exploiting inherent bit sparsity.

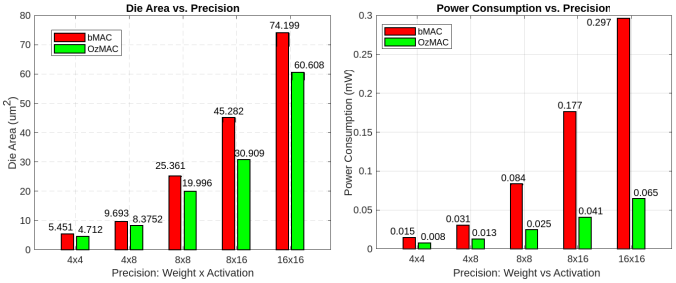
V. Precision Scaling Analysis

Inference precision for DL workloads has been trending from 16-bits in the past to the current 8-bits with projection further down to 4-bits. Table III provides TSMC N5 PPA for five integer precision configurations: 1) 4-bit weights, 4-bit activations (4x4), 2) 4-bit weights, 8-bit activations (4x8), 3) 8-bit weights, 8-bit activations (8x8), 4) 8-bit weights, 16-bit activations (8x16), and 5) 16-bit weights, 16-bit activations (16x16). The mixed precisions, 4x8 and 8x16, are used to accommodate typical workloads that demand higher activation precision compared to weight precision. The corresponding area and power results are also plotted in Fig. 3.

Based on Table III, the smallest (4x4) *Oz*MAC and bMAC designs consume 4.7 μm^2 area, 8 μW power, 22 fJ energy, and 5.4 μm^2 area, 15 μW power, 31 fJ energy, respectively. Compared to 4x4 designs, the largest (16x16) *Oz*MAC incurs about 13x, 8x and 27x increase whereas 16x16 bMAC incurs

TABLE III: TSMC N5 PPA at 500 MHz across varying bit precision of weights and activations: 4 bits, 8 bits and 16 bits.

MAC Hardware (wgt x act)	Area (μm^2)	Power (mW)	Latency (ns)	Energy (pJ)
bMAC (4x4)	5.451	0.015	2	0.031
OzMAC (4x4)	4.712	0.008	2.794	0.022
% Improvement	13.6	49.4	-	29.2
bMAC (4x8)	9.693	0.031	2	0.061
OzMAC (4x8)	8.3752	0.013	2.794	0.035
% Improvement	13.6	58.5	-	42.0
bMAC (8x8)	25.361	0.084	2	0.167
OzMAC (8x8)	19.996	0.025	4.76	0.120
% Improvement	21.2	69.7	-	28.0
bMAC (8x16)	45.282	0.177	2	0.355
OzMAC (8x16)	30.909	0.041	4.76	0.196
% Improvement	31.7	76.8	-	44.9
bMAC (16x16)	74.199	0.297	2	0.594
OzMAC (16x16)	60.608	0.065	9.28	0.601
% Improvement	18.3	78.2	-	-1.2



(a) Die Area (b) Power

Fig. 3: Die area and power costs vs precision configurations.

close to 14x, 20x and 20x increase in area, power and energy, respectively. Both 16x16 designs yield comparable energy while *Oz*MAC still possesses area and power benefits. This indicates going beyond 16-bits for *Oz*MAC is not beneficial.

From Fig. 3, area for *Oz*MAC and bMAC scale up in a similar fashion almost linearly with respect to product of weight and activation bits. However, *Oz*MAC's power consumption scales much better than that of bMAC, which incurs a much sharper increase with precision. Relative to bMAC, 8x16 *Oz*MAC delivers the most area benefit (32% improvement), whereas the mixed precision 4x8 and 8x16 *Oz*MAC designs offer the highest energy benefit up to 45%. Mixed precision designs deliver the highest energy improvements, as they can leverage the lower of the two precisions for *Oz*-encoding, incurring minimum latency (and thereby energy) overhead while taking advantage of the lower hardware complexity. Power benefits increase monotonically with precision due to the serial nature of *Oz* computation with signal transitions that get relatively sparser with higher precision.

Key Takeaway: *Oz*MAC is more area and power-efficient than bMAC across all precision configurations, and more energy efficient across all but one (16x16) configuration. Energy consumption for both designs evens out at 16-bit weight precision, beyond which *Oz*MAC becomes inefficient due to high latency overhead.

VI. Frequency Scaling Analysis

In this section, we evaluate two types of frequency scaling to assess the effects on PPA trends between *Oz*MAC and bMAC.

TABLE IV: TSMC N5 PPA for INT8 (8-bits) OzMAC across varying frequencies: 500 MHz, 1 GHz and 1.5 GHz.

MAC Hardware	Power (mW)	Latency (ns)	Energy (pJ)
bMAC (0.5 GHz)	0.084	2	0.167
OzMAC (0.5 GHz)	0.025	4.76	0.120
% Improvement	69.7	-	28.0
bMAC (1 GHz)	0.166	1	0.166
OzMAC (1 GHz)	0.050	2.38	0.118
% Improvement	70.1	-	28.7
bMAC (1.5 GHz)	0.251	0.667	0.167
OzMAC (1.5 GHz)	0.075	1.587	0.119
% Improvement	70.2	-	29.0

A. Iso-Frequency Evaluation

As can be seen from Table IV, *Oz*MAC consumes 50 μW power and 118 fJ energy at 1 GHz, and only 75 μW and 119 fJ even at 1.5 GHz. At all three frequencies, *Oz*MAC improves power and energy by almost 70% and 29% respectively. As expected, power consumption scales linearly with frequency and energy stays almost constant since power increases and latency (due to clock period) reduces by similar amounts.

TABLE V: TSMC N5 PPA for OzMAC and bMAC across varying bit precisions at throughput-matching frequencies.

MAC Hardware (wgt x act)	Freq GHz	Power (mW)	Latency (ns)	Energy (pJ)
bMAC (4x4)	0.5	0.015	2	0.031
OzMAC (4x4)	0.7	0.011	2	0.022
% Improvement	-	29.2	Equal	29.3
bMAC (4x8)	0.5	0.031	2	0.061
OzMAC (4x8)	0.7	0.018	2	0.036
% Improvement	-	41.5	Equal	41.6
bMAC (8x8)	0.5	0.084	2	0.167
OzMAC (8x8)	1.2	0.059	2	0.118
% Improvement	-	29.5	Equal	29.6
bMAC (8x16)	0.5	0.177	2	0.355
OzMAC (8x16)	1.2	0.096	2	0.192
% Improvement	-	46.0	Equal	46.0

B. Iso-Latency Evaluation

*Oz*MAC's area-power-energy improvements are achieved at the cost of increased latency (1.4x for 4 bits and 2.4x for 8 bits). Such *Oz*MAC designs are ideal for edge inference applications that can tolerate the slight increase in latency (and reduction in throughput) but with stringent area/power/energy constraints. Here, we show that *Oz*MAC can even be used effectively for higher throughput with higher clock frequency.

To bridge the latency gap between *Oz*MAC and bMAC, we can scale *Oz*MAC's frequency by the corresponding ratio to match bMAC's compute latency and throughput. Table V provides TSMC N5 PPA for bMAC (0.5 GHz) and *Oz*MAC at throughput-matching frequencies. 16x16 *Oz*MAC incurs 4.6x higher latency and hence is not considered here.

For the same throughput, INT4 (4x4) and INT8 (8x8) designs deliver close to 30% improvement in power/energy, while mixed precision designs (4x8 and 8x16) achieve even higher improvements in power/energy by up to 46%. Note that *Oz*MAC can potentially deliver even higher throughput than bMAC by leveraging the remaining headroom in power reduction (29% to 46%) to further increase the frequency.

Key Takeaway: *Oz*MAC maintains superiority in area, power and energy efficiency at frequencies ranging from 500 MHz to 1.5 GHz, and can leverage relative frequency scaling to achieve equal or higher throughput compared to bMAC without adversely affecting its power or energy efficiency.

VII. Conclusions

This paper presents rigorous industry standard evaluation of *Oz*MAC, an updated re-implementation of previously proposed Bit-Pragmatic (PRA) MAC design that performs a series of simple shift-and-add operations. It accounts for only the '1' bits in input binary value (skipping the '0' bits) thus leveraging bit sparsity in DL workloads. The main goal of this work is to assess practical commercial implementation potential of dynamic bit sparsity exploitation through such zero skipping MAC designs. We demonstrate the presence of high bit sparsity in eight state-of-the-art DL benchmarks. We implement a wide range of *Oz*MAC designs using commercial design tools and latest TSMC N5 process node, and obtain PPA results across various data precisions and clock frequencies. *Oz*MAC shows substantial improvements in all three metrics: area (up to 30%), power (up to 80%) and energy (up to 46%) relative to conventional binary bMAC. Finally, we demonstrate the significant power reduction of *Oz*MAC and how this can be leveraged to increase throughput by increasing frequency without compromising area and energy efficiency benefits. Future work will evaluate a large array of *Oz*MAC units in an actual DLA at the system level. We believe all DLAs targeting low precision inference should adopt *Oz*MAC design.

References

[1] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks," *Synthesis Lectures on Computer Architecture*, vol. 15, no. 2, pp. 1–341, 2020.

[2] A. Delmas Lascorz, P. Judd, D. M. Stuart, Z. Poulos, M. Mahmoud, S. Sharify, M. Nikolic, K. Siu, and A. Moshovos, "Bit-tactical: A software/hardware approach to exploiting value and bit sparsity in neural networks," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019, pp. 749–763.

[3] J. Albericio, A. Delmás, P. Judd, S. Sharify, G. O'Leary, R. Genov, and A. Moshovos, "Bit-pragmatic deep neural network computing," in *Proceedings of the 50th annual IEEE/ACM international symposium on microarchitecture*, 2017, pp. 382–394.

[4] P. Judd, J. Albericio, T. Hetherington, T. M. Aamodt, and A. Moshovos, "Stripes: Bit-serial deep neural network computing," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2016, pp. 1–12.

[5] H. Sharma, J. Park, N. Suda, L. Lai, B. Chau, V. Chandra, and H. Esmaeilzadeh, "Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018.

[6] N. Wang, J. Choi, and K. Gopalakrishnan, "8-bit precision for training deep learning systems," 2018.