IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 December 2023, accepted 19 December 2023, date of publication 26 December 2023,
date of current version 5 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347578

RESEARCH ARTICLE

Pianissimo: A Sub-mW Class DNN Accelerator With Progressively Adjustable Bit-Precision

JUNNOSUKE SUZUKI[1], (Graduate Student Member, IEEE), JAEHOON YU[1], (Member, IEEE), MARI YASUNAGA[1], (Graduate Student Member, IEEE), ÁNGEL LÓPEZ GARCÍA-ARIAS[1], (Graduate Student Member, IEEE), YASUYUKI OKOSHI[1], (Graduate Student Member, IEEE), SHUNGO KUMAZAWA[1], (Graduate Student Member, IEEE), KOTA ANDO[2], (Member, IEEE), KAZUSHI KAWAMURA[1], (Member, IEEE), THIEM VAN CHU[1], (Member, IEEE), AND MASATO MOTOMURA[1], (Fellow, IEEE)

[1]Tokyo Institute of Technology, Yokohama, Kanagawa 226-8502, Japan
[2]Faculty of Information Science and Technology, Hokkaido University, Sapporo, Hokkaido 060-0814, Japan

Corresponding author: Junnosuke Suzuki (suzuki.junnosuke@artic.iir.titech.ac.jp)

This work was supported in part by KAKENHI, Japan, under Grant 23H05489 and Grant 22KJ1348.

ABSTRACT With the widespread adoption of edge AI, the diversity of application requirements and fluctuating computational demands present significant challenges. Conventional accelerators suffer from increased memory footprints due to the need for multiple models to adapt to these varied requirements over time. In such dynamic edge conditions, it is crucial to accommodate these changing computational needs within strict memory and power constraints while maintaining the flexibility to support a wide range of applications. In response to these challenges, this article proposes a sub-mW class inference accelerator called Pianissimo that achieves competitive power efficiency while flexibly adapting to changing edge environment conditions at the architecture level. The heart of the design concept is a novel datapath architecture with a progressive bit-by-bit datapath. This unique datapath is augmented by software-hardware (SW-HW) cooperative control with a reduced instruction set computer processor and HW counters. The integrated SW-HW control enables adaptive inference schemes of adaptive/mixed precision and Block Skip, optimizing the balance between computational efficiency and accuracy. The 40 nm chip, with 1104 KB memory, dissipates 793-1032 μW at 0.7 V on MobileNetV1, achieving 0.49-1.25 TOPS/W at this ultra-low power range.

INDEX TERMS Ultra-low power, sub-mW, progressive bit-serial datapath, bit-scalable accelerators, adaptive inference, neural network, SW-HW cooperative control.

I. INTRODUCTION

Edge AI offers attractive advantages, including low latency, power efficiency, reduced bandwidth, and enhanced data privacy [1]. However, the high computational complexity of deep neural networks (DNNs) has been a significant obstacle to their deployment at the edge devices. To address this issue, researchers have employed optimization techniques such as quantization [2], [3], [4], [5], pruning [6], [7] and highly efficient model design [8], [9], [10], [11]. These methods reduce computational load and memory footprint, enabling the deployment of advanced DNNs on resource-constraints edge platforms.

The widespread adoption of edge AI has led to diverse application requirements, such as low power consumption, minimal latency, high accuracy, and high efficiency. In response, a broad scope of accelerators have been proposed, ranging from highly flexible and efficient accelerators [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] to operating at ultra-low power consumption [23], [24], [25],

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

[26], [27], [28]. Notably, bit-scalable accelerators accelerators [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] provide arithmetic designs that realize variable bit-precision at runtime, natively supporting mixed-precision operations. It is known that DNNs require different bit-precision levels at different layers [4], [5], such adaptability is crucial for enhancing efficiency.

However, endpoint edge devices present dynamic computational needs that vary both with environmental conditions and over time [29]. Such variability introduces three core challenges in edge environments. 1) Memory budget: Conventional DNN accelerators typically necessitate the use of multiple models to adapt to fluctuating computational demands over time. For instance, surveillance cameras may require different levels of computational resources based on variables like time of day and human activity. The need for such multiple models increases the memory footprint, a critical issue for edge AI systems operating under strict memory limitations. 2) Power constraints: Devices at the extreme edge, including surveillance cameras, have not only highly variable computational needs but also face stringent power restrictions. They have requirements for ultra-low power operation near sub-mW range [30]. 3) Operational flexibility: There is a need for the ability to efficiently process a wide range of neural network models, given the diversity of applications in versatile edge environments. In summary, edge AI systems require adaptive solutions that respond to changing resource demands while maintaining efficiency and minimizing memory and power consumption.

To address these issues, we propose Pianissimo [31], a sub-mW class inference accelerator with progressively adjustable bit-precision. Pianissimo is based on the concept of adjusting the computational complexity according to the inference difficulty: using more computation for complex tasks and less computation for easy tasks. Pianissimo mainly supports the following two features: 1) adaptive model switching to extract models of various bitwidth versions from a single model and avoid unnecessary computations in simple tasks; 2) dynamic processing control to process only the regions of interest (ROI) specified by the image sensor. The adaptive model switching is based on the authors' proposed ProgressiveNN [32], which extracts high bitwidth representations from a single model, reducing the computational complexity for simple tasks. Dynamic processing control reduces computational complexity by only processing the ROIs from the image sensor.

At the heart of our design is a novel datapath architecture with progressive bit-serial datapaths. Our proposed accelerator is distinct from previous flexible bit-precision accelerators [13], [14], [15], [16], [17], [18], [19] in terms of allowing low to high bitwidth representations with a single weight. By adopting a bitwise quantization representation and bit-serial accumulation scheme from the most significant bit (MSB) to the least significant bit (LSB), our design ensures high flexibility despite the simple circuit design. Additionally, our bit-serial datapath can be straightforwardly but efficiently extended to mixed precision. This accumulation scheme allows maximum utilization of the processing element (PE) while ensuring without reducing its functionality.

To further enhance the model-level efficiency, we support well-designed depthwise separable convolutional neural network (DSCNN)–based models [9], [10], [11], [33] with two different datapaths: depthwise (DW) and pointwise (PW) layers. DW and PW layers, introduced in MobileNet [8], have brought innovative lightweight to computational efficiency. By dividing the convolution layer into DW for spatial extraction and PW for channel extraction, MobileNet achieved significant improvements in both computational and parameter efficiency without compromising inference accuracy. Therefore, the lightweight advantage offered by these is crucial for edge AI with limited computation and memory resources. To efficiently process PW and DW layers, we employ two datapath designs to handle distinct DW and PW feature extraction dimensions [34]. Moreover, Pianissimo seamlessly handles the transposition of output feature maps to accommodate these varied processing orders. This transposition allows for layer-type-specific input data supply.

Progressive bit-serial DW/PW and block skipping processing are overseen by software-hardware (SW-HW) cooperative control using a reduced instruction set computer (RISC) processor and HW counter complex; the RISC also contributes to bit-serial PW/DW as well as dynamic processing skipping control using sensor information. Our work shows that the control scheme integrated with the RISC and the HW counters provides a solution that significantly increases the flexibility of the edge AI inference with a bit of power overhead under ultra-low power.

The remainder of this article is structured as follows. Section II introduces the recent bit-level flexible accelerators and ultra-low-power accelerators. Section III presents two core algorithms of the proposed accelerator, facilitating adaptive inference at the edge. Section IV introduces a sub-mW class inference accelerator called Pianissimo that features a progressive bit-by-bit datapath. Section V shows evaluation results using advanced small NNs. Finally, Section VI concludes this article.

II. RELATED WORKS

A. BIT-SERIAL/DECOMPOSED ACCELERATORS

Bit-serial computation straightforwardly provides flexibility for neural networks by its fineness. Stripes [13] provided accuracy and performance flexibility by computing a single input operand in a bit-serial manner. UNPU [16] offers a similar trajectory and improves area efficiency by incorporating lookup table-based PEs.

Differing from the fully bit-serial computation approach adopted by Stripes and UNPU, Bit Fusion [15] unveiled a fused-PE. This innovative design dynamically configures itself based on the bit-precision of the input operands. It achieves the mixed precision of power of two by spatially

distributing bits as 2-bit bricks and then merging them appropriately. The bit-serial scheme of Pianissimo stands out by storing all lower bitwidth values in a single weight value, expanding these values over time.

There is a growing interest in the utilization of bit-level sparsity for further optimization [14], [17], [18], [19], [20]; leveraging sparsity is highly efficient as it eliminates the need for superfluous zero operations in the datapath [35], [36], [37]. Bit-Pragmatic [14] introduced bit-level sparsity compression to the activation using bit-serial computation, significantly improving the computational efficiency of NNs. Bit-Tactical [17] presented value-level weight skipping in addition to using bit-level activation sparsity. Bit-Tactical also reduced job latency by allowing data movement to neighbor lanes to address the issue of the load imbalance caused by sensitive skipping processing. Laconic [18] introduced bit-level sparsity for both activation and weights. Bitlet [19] reduced the load imbalance problem by reducing the need for synchronization using bit-interleaved PE that treats skipping nonzero values irregularly aligned. Ristretto [20] enabled value and bit-level skipping of both weight and activations by streaming flattened bit-brick sequences with compression of nonzero bricks.

The sparsity utilization of bit-level and value-level suggests further efficiency gains in Pianissimo. However, it is also crucial to consider potential challenges. Introducing such fine-grained speedup might result in critical overhead to circuit area and power consumption matters, especially when operating within ultra-low-power conditions.

B. ULTRA-LOW POWER ML ACCELERATOR

The domain of ultra-low power machine learning (ML) accelerators presents a complex challenge: executing the desired NN process with minimal energy. Some accelerators have navigated this challenge in sub-mW/nW, specializing in tiny models for specific applications [23], [24], [28].

Shan et al. [23] has created a keyword spotting, including mel frequency cepstrum coefficients, chip fabricated with a 28 nm process, operating at a mere 510 nW. The NN core uses a binary DSCNN and a small register file-based memory block to achieve 94.6 % on a two-word classification task while minimizing data movement and computational cost. Lu et al. [24] proposed a 65 nm chip running at 184 μW with a slight two-layer edge CNN for hand motion detection and feature extraction. Kosuge et al. [28] achieves speech recognition of 35 keywords at 153 μW in a 40 nm process by fully unrolling the NNs in the circuit and reducing data movement to the limit. In addition, Kosuge et al. [28] employs the model pruned by more than 95 % to significantly reduce the circuit area while improving accuracy by training the activation functions.

In recent years, NN accelerators have been proposed that combine ultra-low power consumption and flexibility [25], [26], [27]; Pianissimo is one of these. Jokic et al. [25] proposes a face recognition system that combines CNN and the binary decision tree and reduces power consumption by running the power-hungry CNN core only when necessary. The CNN core supports 1-bit and 16-bit weights. Park et al. [27] achieve high-quality speech enhancement at 740 uW with band optimization that dynamically adjusts computational complexity based on the frequency band. Furthermore, the use of 4-bit logarithmic quantization allows for a PE that operates without the need for multipliers, and its PE supports DW and PW layers. TinyVers [26] is a highly flexible accelerator despite its ultra-low power. TinyVers' PE array (PEA) supports two datapaths, broadcast, and multicast of weights, with the RISC-V processor overseeing the entire process. The chip fabricated with 22 nm FDX with embedded magnetoresistive random access memory (eMRAM) runs on several types of NNs, including ResNet-8 of MLPerf Tiny [33], and achieves high tera operations per second per watt (TOPS/W) with ultra-low power consumption. Ont achieved using BS, measu supports DW layer dataflow, which is critical for edge AI, and provides adaptive bit-precision with a single weight. A detailed comparison between Pianissimo and these ultra-low power accelerators is presented in Section-V-E.

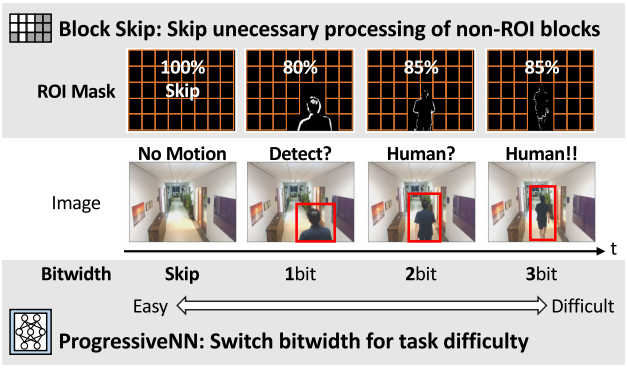
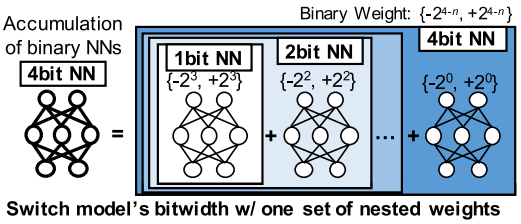


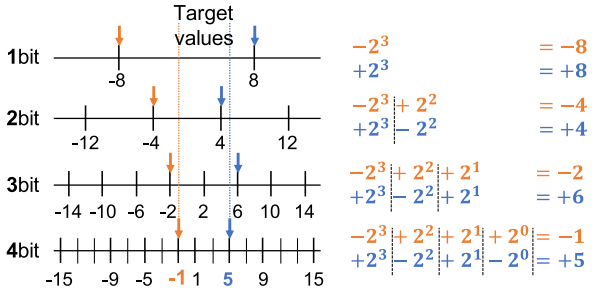
FIGURE 1. Adaptive adjustment of the accuracy-computation tradeoff based on ProgressiveNN and block skip.

III. ADAPTIVE ALGORITHMS BEHIND PIANISSIMO

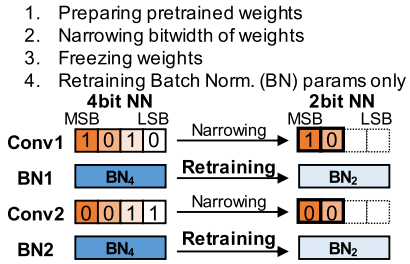
Pianissimo was designed for adaptive inference at extreme edge environments. To accomplish this, Pianissimo employs an adaptive adjustment between inference accuracy and computational complexity, as illustrated in FIGURE 1. This adjustment strategy is achieved through two essential model-level algorithms: ProgressiveNN and Block Skip (BS). Both algorithms excel in dealing with information available at edge environments, each offering a unique approach to adaptive computational complexity management. ProgressiveNN allows for model switching based on the difficulty of the input task as shown in FIGURE 1 bottom. BS focuses on trimming unnecessary computations outside the ROI, as shown in the top of FIGURE 1. Pianissimo leverages these adaptive processing algorithms to enable adaptive inference and improves efficiency for edge AI applications.



(a) Nested structure using a single set of the weights.



(b) Progressive numerical representation.



(c) ProgressiveNN training method.

FIGURE 2. **ProgressiveNN algorithm. (a) In ProgressiveNN, each weight digit represents either +1 or −1, and the accumulation from MSB to LSB yields a high bitwidth network. (b) As the bitwidth expands, quantized values asymptotically approach the intended target values.**

This section details these two algorithms, clarifying the core design concept behind Pianissimo.

A. PROGRESSIVENN

ProgressiveNN is a flexible bit-precision network that dynamically switches the bitwidth of NN weights according to the task difficulty. ProgressiveNN, proposed by the authors, features 1) bitwise numeric representation that allows MSB to LSB computation and 2) batch normalization (BN) retraining to improve accuracies of low-bitwdth models.

Whereas general ML accelerators employ fixed-point representation as their main computation scheme, Pianissimo adopts ProgressiveNN's bitwise binary quantization scheme. Each value is quantized bitwise, with each binary digit representing either +1 or −1. The value can therefore be seen as a set of decomposed binary values.

ProgressiveNN has a nested structure where the MSB is the outermost in the bitwise decomposed values, as shown in FIGURE 2(a). The main difference with other numerical representations is that this nested representation is processed in order from the outermost MSBs. In more detail,

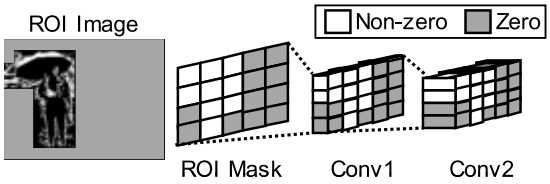


FIGURE 3. **Block skip algorithm using ROI masks obtained from the event-driven sensors.**

ProgressiveNN obtains target high bitwidth values by accumulating from MSB to LSB computation while considering its digit's place value. We describe this accumulation scheme using the fully-connected layer.

For the sake of simplicity, we explain this accumulation scheme using the fully-connected layer with C input channels. The j-th output z_j is described as follows

$$z_j = \sum_{i=1}^{C} w_{i,j}x_i + b_j = \sum_{i=1}^{C}\sum_{n=1}^{N} w_{i,j}[n]x_i \cdot 2^{n-1} + b_j \quad (1)$$

where $w_{i,j}[n] \in \{+1, -1\}$ is the n-th digit of N-bit weights from the i-th input neuron to the j-th output neuron, x_i is the i-th input activation, and b_j is the j-th bias. Recalling that the weights are computed from the upper bits, we notice that stopping the computation in the middle of a calculation can yield a lower bit value. If only the upper M-bits of the N-bit weights are used, (1) is described as follows

$$z_j = \sum_{i=1}^{C}\sum_{n=N-M+1}^{N} w_{i,j}[n]x_i \cdot 2^{n-1} + b_j. \quad (2)$$

Thus, only one set of N-bit weights is needed to achieve the desired smaller bitwidth weight. As shown in FIGURE 2(b) of the example using 4-bit values, increasing the bitwidth used in ProgressiveNN asymptotically close to the target values. However, when applied this bitwise representation directly to NNs, the accuracy is significantly degraded at low bitwidths because of changing distribution [38], [39]. To address this problem, ProgressiveNN restores accuracy by retraining the BN for each weight bitwidth sets while freezing the weight parameters, as shown in FIGURE 2(c).

B. BLOCK SKIP

BS is an integrated approach crafted to amplify computational efficiency by suppressing non-essential processing outside regions of interest (ROIs) designated from low-power event-driven sensors like one proposed in [40]. Its idea is based on the concept of elevating the efficiency of applications that seamlessly merge ultra-low-power edge AI with sensor technology. This integration aims to leverage the information harvested directly from endpoint sensors to optimize computational demands.

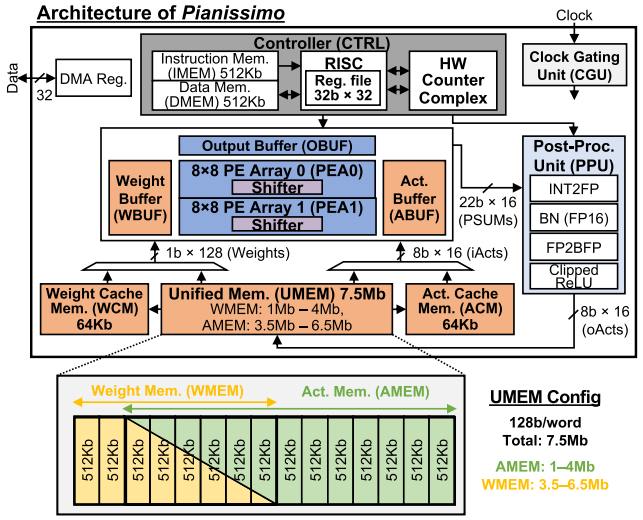
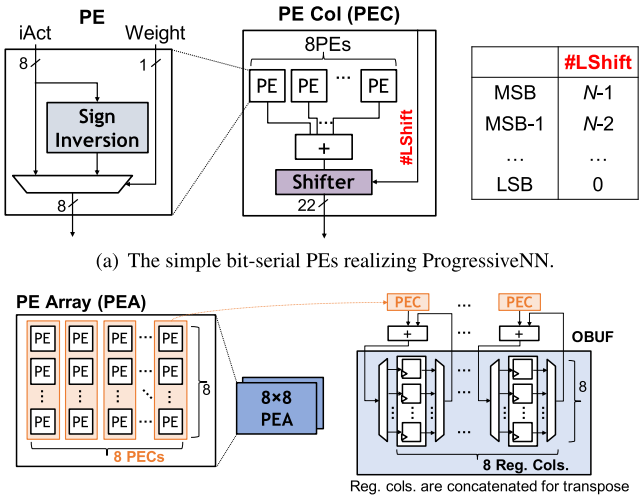


FIGURE 4. Pianissimo architecture overview and the UMEM configuration.



(a) The simple bit-serial PEs realizing ProgressiveNN.

(b) Two parallel PE arrays and accumulation in OBUF.

FIGURE 5. The core arithmetic unit. (a) ProgressiveNN is realizing with sign inversion and shift operations. (b) The register column corresponding to each PEC accumulates the partial sums.

When the sensor detects motion and identifies the region of interest, a binary mask is supplied to Pianissimo accelerator with the ROI position as one and the rest as zero. As illustrated in FIGURE 3, an inference procedure is then initiated using only this masked data, thereby excluding irrelevant areas and increasing efficiency. The figure shows that the grey block of the intermediate feature map indicates that specific operations are unnecessary and save power consumption by skipping computations. In processes that require a reduction of the intermediate feature map, such as downsampling processes, the ROI mask is consistently reduced and maintains its size with the changed data.

BS significantly advances in vision tasks with limited movement or narrow scope. One example is in the area of security cameras. These devices are often installed in zones where sporadic or slight motion is detected, and in such scenarios, these cameras often capture a little or small action from a wide angle of view. By employing BS in such situations, computational loads can be drastically reduced, realizing significant efficiency improvement.

This dynamic process control is conducted by a RISC processor, which oversees the processing of input segments and instructs where to use BS processin; dynamically adjusts processing based on ROI mask data to ensure that only essential data is processed; and ensures that the RISC processor is able to process the input segments in the same way as the RISC processor. Details of the RISC control are described in Section IV-C. RISC also has dedicated instructions for reducing the BS mask in downsampling.

IV. PIANISSIMO

We designed the inference accelerator called Pianissimo to realize ultra-low-power yet flexible inference at extreme edge circumstances. The key feature is the progressive bit-by-bit datapath, enabling ProgressiveNN inference and ensuring it meets various requirements in edge environments. Pianissimo incorporates dynamic model switching and dynamic BS processing via an integrated SW-HW control approach. In addition, Section IV-E describes two usecase-level algorithms using ProgressiveNN: adaptive precision (AP) and mixed precision (MP).

A. PIANISSIMO OVERALL ARCHITECTURE

FIGURE 4 displays the overall Pianissimo architecture that realizes the adaptive inference shown in FIGURE 1. The architecture mainly consists of five parts: unified memory (UMEM), PEA, post-processing unit (PPU), controller (CTRL), and clock gating unit (CGU).

For the datapath, a three-layer memory hierarchy–buffer (ABUF/WBUF), cache (ACM/WCM), and unified memory (UMEM)–is used to suppress the power consumption by maximizing data reuse and minimizing data movement. As shown in the bottom of FIGURE 4, UMEM is a configurable 7.5 Mb memory space for both weight and activation in which the sizes of weight memory (WMEM) and activation memory (AMEM) are 1-4 Mb, and 3.5- 6.5 Mb, respectively. Such adaptability is crucial to accommodate this study's diverse neural network (NN) demands on-chip. For instance, when dealing with significant weight parameters or extensive bitwidth, the approach is to increase the proportion of WMEM. Contrarily, when the activation is prominent, the balance of AMEM is increased to meet the various requirements.

The ACM and WCM adopt direct-mapped caches focusing on the spatial and temporal locality in NN inference. The 64 Kb ACM/WCM is more energy-efficient in terms of reading and writing operations compared to the more expansive 512 Kb UMEM (from memory specifications). Therefore, utilizing the cache becomes particularly beneficial when data is reused more than thrice. However, considering the power

consumption of transferring data from the UMEM to the ACM/WCM, bypassing and deactivating the ACM/WCM is more power-saving if reuse falls below this threshold. The 64 Kb memory used as a cache consumes 38% less power when reading data than the 512 Kb memory used in UMEM for memory used in our design. When data is accessed three times, it requires three cache reads and one UMEM read. This power consumption is lower than reading from the UMEM three times.

The controller consists of a customized RISC processor with dedicated instructions for BS and a counter complex for bit-serial multiply-accumulate (MAC) operations. The controller architecture includes both an instruction memory (IMEM) and a data memory (DMEM), each holding a memory capacity of 512 Kb. The IMEM is spacious enough to store the instructions required to execute NN models. Concurrently, the DMEM can keep all the BN parameters for high bitwidth sets on-chip. The cooperative control of the on-chip RISC processor and HW counters allows for flexibility and speed of model switching and ROI processing. The CGU governs the entire core to improve power efficiency further, reducing redundant power consumption.

Two 8×8 processing element arrays (PEA0 and PEA1) work jointly to fill the datapath pipeline according to the convolution mode and perform bit-serial MAC operations. The output buffer (OBUF) accumulates the output partial sums, then sends the accumulating results into the PPU after adequately transposing the output direction. OBUFs are double-buffered for efficient processing, ensuring a seamless PEA and PPU pipeline process.

The PPU processes three tasks: the BN process, the clipped rectified linear unit function, and conversions of the quantization format. The BN operation is executed in a 16-bit floating point (FP16) format to enhance inference accuracy, as mentioned in [41]. Therefore, the accumulation results in a 22-bit integer format are converted to FP16 and then converted to an 8-bit block floating point after the affine process, where a common 5-bit exponent is directed from the RISC. The 16 post-processed results are then packed and written to AMEM.

B. BIT-SERIAL PE AND PW/DW DATAFLOWS

As illustrated in FIGURE 5(a), ProgressiveNN is realized with a straightforward bit-serial PE. This PE is primarily composed of a sign-inverter and a shifter.

In Pianissimo, the weights, where each bit digit representing the binary value $\{-1, +1\}$, are fed in a bit-serial manner, transitioning from the MSB to the LSB. The sign inverter inverts the sign of the input activations according to the corresponding binary weights. Subsequently, the PE column (PEC), with 8 PEs, aggregates the partial sums of the PEs, and the shifter multiplies the place value given by its weight digit. In the case of N-bit weights, the amount of shifting is N-1 bits when processing MSB and 0 for LSB.

To put it simply, the ProgressiveNN PE is essentially the specialized PE for binary NNs [42], augmented with a shifter. Compared to a fixed-point MAC PE with an 8-bit multiplication and a 22-bit accumulation, a bit-serial MAC PE shows a circuit overhead of around 23% in simulation using Synopsys Design Compiler. For a fair comparison, the bit-serial PE includes eight 1-bit operations on 8-bit values and their addition instead of 8-bit multiplication. However, this approach offers the gain of progressive progressively adjustable bit-precision.

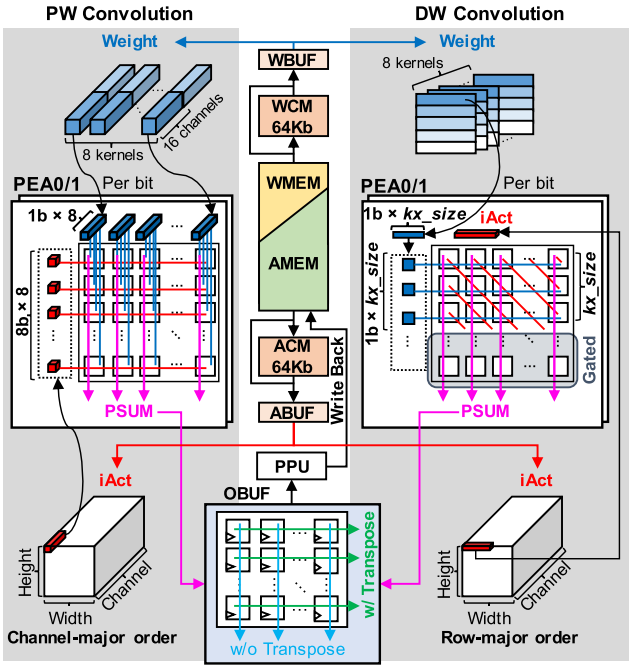
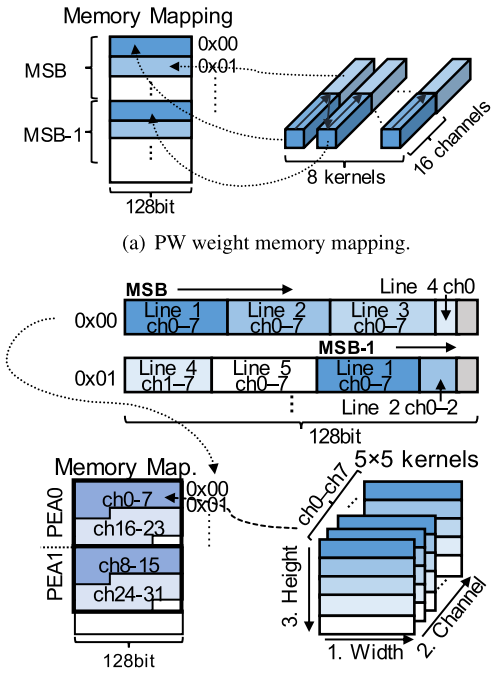


FIGURE 6. The bit-serial PW/DW dataflows. Weights are supplied bit-by-bit.

A noticeable distinction lies in the numerical representation. In general numerical expressions, such as fixed-point numerical expressions, the value is calculated from the LSB, which causes carry and an increase in the number of digits. However, the calculation can be interrupted by processing the weights bit-serially from the MSB. This unique approach facilitates the implementation of high bitwidth weights without compromising the efficiency and utilization of the PE.

A PEA consists of 8 PECs (8 × 8 PEs), as shown in FIGURE 5(b), and two PEAs operate in parallel. The OBUF also has eight register columns; one column consists of 8 registers, and each register column is tasked with accumulating the output of the corresponding PEC. The separate registers in the column are for accumulating the partial sums with the different output channels. The specific register to be utilized is determined based on the scheduled loop of the output channel. The details are described in Section IV-C. On the other hand, each register column is responsible for the accumulation of different output activation widths. These register columns are concatenated for the purpose of transposition of the output direction.



(a) PW weight memory mapping.

(b) DW weigth memory mapping.

(c) Bit-serial processing with minimised memory reads.

FIGURE 7. Memory mapping for bit-serial PW and DW weights to maximize data reuse.

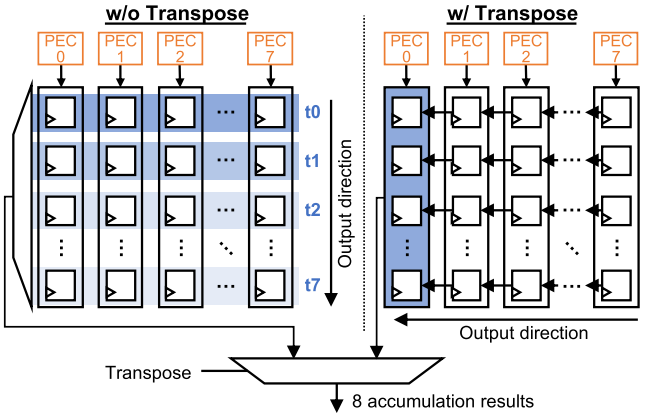


FIGURE 8. Transposition of the output direction in OBUF for a single PEA. 16 resulting outputs from two PEA are sent to the PPU in parallel.

Pianissimo supports both PW and DW layers, which are essential for inference at the edge. Each of these layers is executed via different datapaths to utilize the PEA computation resources fully. 2D convolution operations (Conv2D) can be sequentially processed as multiple PW operations. FIGURE 6 shows the bit-serial PW and DW dataflows. This figure also highlights a three-level memory hierarchy that supplies weights and activations (center of FIGURE 6).

In the PW mode, each PEA takes as input an 8-bit activation for each of the eight rows and a 1-bit weight for each of the 8 × 8 PEs. To provide input data to both PEAs without delay, the WMEM packs 128 sets of 1-bit weights from 16 channels in each of the eight kernels into a single word, and the AMEM packs 16 sets of 8-bit activations into a single word in channel-major order illustrated in FIGURE 7(a) left. To avoid unnecessary weight read at low bitwidth inference, each memory word stores only the weights of a particular digit FIGURE 7(c). Therefore, high bitwidth weights are read over time with a constant address stride according to the loop framework shown in Section IV-C and the number of memory reads is proportional to the required bitwidth.

In DW mode, the PEA handles 8-bit activation for every diagonal and 1-bit weights for each row. The requisite number of activations and weights fluctuates based on the kernel size and stride (FIGURE 6 right). The ABUF and WBUF adjust the input accordingly depending on the size, and the CGU deactivated any superfluous PEs to tackle this. Unlike in PW mode, the AMEM groups 16 sets of 8-bit activations into a single word, organized in a row-major order. As for weights, they are stored with a specific memory mapping, as illustrated in FIGURE 7(b) right, to suit the varying DW kernel sizes of 3, 5, and 7. This scheme sequences weights for the eight grouped kernels in the order of priority: row, channel, bit, and column. This structure is optimized to handle multiple kernel sizes and to ensure the better allocation of bit-serial weights to these sizes. The grey hatches indicate zero padding. In DW mode, eight kernel rows are utilized simultaneously, making the memory readout design emphasize row readout efficiency.

Finally, OBUF accumulates the resulting partial sums from both modes, transposes the output direction on demand, and passes them to PPU. The details of the OBUF transpose for a single PEA are illustrated in FIGURE 8. In scenarios without the transpose operation, the accumulation results are read vertically from the register column during each cycle, from time t0 to t7, with the specific output determined by a multiplexer. In contrast, when the transpose function is activated, the accumulated result is solely read from register column 0 throughout the output phase. During each cycle in this mode, data is read horizontally from each register column. Each register column then transfers its own data to the adjacent column on the left. The output direction is selected by the transpose flag managed by the RISC

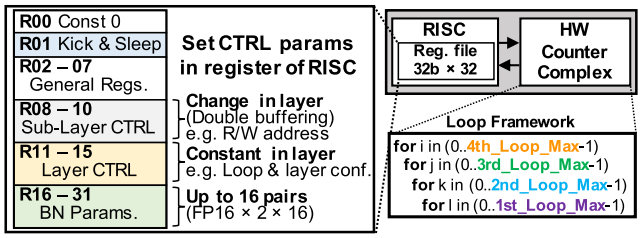


FIGURE 9. The RISC processor and HW counter complex.

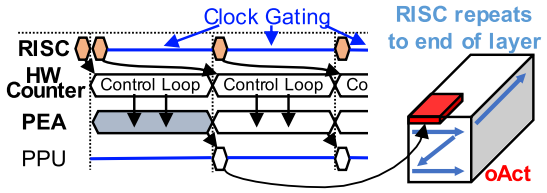


FIGURE 10. Timing chart for the control using the RISC and HW counters. THe RISC is gated after the parameter set and become active after HW loop is completed.

processor, resulting in 8 accumulative results from one PEA being sent to the PPU. Therefore, the PPU handles 16 outputs from 2 PEAs in parallel.

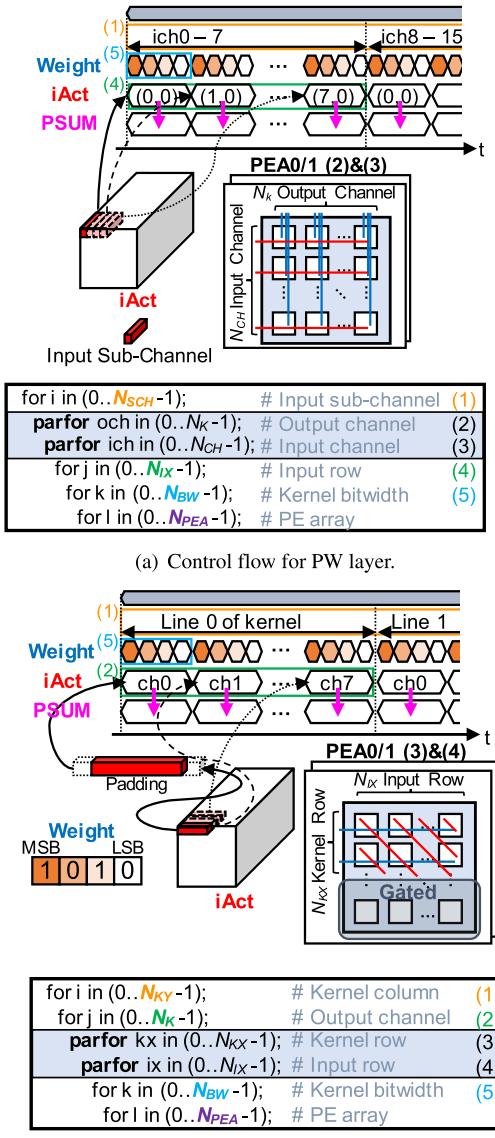
C. SW-HW COOPERATIVE CONTROL

FIGURE 9 shows the SW-HW cooperative control by the customized RISC processor and the HW counter complex. The RISC is equipped with 32 registers, each 32-bit. Register R00 is the zero constant register, and R01 is the special purpose register (SPR) for the core kick and RISC sleep flag. Registers R02 to R07 are general-purpose registers. The remaining registers, R08 to R10, R11 to R15, and R16-R31, are specialized for sublayer control, layer control, and batch normalization parameters, respectively.

The HW counter implements a quadruple nested loop that the RISC processor orchestrates. This loop configuration allows pianissimo the flexibility to operate in different modes, enabling it to switch between PW and DW modes and adaptively select the bitwidth for bit-serial weights.

FIGURE 10 shows a timing chart of the RISC processor and the HW counter complex. Initially, the RISC sets up the control information and parameters necessary for subsequent core processing in the background of the core processing. It triggers the beginning of the RISC operations and subsequently goes into a deactivated state managed by the CGU. Once kicked off, the nested loops of the HW counter start to move, and the PEA follows the loops and operates the processing instructed by the RISC. After all loop processing is completed, the red chunk in FIGURE 10 is obtained, and control is returned to the RISC. Once the output chunk is obtained, the RISC repeats the process until the end of the layer. The RISC performs BS by checking the ROI mask and skipping this chunk generation step.

FIGURE 11 shows a bit-serial PW/DW processing flow with 4-bit weights. The loops with grey hatches are processed



(a) Control flow for PW layer.

(b) Control flow fo DW layer.

FIGURE 11. DW/PW Control flows using 4-bit weights. The gray hatches in loop frameworks indicate parallel processing in PEA.

in parallel in the PEAs. In PW mode, as shown in FIGURE 11, both the input data and weights are supplied in channel-major order, with weights being supplied bit-by-bit. The innermost loop of PEA is the output channel in consecutive PW layers and row direction in the PW-DW layer. For example, in a consecutive PW, PEA0 treats the 0–7 output channels, and PEA1 is 8–15 output channels. Outside the weight loop is an input row loop of up to 8, where the weights are reused in the time direction using WBUF. The outermost processes the input sub-channel loop, completing the inner product operation.

DW control flow is shown in FIGURE 11. The difference between PW and DW in terms of kernel geometry occurs in the first and third loops from the outside. In the case of PW, it is the kernel's horizontal and vertical loops, whereas in

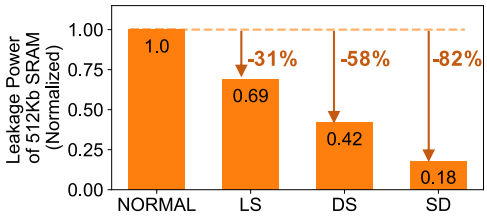
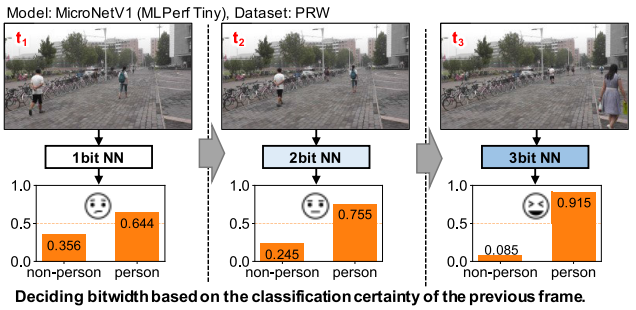
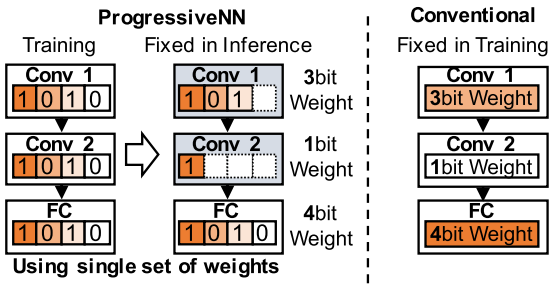


FIGURE 12. Leakage power consumption with 512 Kb UMEM's power management modes. The vertical axis is normalized with reference to the normal mode.



(a) Adaptive precision.



(b) Mixed precision.

FIGURE 13. Efficiency improvement with AP/MP of ProgressiveNN. (a) AP adjusts the bitwidth based on the previous classification confidence (entropy). (b) MP decides bitwidth pairs at the inference time.

DW, it is the horizontal and vertical loops of the kernel. The difference is due to the difference in kernel shape between PW and DW. In DW mode, the innermost loop of PEA is the output channel loop. Additionally, ABUF autonomously manages padding in the width direction, while padding in the height direction is handled through RISC-controlled processing skipping.

D. POWER MANAGEMENT

Minimizing power usage is crucial for achieving ultra-low power operation. Pianissimo adopts a fine-grained gating strategy using the CGU, which turns off the clock input to idle registers on pipeline. The gating is applied to various components, such as UMEM I/O, WBUF/ABUF, PEA/OBUF, PPU, and the RISC processor, but DMA peripherals and CTRL units except for the RISC processor. As partially depicted in the figure (see FIGURE 10), both the RISC processor and the PPU have shorter execution times compared to the PEA, making them ideal candidates for significant power savings through CGU. Moreover, in the PEA, idle PEs are actively gated, particularly in cases involving DW layers with small kernel sizes, layers with small input feature maps, or layers not extra-allocated to a PEA's size.

UMEM and WCM/ACM incorporate three distinct power management strategies: light sleep (LS), deep sleep (DS), and shutdown (SD). As shown in FIGURE 12, these power-saving modes significantly reduce power consumption compared to the normal operating mode. Specifically, LS, DS, and SD cut leakage power by 31 %, 58 %, and 82 %, respectively. Note that the power consumption is normalized in the figure. These modes are offered as a function of memory.

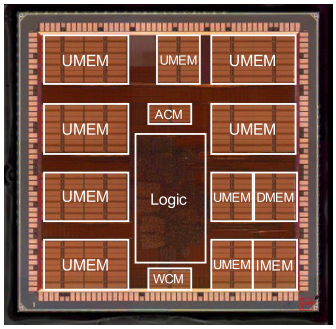
LS is designed for modest but immediate power savings and is dynamically applied throughout the inference process. DS is employed for memory components that are temporarily inactive, offering more substantial power reductions at the cost of longer resumption times. The SD mode is initiated for unused memory spaces during the inference process. While UMEM is designed with a comparatively larger memory space to accommodate a diverse range of models, its energy efficiency is optimized by using the SD mode adequately, thus keeping the power overhead to a minimum, even when executing smaller models with small memory requirements. In our design, since SD remains unchanged throughout the execution, it is directly managed externally through flags in the control register for DMA. On the other hand, DS and LS flags are overseen by the RISC to control flexibly.

E. ADAPTIVE/MIXED PRECISION USING PROGRESSIVENN

This section introduces AP and MP, the usecase-level algorithms using ProgressiveNN that Pianissimo employs to enhance the efficiency. Through these strategies, Pianissimo dynamically adjusts the bitwidth associated with weights according to the various computational requirements.

AP is the strategy for continuous time series data, where the current classification result determines the bitwidth for processing the next data FIGURE 13(a). The bitwidth switch is based on the classification confidence level, which indicates how dominant the probability of the inferred class is compared to the probabilities of other classes [32]. The confidence level is defined as the entropy, which is the amount of information that an external processor should calculate. When the confidence exceeds a threshold, a strategy is taken to either narrow the bitwidth or maintain the current bitwidth, depending on the specific requirements of the task. The confidence can depend on both the weight precision and the image-specific features.

MP is another strategy that employs different bitwidths for different layers, based on the fact that different layers require different levels of computational accuracy FIGURE 13(b). It is worth noting that the MP implementation of ProgressiveNN differs from traditional methods in that it uses only a single set of weights. This means that no additional memory cost is required to realize multiple MP weight sets. As a result,



Technology	TSMC 40nm CMOS (ULP)
Package	QFN80
Chip Size	3mm × 3mm
Core Area	SRAM: 4.51mm^2 Logic: 0.41mm^2
Core V_{DD}	0.7 – 0.9V
I/O V_{DD}	3.3V
Max Freqency	80MHz@0.9V 20MHz@0.7V
Gate Count	584K Gates
SRAM	UMEM: 7.5Mb - WMEM: 1 – 4Mb - AMEM: 3.5 – 6.5Mb DMEM/IMEM: 512Kb WCM/ACM: 64Kb

FIGURE 14. Pianissimo chip microphotograph and its specification.

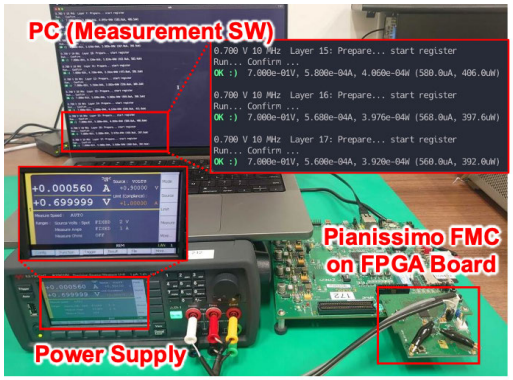


FIGURE 15. Evaluation environment for Pianissimo.

Pianissimo can seamlessly continue inference with only a single set of weights even if the bitwidth set changes during the inference process. Thus, Pianissimo's basic strategy is to use AP and MP together in order to maximize efficiency.

V. MEASUREMENT RESULTS

This section reports the Pianissimo accelerator's competitive performance within the ultra-low power domain. Furthermore, Pianissimo demonstrates its versatility by facilitating flexible inference across a wide range of NNs [9], [10], [11], [33]. Crucially, for all our evaluations, after the input data and the network model are loaded into the UMEM, the inference is carried out seamlessly until the end, eliminating the need for data transfers to external memory during inference. Also, we employed an 8-bit quantization for activations throughout the evaluations, and a multiply-accumulate operation is counted as two operations.

Section-V-A introduces the microphotograph and specifications of the fabricated chip. Section-V-B provides a power consumption analysis and highlights that Pianissimo operates in the sub-mW range. Section-IV-E observes the tradeoffs when applying AP/MP, exploring the potential for adaptive inference at the edge. Section-V-D examines the performance impact of using BS and demonstrates the potential for substantial performance gains. Finally, with comprehensive evaluations, Section-V-E compares Pianissimo with recent ultra-low power ML accelerators.

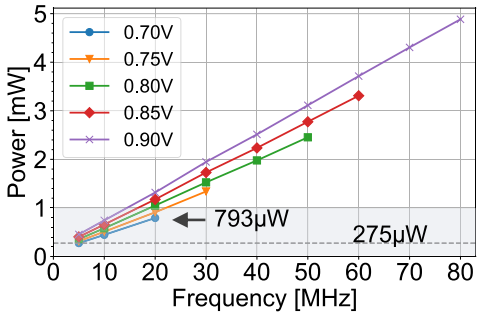


FIGURE 16. Power consumption vs. frequency with the operational voltages from 0.7 V to 0.9 V. The gray hatch indicates the sub-mW region, and the measured model is 4-bit MobileNetV1 0.25× (MLPerf Tiny).

A. CHIP IMPLEMENTATION AND EVALUATION ENVIRONMENT

This section reports the measured results of the Pianissimo fabricated on a 9 mm^2 die using TSMC 40 nm CMOS (ULP) technology. FIGURE 14 includes a chip microphotograph and a specification table. The core logic area occupies 4.92 mm^2, with memory components occupying 92 % of this space. UMEMs dedicated to AMEM or WMEM are strategically located close to their respective ACM or WCM, respectively, and switchable UMEMs are placed near both caches. The core logic is placed between ACM and WCM to minimize routing delays. Twenty chips were produced, with slight variation. The results of one of them are reported in this article.

We verified Pianissimo behavior using Verilog HDL and ModelSim simulator of version 2019.4. Pianissimo is implemented in 68,283 lines of Verilog HDL codes, partly expanded with the Ruby language.

FIGURE 15 shows evaluation environment for Pianissimo. A Pianissimo chip mounted on a field programmable gate array (FPGA) mezzanine card (FMC) connects to a ZC702 FPGA board that handles the input/output data. The PC controls the power supply unit and the FMC's clock generator via LAN to sweep the voltage and operation frequency. For evaluation, the PC transfers the test data to Pianissimo through FPGA, and the resulting outputs are transferred back and verified to the expected values. The recorded power measurement is related solely to the core and does not account for external memory accesses.

B. POWER CONSUMPTION ANALYSIS

FIGURE 16 depicts the power consumption trends across operational voltages ranging from 0.7 V to 0.9 V, taking into account varying clock frequencies. The observed power consumption oscillates between 275 μW and 5 mW, with the frequency spanning from 5 MHz to 80 MHz. Notably, for frequencies under 10 MHz, power usage remains below 1 mW across all voltage levels. Furthermore, this sub-mW consumption is also attainable at 20 MHz when operating at 0.7 V and 0.75 V. Pianissimo achieves ultra-low-power inference, registering power consumptions of 275 μW at 5 MHz and 793 μW at 20 MHz.

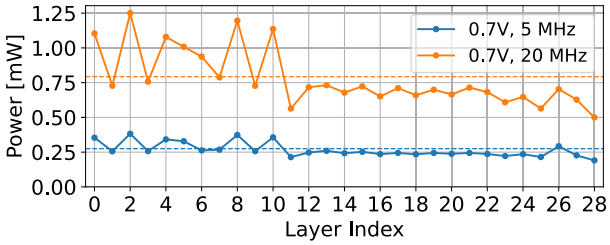


FIGURE 17. Power consumption in each layer of MobileNetV1 (MLPerf Tiny). The doted lines indicate the average power (see FIGURE 16).

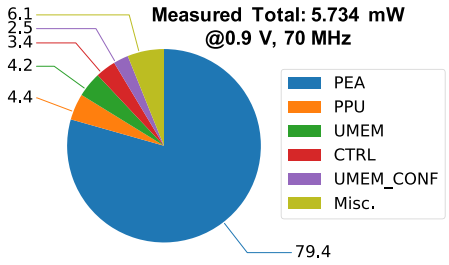


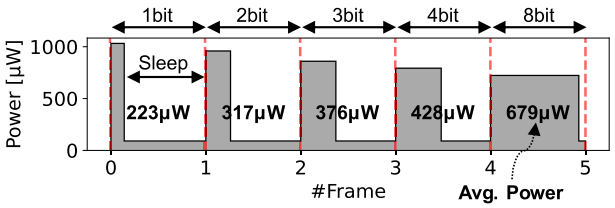
FIGURE 18. Power breakdown using 4-bit PW layer. The total power consumption is 5.734 mW at 0.9 V and 70 MHz.

FIGURE 17 depicts the power in each layer at 5 MHz and 20 MHz at 0.7 V in FIGURE 16. The dotted lines represent the average power of 275 μW and 793 μW, respectively. Since the odd-numbered layers are DW layers, they typically consume less power than the even layers. This power consumption reduction is attributed to the smaller DW kernel size in MobileNetV1, leading to some PEs being gated when operating in DW mode. The power is reduced in the latter half layers, mainly due to the smaller size of the input feature map, with some PEs being gated.

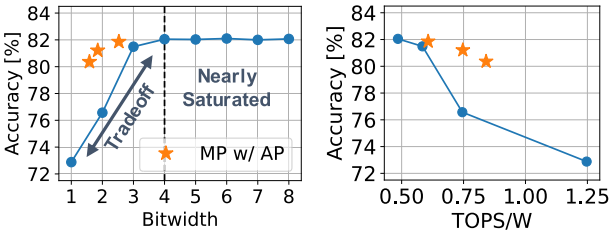
For further power usage analysis in Pianissimo, FIGURE 18 presents a detailed power consumption breakdown when operating with a 4-bit PW layer at 0.9 V and 70 MHz. This breakdown organizes power usage among five primary components: PEA, PPU, UMEM, CTRL, and UMEM CONF. Here, UMEM CONF plays a key role in managing the UMEM configuration, thereby facilitating configurable memory space. Interestingly, PEAs emerge as the most power-intensive, accounting for 79.4 % of total consumption. In contrast, memory's share was only 4.2 % despite typically being a significant power consumption. This efficiency is largely attributed to the data management within the three levels of the memory hierarchy, despite when working with models that inherently offer limited data reuse. Furthermore, the CTRL module, containing the RISC processor and the HW counter complex, consumes less than 3.4 % of the power, ensuring flexible control with minimal overhead.

C. POWER AND ACCURACY OF AP/MP

We analyzed power consumption in light of dynamic bitwidth variations using AP with MobileNetV1 0.25× (MLPerf Tiny benchmark [11]). The inference was recorded at a fixed 12 frame per second (FPS) at 0.7 V, with



(a) Power in each bitwidth at 12 FPS and 0.7 V.



(b) The tradeoff between bitwidth, accuracy, and TOPS/W.

FIGURE 19. AP/MP evaluation using MobileNetV1 (MLPerf Tiny): (a) Power vs. bitwidth. (b) AP/MP accuracy vs. bitwidth (left), and accuracy vs. TOPS/W (right). Three orange stars is the AP accuracy combined with MP.

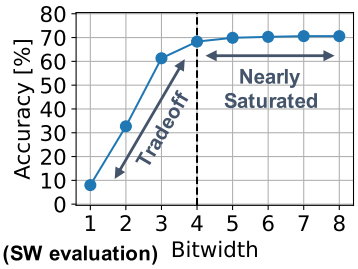


FIGURE 20. AP accuracy using MobileNetV2 1.0× on ImageNet dataset. Note that this assessment solely focuses on SW evaluation.

the power consumption during idle states also taken into account. As illustrated in FIGURE 19(a), the lower bitwidths resulted in shorter inference execution times at fixed FPS, subsequently reducing the average power consumption. Importantly, across all bitwidths from 1-bit to 8-bit, Pianissimo consistently maintained sub-mW power levels during inferences. Peak power consumption escalated at narrower bitwidths, primarily because the constant power usage of the PPU became more dominant.

For a comprehensive analysis, we further investigated the relationship among bitwidth, accuracy, and energy efficiency using the same MobileNetV1 model. The left segment of FIGURE 19(b) indicates that the tradeoff between accuracy and bitwidth is most prominent between 1-bit and 3-bit, nearly saturating beyond 4-bit. The accuracy of AP falls into 72 % at 1-bit. Nevertheless, our findings confirm that the combination of AP and MP considerably improves this tradeoff. As represented by the three distinct orange stars, the combination of MP and AP delivers accuracy levels comparable to the 4-bit to 8-bit model on an average of 2-bit. The practical results are also obtained on ImageNet dataset, as shown in FIGURE 20. It should be noted that these

TABLE 1. DMEM requirement for multiple sets of the BN parameters of the two large models in our evaluation.

	MobileNetV1 [33]		VWW'19 Champ [10]	
	1-set	4-set (1–4bit)	1-set	4-set (1–4bit)
Usage (Kb)	85.5	342	125	500
Rate (%)	16	65	24	98

TABLE 2. Summary of the measurement results using advanced tiny NNs. (b) indicates the weight bitwidth.

20MHz@0.7V 1MAC=2OPS	Dataset	(b)	Acc (%)	TOPS/W	GOP	Infer/sec
MobileNetV2 0.5×	CIFAR-100	1 4	41.6 66.3	2.01 0.53	1.81 0.58	33.1 10.6
MobileNetV1 (MLPerf Tiny)	VWW 96×96	1 4	72.2 81.7	1.25 0.49	1.29 0.38	83.3 24.9
VWW Challenge'19 Champion [10]	Edge VWW 320×240	1 4	72.5 83.4	1.54 0.62	1.99 0.61	13.7 4.15
MicroNet VWW-2 [11]	VWW 50×50	8	78.2	0.20	0.15	28.0
MicorNet AD-S [11]	MIMII [45]	8	92.8 (AUC)	0.36	0.28	7.35

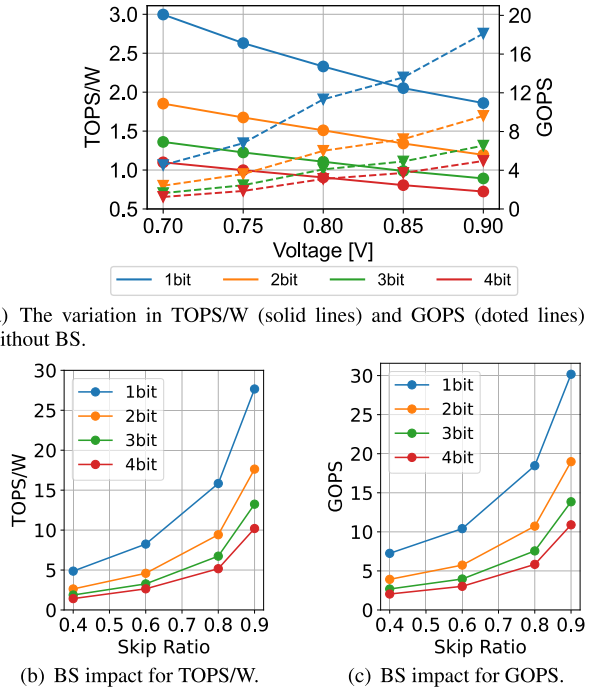
bitwidth allocations for layers were determined empirically, suggesting potential for further efficiency improvements by using neural architecture search algorithms like those proposed in [10], [43], and [44].

The right section of FIGURE 19(b) shows the relationship between AP/MP accuracy and TOPS/W, where the vertical axis is consistent with the left figure. The combination of MP and AP outperforms AP-only configurations in delivering higher accuracy at similar TOPS/W levels, indicating a more favorable tradeoff. The TOPS/W of an MP approximately follows the same trajectory as an AP with the same average bitwidth. A proportional relationship also exists between the average bitwidth and execution time.

It should be highlighted that both AP with MP and AP-only make use of the same weight sets. This implies that Pianissimo can handle MP and AP variations without requiring additional weight parameters. The only extra overhead comes from BN parameters to improve accuracy, but their memory footprint is significantly smaller compared to the NN weights. In Pianissimo, the DMEM storing the BN parameters has enough memory space to hold multiple sets of parameters. TABLE 1 shows DMEM requirements and utilization for two large evaluation models. From the table, we can confirm that four sets of BN parameters of 1–4 bit with accuracy-computation tradeoff can be held.

D. IMPACT OF BLOCK SKIP

In this section, we report on the influence of BS on performance metrics. The performance derived from BS depends on the size of the input feature map and the model. Therefore, our evaluation targeted standard 3 × 3.2D convolutional layers (Conv2D) equipped with 32 input and output channels. For the ROI image, we used bounding boxes with the motion labels, such as creatures and vehicles, from the Microsoft COCO dataset [46] to define the ROI region.



(a) The variation in TOPS/W (solid lines) and GOPS (doted lines) without BS.

(b) BS impact for TOPS/W.

(c) BS impact for GOPS.

FIGURE 21. Efficiency improvement of BS in typical Conv2d layer at 20 MHz and 0.7 V: (a) without BS and (b)(c) with BS. Each color consistently represents the bitwidth variation.

FIGURE 21(a) illustrates the variation in energy efficiency and peak performance across different weight bitwidths without BS. The solid lines depict energy efficiency (TOPS/W) mapped on the left vertical axis, whereas the dotted lines indicate peak performance (GOPS) on the right vertical axis. Each color variation corresponds to a different bitwidth. Evaluation results show that energy efficiency spans 0.7 to 1.1 TOPS/W at 4-bit, extending from 1.8 to 3.0 TOPS/W at 1-bit. Maximum efficiencies were consistently achieved at 0.7 V across various bitwidths. Peak performance ranged from 4.6 to 1.2 GOPS for 4-bit and 18.1 to 4.5 GOPS for 1-bit. GOPS and TOPS/W are calculated as follows: $\text{GOPS} = \frac{\text{OPS}\times 2\times \text{frequency}}{\text{cycles}} \times 10^{-9}$, $\text{TOPS/W} = \frac{\text{OPS}\times 2\times \text{frequency}}{\text{cycles}\times \text{power}} \times 10^{-12}$, where cycles are the number of cycles per inference and are from ModelSim simulation. For instance, TOPS/W of 1-bit Conv2D layer at 20 MHz and 0.7 V is calculated as $\frac{(44\times 10^6)\times 2\times (20\times 10^6)}{(391\times 10^3)\times 0.0015} \times 10^{-12} \simeq 3.0$.

FIGURE 21(b) and FIGURE 21(c) show the efficiency improvement achieved using BS, measured at 20 MHz and 0.7 V. The horizontal axis indicates the skip ratio; the higher percentage corresponds to the smaller ROI area within the image. In the scenario without BS, a 1-bit Conv2D registers a performance of only 3.0-1.8 TOPS/W at voltages between 0.7-0.9 V (see FIGURE 21(a)). With the application of BS, this range is boosted into a span between 27.7-10.2 TOPS/W. Notably, at a skip ratio of 0.9, BS provides a significant efficiency improvement, resulting in an enhancement of roughly 9.2× compared to the scenario excluding BS. The

TABLE 3. List of measurement results for the two representative evaluation models.

Model	Dataset	Weight Bitwidth	Accuracy (%)	20MHz@0.7V			80MHz@0.9V		
				mW	TOPS/W	GOPS	mW	TOPS/W	GOPS
MobileNetV1 (MLPerf Tiny)	VWW 96×96	1	72.2	1.032	1.247	1.287	6.380	0.807	5.148
		2	78.0	0.959	0.745	0.714	5.930	0.482	2.856
		3	80.6	0.860	0.584	0.502	5.193	0.387	2.010
		4	81.7	0.793	0.485	0.385	4.884	0.315	1.539
		8	81.9	0.723	0.274	0.198	4.380	0.181	0.794
VWW Challenge'19 Champion	Edge VWW 320×240	1	72.5	1.295	1.537	1.990	8.003	0.995	7.961
		2	78.3	1.129	1.011	1.141	7.081	0.645	5.466
		3	82.4	1.004	0.787	0.791	6.301	0.502	3.162
		4	83.4	0.974	0.621	0.605	6.024	0.402	2.419

effectiveness of BS is especially pronounced in the initial layers, attributed to their expansive input feature map sizes.

E. OVERALL EVALUATIONS AND COMPARISON

TABLE 2 summarizes the overall results at 20 MHz and 0.7 V from five modern tiny network models: MobileNetV2 [9], MobileNetV1 [33], the Visual Wake Words (VWW) [47] challenge 2019 champion model [10], and two MicroNet variants [11]. Note that MicroNet was used for the 8-bit model in accordance with the original paper. In addition, we evaluated a classification task using edge images to investigate the further possibility of data available at the edge environment, considering a potential integration with event vision sensors such as one proposed in [48]. For this purpose, we created and evaluated the edge VWW dataset using the edge extraction technique described in [49].

These comprehensive results highlight the capability of Pianissimo to provide practical inference speeds (inference/sec) throughout all model variations, including the 1-bit to 8-bit models. Inference/sec is calculated as $\frac{\text{frequency}}{\text{cycles}}$, where the number of cycles is obtained from ModelSim simulation. Since external memory accesses during inference are limited to input images and output results, these transfer times are negligible compared to the overall execution time. The results show competitive performance in the general image classification tasks, such as CIFAR-100 and VWW dataset, with results like 66.3 % accuracy on CIFAR-100. We also achieve an accuracy of 83.4 % for VWW with only contour edges. Furthermore, it performs an accuracy of 81.7 % at 24.9 FPS ($\frac{20\times10^6[\text{Hz}]}{803\times10^3[\text{cycles}]}$), consuming just 793 μW using 4-bit MobileNetV1. Additionally, Pianissimo has shown practical results with an AUC score of 92 % and throughput of 7.35 FPS in anomaly detection using 8-bit MicroNet and the MIMII dataset [45] of toy sound. This underscores that the utility of Pianissimo is not limited to visual tasks but can be applied to a broader range of applications.

TABLE 3 lists the measurement results for the two main evaluation models. As mentioned in Section III-A, the observation that power consumption decreases as bitwidth increases is also confirmed at the model-level analysis. When operating at 20 MHz and 0.7 V, the models work around 1 mW. When the conditions are adjusted to 80 MHz and 0.9 V, they work in the low-power range, staying below 10 mW. In addition, at these settings, peak performances of 5.148 GOPS and 7.961 GOPS were registered at 80 MHz. The table suggests that the models deliver competitive performance, even when accounting for their relatively modest levels of parallelism. In summary, Pianissimo ensures the practical inference capability in the wide range of NNs under the condition of ultra-low power.

TABLE 4 compares pianissimo with recent ultra-low-power inference accelerators [24], [25], [26], [27], [28]. Since the 4-bit weight model offers adequate precision (see 19(b)), we use this weight model as our evaluation standard. Pianissimo exclusively supports 8-bit activation to ensure sufficient accuracy. Typically, TOPS/W and GOPS have an inverse proportionality with bitwidth of both weight and activation.

While the accelerators proposed in [24] and [28], operate with ultra-low power consumption, they are confined to specific NN models, limiting their applicability across diverse edge environments. Similarly, CNN core in [25] can handle mixed 1-bit and 16-bit precision, but the minor impact on peak performance suggests its implementation lacks efficiency. [27] The speech enhancement accelerator presented in [27] supports DSCNNs and optimizes its computational complexity based on the frequency band. However, its range of supported and applicable applications is narrow. Unlike Pianissimo, it lacks a flexible control mechanism, such as RISC, to optimize power consumption.

TinyVers [26] stands out for its commendable efficiency across several models within the ultra-low power spectrum but presents certain limitations. Particularly, its adaptability leaves room for improvement. Notably, TinyVers does not accommodate the parameter-efficient DSCNNs. Also, a clear gap exists between its performance and theoretical efficiency when downscaling both weights and activations from 8-bit to 2-bit. Instead of achieving the ideal 16× efficiency improvement, it reaches only 4.8×. This disparity arises from TinyVers' approach to mixed precision: it gates parts of the PE. This reveals suboptimal support for mixed precision in its design. Note that TinyVers utilizes a more advanced 22 nm FDX process, incorporates an eMRAM technology,

TABLE 4. Comparison with the recent ultra-low power ML accelerators.

	ISSCC'21 [24]	VLSI'21 [25]	VLSI'22 [26] (TinyVers)	ISSCC'23 [27]	VLSI'23 [28]	This work (Pianissimo)
Technology (nm)	65	22	22 FDX	28	40	40 (ULP)
Supported ML	CNN, BDT	CNN, BDT	CNN, FC/RNN, GAN, AE, TCN, SVM	DSCNN[1], GRU[2], FC	CNN	DSCNN[1], CNN, FC/RNN
MLPerf Tiny	N/A	N/A	ResNet-8	N/A	N/A	MobileNetV1 0.25×
Flexibility	None	MP	MP, RISC-V	Band Optim.	None	AP/MP, BS, RISC
NN Quant. (b)	A/W: 4, 6	A: 16, W: 1, 16	A/W: 2, 4, 8	A: 8, W: 4	A: 14, W: 1	A: 8, W: 1-8
On-Chip Memory Size	1.8KB	1.2MB SRAM	SRAM 132KB (L1) 64-512KB (L2) eMRAM 512KB	35 kB	0 KB	1104 KB SRAM
Die/Core (mm^2)	1.5 / 0.276	3.42 / 2.10	6.25 / N/A	0.81 / N/A	9 / 7.63	9 / 4.92
Voltage (V)	0.6-1.2	0.65 (+/-10%)	0.4-0.9	0.8-1.1	0.5-1.1	0.7-0.9
Freq. (MHz)	25	125-265	0.033-150	2.5	0.02-0.12	5-80
Peak Performance (GOPS)	N/A	5.12 (16/16) 5.76 (16/1)	17.6 (8/8)	N/A	N/A	5.0 (8/4) 18.1 (8/1)
Power Efficiency (TOPS/W)	N/A	0.32 (16/16) 1.07 (16/1) M/L[3]: 0.8V, 0.65V	2.47 (8/8) 11.9 (2/2) 17.1 (8/8, sparse 0.875) M/L[3]: 0.5V, 0.4V	N/A	N/A	1.10 (8/4) 3.00 (8/1) 27.8 (8/1, BS 0.9) 0.7V
Power (μW)	184@25MHz 0.6V (CNN + BDT)	410@1FPS M/L[3]: 0.8V, 0.65V (CNN + BDT)	228@5MHz M/L[3]: 0.5V, 0.4V (ResNet-8)	740@2.5MHz 0.8V (DSCNN, GRU)	153@120kHz 0.5V (CNN)	275@5MHz 793@20MHz, 0.7V (MobileNetV1)

[1]Depthwise Separable CNN, [2]Gated Recurrent Unit, [3]Memory and Logic Supply Voltages

and operates at an extremely low voltage of 0.4 V. On the other hand, Pianissimo runs under relatively older technology and higher operational voltages.

VI. CONCLUSION

This paper presents a sub-mW class inference accelerator called Pianissimo, supporting progressively adjustable bit-precision. Leveraging a progressive bit-by-bit datapath, Pianissimo achieves adaptive precision that ranges from 1-bit to 8-bit. Remarkably, scalable precision applications, AP and MP, are obtained using a single weight set without reducing PE utilization. Pianissimo also supports BS processing using sensor information and suppresses unnecessary computation of non-ROIs. SW-HW cooperative control enhances the system's flexibility, enabling it to accommodate various adaptive inference approaches. Our results show that Pianissimo achieves 0.49–1.25 TOPS/W at 0.7 V on MobileNetV1. Additionally, Pianissimo demonstrates practical performance across various models while operating on sub-mW class power. Thus, Pianissimo introduces a new dimension of flexibility to ultra-low power applications and shows promise in broadening the scope of use cases that can be efficiently supported. Future work will focus on integrating Pianissimo with actual sensor systems and exploring further flexibility enhancements through sparsity utilization within the low-power paradigm.

REFERENCES

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[2] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "PACT: Parameterized clipping activation for quantized neural networks," 2018, *arXiv:1805.06085*.

[3] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," 2019, *arXiv:1902.08153*.

[4] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "HAQ: Hardware-aware automated quantization with mixed precision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8604–8612.

[5] Z. Cai and N. Vasconcelos, "Rethinking differentiable search for mixed-precision neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2346–2355.

[6] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[7] Y. Li, K. Adamczewski, W. Li, S. Gu, R. Timofte, and L. Van Gool, "Revisiting random channel pruning for neural network compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 191–201.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[10] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–13.

[11] C. Banbury, C. Zhou, I. Fedorov, R. M. Navarro, U. Thakker, D. Gope, V. J. Reddi, M. Mattina, and P. N. Whatmough, "MicroNets: Neural network architectures for deploying TinyML applications on commodity microcontrollers," in *Proc. Int. Conf. Mach. Learn. Syst.*, 2021, pp. 517–532.

[12] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[13] P. Judd, J. Albericio, T. Hetherington, T. M. Aamodt, and A. Moshovos, "Stripes: Bit-serial deep neural network computing," in *Proc. 49th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2016, pp. 1–12.

[14] J. Albericio, A. Delmás, P. Judd, S. Sharify, G. O'Leary, R. Genov, and A. Moshovos, "Bit-pragmatic deep neural network computing," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2017, pp. 382–394.

[15] H. Sharma, J. Park, N. Suda, L. Lai, B. Chau, V. Chandra, H. Esmaeilzadeh, and J. K. Kim, "Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network," in *Proc. ACM/IEEE 45th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2018, pp. 764–775.

[16] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H.-J. Yoo, "UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 173–185, Jan. 2019.

[17] A. D. 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 *Proc. 24th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, Apr. 2019, pp. 749–763.

[18] S. Sharify, A. D. Lascorz, M. Mahmoud, M. Nikolic, K. Siu, D. M. Stuart, Z. Poulos, and A. Moshovos, "Laconic deep learning inference acceleration," in *Proc. ACM/IEEE 46th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2019, pp. 304–317.

[19] H. Lu, L. Chang, C. Li, Z. Zhu, S. Lu, Y. Liu, and M. Zhang, "Distilling bit-level sparsity parallelism for general purpose deep learning acceleration," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2021, pp. 963–976.

[20] G. Li, W. Xu, Z. Song, N. Jing, J. Cheng, and X. Liang, "Ristretto: An atomized processing architecture for sparsity-condensed stream flow in CNN," in *Proc. 55th IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2022, pp. 1434–1450.

[21] F. Conti, D. Rossi, G. Paulin, A. Garofalo, A. Di Mauro, G. Rutishauer, G. M. Ottavi, M. Eggimann, H. Okuhara, V. Huard, O. Montfort, L. Jure, N. Exibard, P. Gouedo, M. Louvat, E. Botte, and L. Benini, "A 12.4 TOPS/W @ 136GOPS AI-IoT system-on-chip with 16 RISC-V, 2-to-8 b precision-scalable DNN acceleration and 30%-boost adaptive body biasing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 21–23.

[22] S. Moon, H.-G. Mun, H. Son, and J.-Y. Sim, "A 127.8 TOPS/W arbitrarily quantized 1-to-8 b scalable-precision accelerator for general-purpose deep learning with reduction of storage, logic and latency waste," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 21–23.

[23] W. Shan, M. Yang, J. Xu, Y. Lu, S. Zhang, T. Wang, J. Yang, L. Shi, and M. Seok, "A 510 nW 0.41 V low-memory low-computation keyword-spotting chip using serial FFT-based MFCC and binarized depthwise separable convolutional neural network in 28 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 230–232.

[24] Y. Lu, V. L. Le, and T. T.-H. Kim, "A 184 μW real-time hand-gesture recognition system with hybrid tiny classifiers for smart wearable devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 156–158.

[25] P. Jokic, E. Azarkhish, R. Cattenoz, E. Türetken, L. Benini, and S. Emery, "A sub-mW dual-engine ML inference system-on-chip for complete end-to-end face-analysis at the edge," in *Proc. Symp. VLSI Circuits*, Jun. 2021, pp. 1–2.

[26] V. Jain, S. Giraldo, J. D. Roose, B. Boons, L. Mei, and M. Verhelst, "TinyVers: A 0.8-17 TOPS/W, 1.7 μW-20 mW, tiny versatile system-on-chip with state-retentive eMRAM for machine learning inference at the extreme edge," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2022, pp. 20–21.

[27] S. Park, S. Lee, J. Park, H.-S. Choi, and D. Jeon, "A 0.81 mm^2 740 μW real-time speech enhancement processor using multiplier-less PE arrays for hearing aids in 28 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 340–342.

[28] A. Kosuge, R. Sumikawa, Y.-C. Hsu, K. Shiba, M. Hamada, and T. Kuroda, "A 183.4 nJ/inference 152.8 μW single-chip fully synthesizable wired-logic DNN processor for always-on 35 voice commands recognition application," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2023, pp. 1–2.

[29] E. Nunez, M. Horton, A. Prabhu, A. Ranjan, A. Farhadi, and M. Rastegari, "LCS: Learning compressible subspaces for efficient, adaptive, real-time network compression at inference time," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 5188–5197.

[30] M. Rusci, D. Rossi, E. Farella, and L. Benini, "A sub-mW IoT-endnode for always-on visual monitoring and smart triggering," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1284–1295, Oct. 2017.

[31] J. Suzuki, J. Yu, M. Yasunaga, Á. L. García-Arias, Y. Okoshi, S. Kumazawa, K. Ando, K. Kawamura, T. V. Chu, and M. Motomura, "Pianissimo: A sub-mW class DNN accelerator with progressive bit-by-bit datapath architecture for adaptive inference at edge," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2023, pp. 1–2.

[32] J. Suzuki, T. Kaneko, K. Ando, K. Hirose, K. Kawamura, T. V. Chu, M. Motomura, and J. Yu, "ProgressiveNN: Achieving computational scalability with dynamic bit-precision adjustment by MSB-first accumulative computation," *Int. J. Netw. Comput.*, vol. 11, no. 2, pp. 338–353, 2021.

[33] C. Banbury et al., "MLPerf tiny benchmark," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–12. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/da4fb5c6e93e74d3df8527599fa62642-Paper-round1.pdf

[34] K. Goetschalckx and M. Verhelst, "DepFiN: A 12 nm, 3.8 TOPs depth-first CNN processor for high research image processing," in *Proc. Symp. VLSI Circuits*, Jun. 2021, pp. 1–2.

[35] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally, "SCNN: An accelerator for compressed-sparse convolutional neural networks," in *Proc. ACM/IEEE 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2017, pp. 27–40.

[36] A. Gondimalla, N. Chesnut, M. Thottethodi, and T. N. Vijaykumar, "SparTen: A sparse tensor accelerator for convolutional neural networks," in *Proc. 52nd Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2019, pp. 151–165.

[37] Y. N. Wu, P.-A. Tsai, A. Parashar, V. Sze, and J. S. Emer, "Sparseloop: An analytical, energy-focused design space exploration methodology for sparse tensor accelerators," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, Mar. 2021, pp. 232–234.

[38] Q. Jin, L. Yang, and Z. Liao, "AdaBits: Neural network quantization with adaptive bit-widths," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2143–2153.

[39] A. Bulat and G. Tzimiropoulos, "Bit-mixer: Mixed-precision networks with runtime bit-width selection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5168–5177.

[40] O. Kumagai, A. Niwa, K. Hanzawa, H. Kato, S. Futami, T. Ohyama, T. Imoto, M. Nakamizo, H. Murakami, T. Nishino, A. Bostamam, T. Iinuma, N. Kuzuya, K. Hatsukawa, F. Brady, W. Bidermann, T. Wakano, T. Nagano, H. Wakabayashi, and Y. Nitta, "A 1/4-inch 3.9 Mpixel low-power event-driven back-illuminated stacked CMOS image sensor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 86–88.

[41] K. Hirose, J. Yu, K. Ando, Y. Okoshi, Á. L. García-Arias, J. Suzuki, T. V. Chu, K. Kawamura, and M. Motomura, "Hiddenite: 4K-PE hidden network inference 4D-tensor engine exploiting on-chip model construction achieving 34.8-to-16.0 TOPS/W for CIFAR-100 and ImageNet," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[42] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1," 2016, *arXiv:1602.02830*.

[43] J. Lin, W.-M. Chen, Y. Lin, J. Cohn, C. Gan, and S. Han, "MCUNet: Tiny deep learning on IoT devices," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jul. 2020, pp. 11711–11722.

[44] J. Lin, W.-M. Chen, H. Cai, C. Gan, and S. Han, "MCUNetV2: Memory-efficient patch-based inference for tiny deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 2346–2358.

[45] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," 2019, *arXiv:1909.09347*.

[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[47] A. Chowdhery, P. Warden, J. Shlens, A. Howard, and R. Rhodes, "Visual wake words dataset," 2019, *arXiv:1906.05721*.

[48] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, E. Reynaud, P. Mostafalu, F. Brady, L. Chotard, F. LeGoff, H. Takahashi, H. Wakabayashi, Y. Oike, and C. Posch, "A 1280 × 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066GEPS readout, programmable event-rate controller and compressive data-formatting pipeline," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 112–114.

[49] C. Young, A. Omid-Zohoor, P. Lajevardi, and B. Murmann, "A data-compressive 1.5/2.75-bit log-gradient QVGA image sensor with multi-scale readout for always-on object detection," *IEEE J. Solid-State Circuits*, vol. 54, no. 11, pp. 2932–2946, Nov. 2019.

JUNNOSUKE SUZUKI (Graduate Student Member, IEEE) received the B.E. degree in electronics from Hokkaido University, Sapporo, Japan, in 2020, and the M.E. degree in information and communication engineering from the Tokyo Institute of Technology, Yokohama, Japan, in 2022, where he is currently pursuing the Ph.D. degree.

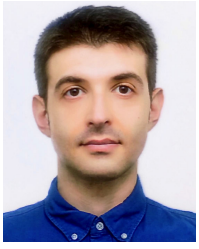
He received the Research Fellowship for Young Scientists from JSPS, in 2022. His research interests include energy-efficient accelerator design and machine learning.

JAEHOON YU (Member, IEEE) received the B.E. degree in electrical and electronic engineering and the M.S. degree in informatics (communications and computer engineering) from Kyoto University, Kyoto, Japan, in 2005 and 2007, respectively, and the Ph.D. degree in informatics (information systems engineering) from Osaka University, Osaka, Japan, in 2013.

From 2013 to 2019, he was an Assistant Professor with Osaka University. From 2019 to 2023, he was an Associate Professor with the Tokyo Institute of Technology, Japan. He is currently the Vice President of Technology with Samsung Electronics. His research interests include computer vision, machine learning, and system-level design. He is a member of IEICE and IPSJ.

MARI YASUNAGA (Graduate Student Member, IEEE) received the B.E. degree in information and communication engineering from the Tokyo Institute of Technology, Yokohama, Japan, in 2022, where she is currently pursuing the M.E. degree in information and communication engineering.

Her research interests include machine learning and computer architecture.



ÁNGEL LÓPEZ GARCÍA-ARIAS (Graduate Student Member, IEEE) received the B.E. degree in telecommunications engineering from the Autonomous University of Madrid, Madrid, Spain, in 2017, and the M.S. degree in information science and technology from Osaka University, Osaka, Japan, in 2021. He is currently pursuing the Ph.D. degree in information and communication engineering with the Tokyo Institute of Technology, Yokohama, Japan.

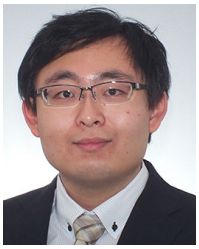
His graduate studies are sponsored by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan, since 2018. His research interests include neural network architecture, non-linear systems, and digital design.

YASUYUKI OKOSHI (Graduate Student Member, IEEE) received the B.E. and M.S. degrees in engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2021 and 2023, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include deep learning and computer architecture.

SHUNGO KUMAZAWA (Graduate Student Member, IEEE) received the B.E. degree in electronics from Hokkaido University, Sapporo, Japan, in 2020, and the M.E. degree in information and communication engineering from the Tokyo Institute of Technology, Yokohama, Japan, where he is currently pursuing the Ph.D. degree in information and communication engineering.

He received the Research Fellowship for Young Scientists from JSPS, in 2022. His research interests include machine learning and computer architecture.



KOTA ANDO (Member, IEEE) received the B.E. degree in electronics and the M.S. degree in information technology from Hokkaido University, Sapporo, Japan, in 2016 and 2018, respectively, and the Ph.D. degree in engineering from the Tokyo Institute of Technology, Yokohama, Japan, in 2021.

He is currently an Assistant Professor with Hokkaido University. He was an Assistant Professor with the Tokyo Institute of Technology, from 2021 to 2022. He was a JSPS Research Fellow, from 2018 to 2021, during the Ph.D. degree. His research interests include reconfigurable architectures, memory-centric processing, and hardware-aware algorithms for efficient deep learning processing.

Dr. Ando has been a member of IEICE, since 2016. He received the Best Student Presentation Award from the Technical Committee on Reconfigurable Systems of IEICE, Japan, in 2016 and 2017, respectively, the Best Student Poster Award from the Technical Committee on Integrated Circuits and Devices of IEICE, in 2018, and the Best Paper Award at the 2018 International Conference on Field-Programmable Technology.

KAZUSHI KAWAMURA (Member, IEEE) received the B.Eng., M.Eng., and Dr.Eng. degrees in computer science from Waseda University, in 2012, 2013, and 2016, respectively.

From 2018 to 2019, he was an Assistant Professor with the Department of Communications and Computer Engineering, Waseda University. He is currently a specially appointed Assistant Professor with the Institute of Innovative Research, Tokyo Institute of Technology. His research interests include parallel algorithms and architectures for annealing computation and machine learning. He is a member of IEICE and IPSJ.

THIEM VAN CHU (Member, IEEE) received the Ph.D. degree from the Tokyo Institute of Technology, in 2018.

Upon graduation, he joined the School of Information Science, Japan Advanced Institute of Science and Technology as an Assistant Professor. In 2020, he moved to the Tokyo Institute of Technology. His research interests include the intersection of computer architecture, reconfigurable computing, and machine learning.

MASATO MOTOMURA (Fellow, IEEE) received the B.S., M.S., and Dr.Eng. degrees from Kyoto University, Kyoto, Japan, in 1985, 1987, and 1996, respectively.

In 1987, he joined NEC Central Research Laboratories, Kawasaki, Japan, working on various hardware architectures, including approximate text search engines, multi-threaded on-chip parallel processors, computing-in-memory chips, and reconfigurable systems. From 2001 to 2008, he was with NEC Electronics, Kawasaki, Japan, where he led the research and business development of the dynamically reconfigurable processor (DRP) he invented. He was also a Visiting Researcher with the MIT Laboratory for Computer Science, Cambridge, USA, from 1991 to 1992, and a Group Manager of architecture-circuits interdisciplinary research with the NEC Central Laboratory, from 2008 to 2011, respectively. In 2011, he changed his position from industry to academia and became a Professor with Hokkaido University, Sapporo, Japan, to cultivate solid-state circuit research activities with younger generations. Later, he became a Professor with the Tokyo Institute of Technology (TokyoTech), Yokohama, Japan, in 2019, where he established and he has been leading the artificially intelligent computing (ArtIC) research unit. Since 2011, he has been actively working on reconfigurable and parallel architectures for deep neural networks, machine learning, annealing machines, and general intelligent/domain-specific computing. His group has been published "AI chip" papers almost every year at ISSCC and Symposium on VLSI, since 2017. He received the IEEE JSSC Annual Best Paper Award, in 1992, the IPSJ Annual Best Paper Award, in 1999, and the IEICE Achievement Award, in 2011, respectively. He was also awarded Ichimura Academic Award and Yamasaki Award, in 2022, for his leadership in developing and productizing DRP technology (a series of DRP-based microcontroller products are now produced by Renesas Electronics), and the accumulation of AI-chip achievements in recent years. He is a member of IEICE, IPSJ, JSAI, and EAJ. He is a 2022 IEEE Fellow (SSCS) for contributions to memory-logic integration of reconfigurable chip architecture.