IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 June 2023, accepted 12 July 2023, date of publication 18 July 2023, date of current version 25 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296429

RESEARCH ARTICLE

Integrated Imager and 3.22 μs/Kernel-Latency All-Digital In-Imager Global-Parallel Binary Convolutional Neural Network Accelerator for Image Processing

RUIZHI WANG, CHENG-HSUAN WU, AND MAKOTO TAKAMIYA, (Senior Member, IEEE)
Institute of Industrial Science, The University of Tokyo, Tokyo 153-8505, Japan

Corresponding author: Ruizhi Wang (oueichi@iis.u-tokyo.ac.jp)

This work was partly supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 19H02188.

ABSTRACT This paper presents an innovative approach to achieve ultralow-latency convolutional neural network (CNN) processing, which is critical for real-time image processing applications such as autonomous driving and virtual reality. Traditional CNN accelerators employing in/near-array-computing (inclusive of in/near-memory-computing and in/near-sensor-computing) architectures have struggled to meet real-time requirements due to latency bottlenecks encountered with conventional column-parallel processing for image processing. To address this challenge, we propose a novel, all-digital in-imager global-parallel binary convolutional neural network (IIGP-BNN) accelerator. This new approach employs a global-parallel processing concept, which enables multiply-and-accumulate operations (MACs) to be executed simultaneously within the imager array in a 2D manner, eliminating the additional latency associated with row-by-row processing and data access from random access memories (RAMs). In this design, convolution and subsampling operations using a 3×3 kernel are completed within just nine steps of global-parallel processing, regardless of image size. This results in a theoretical reduction of over 88.5% of repeated row scans compared to conventional column-parallel processing architectures, thus significantly reducing computing latency. We have designed and prototyped a 30×30 integrated imager and IIGP-BNN accelerator IC using a 0.18 μm CMOS process. This prototype achieved a latency of 3.22 μs/kernel on the first layer convolution at a power supply of 1 V and a clock frequency of 35.7 MHz. This represents a latency reduction of 35.6% compared to the state-of-the-art in/near-imager-computing works. This proposed global-parallel processing concept opens up the potential for processing high-resolution images in 4K and 8K with the same ultralow latency, marking a significant advancement in high-speed image processing.

INDEX TERMS Convolutional neural network, ultralow latency, global-parallel processing, in-imager-computing, image processing.

I. INTRODUCTION

Convolutional Neural Networks (CNNs) have recently advanced considerably within the machine learning (ML) domain, becoming a piece of crucial innovative technology in the majority of image and vision processing tasks [1], [2]. Proposals for CNN accelerators employing in/near-array-computing (inclusive of in/near-memory-computing and in/near-sensor-computing) architectures aim to achieve superior energy or area efficiency due to their effective reduction on the computational data loading [3], [6], especially the in/near-sensor-computing architectures have shown better energy efficiency and communication latency performance than in/near-memory-computing architectures due to the further reduction of data access for raw image and extracted features [7], [8]. Among various sensors, image

The associate editor coordinating the review of this manuscript and approving it for publication was Mario Donato Marino.

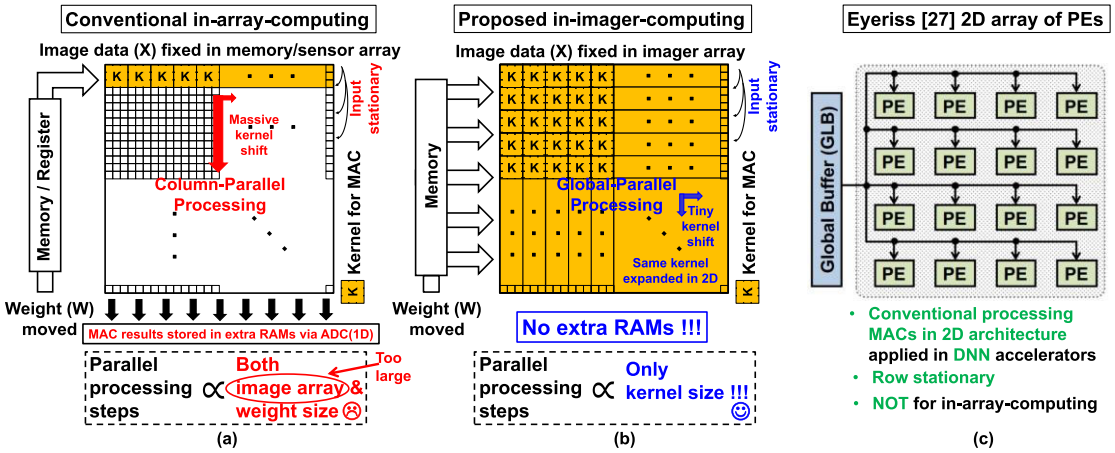

FIGURE 1. Convolution processing by (a) conventional column-parallel processing, (b) proposed global-parallel processing using IIGP-BNN architecture and (c) conventional processing MACs in 2D architecture in Eyeriss [27].

sensors (or imagers) can be fabricated with a complementary metal–oxide–semiconductor (CMOS)-compatible process over a large scale, which makes the proposal of in/near-sensor-computing works becoming a trend for CNN accelerators [9], [14]. The in/near-array-computing proposals usually necessitate trade-offs involving accuracy and latency. However, in the context of CNN accelerators, computation latency is often viewed as a critical metric, especially for applications contingent on real-time image processing that necessitates high-speed computing. For instance, past research illustrates that to prevent noticeable effects on the performance of autonomous vehicles, the total latency level should ideally be under 170 ms [15]. Within Virtual Reality (VR) environments, player performance may be compromised by latencies exceeding 100 ms, suggesting that the target latency range lies between 50 ms and 100 ms [16]. Current CNN technologies face two principal challenges when applied to real-time image processing. Firstly, an increase in the depth and size of convolutional layers in recently developed CNN models elevates computing latency [17]. This rise in latency makes it challenging for these models to satisfy the needs of real-time vision applications, considering that task runtime is governed by convolution latency [18]. Recent research on prevalent CNN architectures used in image recognition, such as VGG16, shows computing latencies ranging from 169 ms to 4.3 s—a duration too lengthy for real-time vision processing [19], [22]. Consequently, it is essential to reduce the computing latency in each convolutional layer.

Secondly, the necessity for higher image data resolution in real-time image processing has surged in recent years. The resolution requirement in the autonomous driving sector has grown from the conventional Video Graphics Array (VGA, 640 $\times$ 480) to 4K (3840 $\times$ 2160) [23]. Past research on machine vision systems and chips suggest that slow processing speed is partly due to the vast amounts of column-parallel processing that the processing elements (PEs) must execute, which is ultimately dependent on the image data array size [24]. To better demonstrate this phenomenon, Fig. 1 (a) illustrates how multiply-and-accumulate operations (MACs) function in conventional in-array-computing CNN circuits when conducting image recognition. Weight (W) is input into the image data array (X) that capture and store image pixel values to carry out column-parallel processing, employing row-by-row kernel (K) scans (including right shifts for each row of kernel scans) for the convolution operation. Subsequently, convolution results are then stored in additional Random-Access Memories (RAMs) in a 1D manner [25], [26]. This type of column-parallel processing must be repeatedly executed to complete the entire MAC process for the entire data array and the kernel shift is huge, rendering the convolution latency per kernel to be reliant on the image data array size. For instance, when a 3 $\times$ 3 kernel is used at a stride of 1 in architectures with column-parallel processing for 28 $\times$ 28 image data processing (using the MNIST dataset), it's estimated that the convolution operation is completed after 26 repetitions of row-by-row kernel scans. This figure escalates to 1,078 for Full-HD (1920 $\times$ 1080) resolution image processing and 2,158 for 4K resolution image processing (at the same stride of 1, not considering other factors such as overfitting), resulting in convolution latencies that are comparatively over 40$\times$ and 80$\times$ higher, respectively. Hence, there is a clear need to reduce the convolution latency of high-resolution image processing.

Implementing Multiply-and-Accumulate (MAC) operations in 2D presents an optimal solution to both identified issues related to computing latency in CNN processing, which is emphasized by previous works such as Eyeriss and Eyeriss v2 [27], [28]. Fig. 1 (b) depicts a proposed concept of global-parallel processing, contrasting it with the traditional column-parallel processing method applied in conventional in/near-array-computing works illustrated in Fig. 1 (a). This innovative approach simultaneously processes all 2D image data, eliminating the need for repeated row-by-row kernel scans that are determined by the image resolution (either

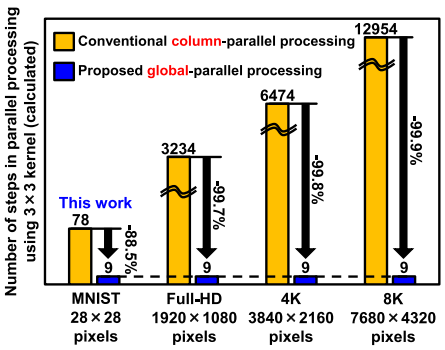

FIGURE 2. Theoretical reduction in the number of steps in parallel processing with proposed global-parallel processing concept (at stride of 1).

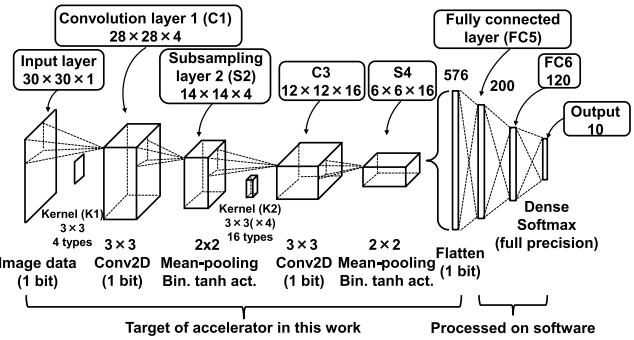

FIGURE 3. CNN model based on LeNet-5 applied in proposed IIGP-BNN system.

the row or the column size) as in conventional in-array-computing architectures. Tiny kernel shifts are needed in this proposed global-parallel processing architecture. Instead, the number of steps required depends solely on the size of the kernel. Furthermore, there are no extra RAMs needed for the storage of processed results, which is also an advantage of the proposed architecture. Additionally, Eyeriss [27], [28], which is considered as a conventional processing MAC in 2D architecture, is also shown in Fig. 1(c). The Eyeriss adopts the same idea of processing MACs through the whole map two-dimensionally as this work, with row stationary dataflow, and is applied in the DNN accelerators instead of in-array-computing works. As a result, these processing MACs in 2D significantly curtail computing latency by at least 88.5%, a reduction correlated with image data resolution as displayed in Fig. 2. This figure also shows the number of steps of parallel processing that is demanded in a higher resolution of architectures applying conventional column-parallel processing and the proposed global-parallel processing. The convolution latency is cut by 99.7% for Full-HD resolution (1920 $\times$ 1080) image processing and by 99.8% for 4K resolution (3840 $\times$ 2160) image processing when implemented through the proposed global-parallel processing. Moreover, this latency reduction is computed to exceed 99.9% for 8K resolution (7680 $\times$ 4320) image processing using the proposed architecture design. Theoretically, this results in ultrafast processing suitable for real-time image processing tasks.

However, another challenge arises when considering the memory bandwidth limitation [29], necessitating that not only the convolution operation process, but also data transmission and subsampling operations must be processed in 2D. In this study, we introduce an all-digital in-imager global-parallel binary convolutional neural network (IIGP-BNN) architecture that leverages the proposed global-parallel processing concept depicted in Fig. 1 (b). This IIGP-BNN architecture integrates a binary CMOS imager circuit with the proposed IIGP-BNN accelerator, allowing for the capture of binary image data and processing of the captured data in 2D within the same circuit. The IIGP-BNN architecture is presented in an all-digital format for the benefit of circuit design and compatibility with the scaled CMOS process. The subsampling operation (encompassing both pooling and activation) is sequentially executed after the completion of the convolution operation within the same circuit, eliminating the need to store convolution results which consumes additional RAMs. To validate this concept, particularly to verify the two key elements, namely in-imager (II in the IIGP-BNN) and global-parallel processing (GP in the IIGP-BNN) in this work, we designed and fabricated a 30 $\times$ 30 sized IIGP-BNN accelerator prototype in a 0.18 μm CMOS process. This prototyped integrated circuit (IC) comprises a 30 $\times$ 30 binary pixel array and an IIGP-BNN accelerator capable of processing two layers of both convolution and sub-sampling operations (with fully-connected layers being processed off-chip) on the MNIST dataset. Our measurements reveal that this accelerator achieved an ultralow latency of 3.22 μs/kernel on the first convolution and subsampling layer at a supply voltage of 1 V, a clock frequency of 35.7 MHz, and a throughput of 4.36 GOPS, thereby reducing latency by 35.6% compared with current state-of-the-art in/near-imager-computing work [12].

Specifically, this paper's primary contributions are twofold:

1. The paper highlights that current hardware latency remains excessively lengthy to meet the requirements of real-time image processing. In response, we propose a global-parallel processing concept, which theoretically results in an over 88.5% latency reduction compared to architectures using column-parallel processing, irrespective of image resolution.

2. The paper also provides designs and prototypes for an integrated circuit (IC) that combines an imager and an ultralow latency, all-digital in-imager global-parallel binary convolutional neural network (IIGP-BNN) accelerator. Furthermore, these measurements demonstrate that the suggested accelerator reduces latency on the first convolutional and subsampling layer by 35.6% in comparison to the state-of-the-art in/near-imager-computing works.

This paper is organized as follows. Section II introduces the proposed IIGP-BNN architecture, elaborating on the circuit design and the functionality of the suggested global-parallel

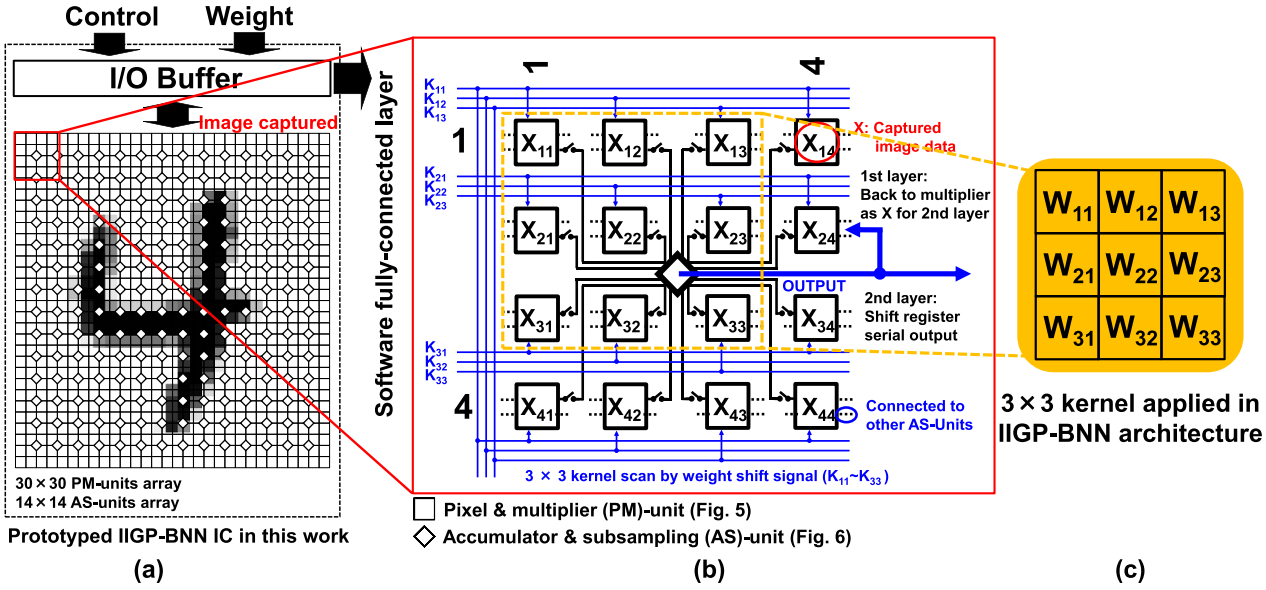

FIGURE 4. (a) Overall schematic of proposed IIGP-BNN architecture. (b) Connection between proposed PM-units and AS-units. (c) 3 × 3 kernel applied in IIGP-BNN architecture.

processing concept. Section III presents the experimental results. Section IV encompasses discussions and highlights potential areas for future improvements. Finally, Section V concludes the paper.

II. DESIGN OF PROPOSED IIGP-BNN ARCHITECTURE

This section elucidates the proposed IIGP-BNN architecture. We begin by presenting a schematic overview of the system, followed by a comprehensive explanation of the circuit design. Fig. 3 illustrates the binary CNN model deployed within the proposed IIGP-BNN accelerator for MNIST database processing. This model, a customized version of LeNet-5, is adapted for the proposed architecture to perform 3 × 3 kernel convolution and subsampling operations in 2D, thereby achieving lower latency. The accelerator executes four layers of both convolution and subsampling operations, namely C1, S2, C3, and S4, using fixed-size 3 × 3 kernels. The kernel size on C1 and C3 is set to 4 and 16 respectively based on a tradeoff between the kernel size and the image processing accuracy when training on Python using the MNIST dataset. The computed results from the S4 layer are exported from the accelerator and subsequently processed by software in this prototype.

A. OVERALL ARCHITECTURE OF PROPOSED IIGP-BNN

Fig. 4 (a) provides a comprehensive view of the proposed 30 × 30 IIGP-BNN accelerator circuit's architecture. This circuit processes image data captured by a binary imager circuit (consisting of a pixel array) that is integrated into the accelerator two-dimensionally. The IIGP-BNN architecture processes Multiply-Accumulate operations (MACs) within the image data array on a global 2D scale, leading to a significant convolution latency reduction of 88.5% when employing the 28 × 28 MNIST dataset. In addition, the subsampling operations are executed within the same circuit, obviating the need to store convolution results in additional RAMs. The proposed architecture integrates a 14 × 14 accumulator & subsampling-units (AS-units) array into the larger 30 × 30 pixel & multiplier-units (PM-units) array. As a result, all convolution and subsampling operations are processed within the same array, avoiding the need for extra RAM data access. This theoretically speeds up the computational process and enhances energy efficiency.

Fig. 4 (b) shows the detailed connection scheme between PM-units and AS-units. An AS-unit, connected to a 4 × 4 matrix of surrounding PM-units, is capable of concurrently processing 3 × 3 convolution and 2 × 2 mean-pooling operations, as indicated in Fig. 3. The operational flow of the proposed 30 × 30 IIGP-BNN accelerator circuit is introduced as follows. Initially, the pixel circuit within the PM-unit captures image data X_{11}–X_{33}, which is subsequently multiplied by the weight within the same unit. Following this, each AS-unit performs an accumulation process, leveraging multiplication results procured from the surrounding 4 × 4 PM-units. The outcomes, following the subsampling operation, are preserved within Flip-Flops (FFs) housed within the AS-unit. During each convolution step, the AS-unit garners multiplication results solely from the 3 × 3 PM-units in the connected 4 × 4 PM-units array, as dictated by control signals from outside of the accelerator. The 9-bit weight shift signal wires, labeled K_{11}–K_{33}, establish connections to all PM-units within the 30 × 30 array. Every fourth PM-unit, both row and column-wise, shares the same single weight shift signal wire. The connection rules for the weight shift signals K_{11}–K_{33} can be summarized as follows:

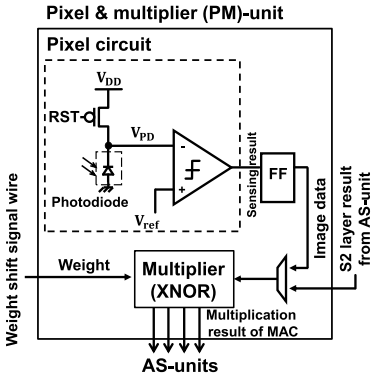

FIGURE 5. Proposed PM-unit circuit in IIGP-BNN architecture.

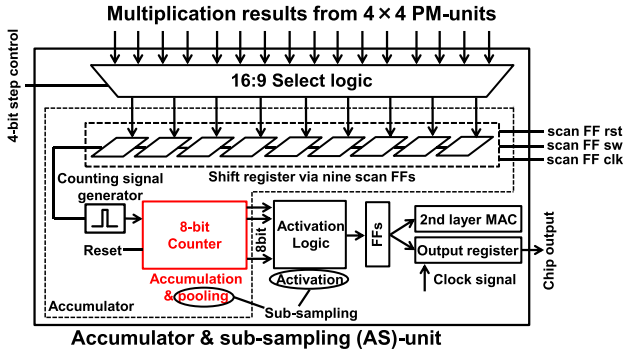

FIGURE 6. Proposed AS-unit circuit in IIGP-BNN architecture.

For each $K_{i,j}$ within the range of K_{11}–K_{33} (where i, j = 1–3), wire $K_{i,j}$ will be linked to PM-units in the row of (3 × m + i) and the column of (3 × n + j), where m, n are integers ranging from 0 to 9.

This unique signal wire connection scheme allows for a 3 × 3 kernel scan performed by data shifting across nine weight shift signal inputs. In the initial convolutional layer C1 and the subsequent subsampling layer S2, results processed using all four types of kernels, as illustrated in Fig. 3, are stored within four Flip-Flops (FFs) housed in the AS-units. After the first subsampling layer S2, stored data are reused as input data to the second convolutional layer C3. After the first subsampling layer S2, stored data are reused as input data to the second convolutional layer C3. Following the second subsampling layer S4, the computed data are stored in the shift register and are sequentially output from the accelerator circuit. The proposed IIGP-BNN circuit completes the convolution operation in nine steps of global-parallel processing, utilizing a 3 × 3 kernel scan. This reduces the number of processing steps by a theoretical 88.5% compared to the conventional architecture with column-parallel processing when processing 28 × 28 image data from the MNIST dataset, as depicted in Fig. 2.

B. DETAILS OF IIGP-BNN CIRCUIT DESIGN

Fig. 5 provides a detailed view of the components inside the proposed PM-unit cell, which includes a conventional binary pixel circuit and a multiplier. The binary image data produced by the pixel circuit is stored in a scan-flip-flop (scan-FF) prior to the multiplier in each PM-unit. Raw image data can be retrieved from the accelerator before processing through the shift register connected by these scan-FFs. An XNOR gate operates as a binary multiplier in the proposed PM-unit cell circuit [6]. Fig. 6 depicts the details of the circuit inside the proposed AS-unit cell, incorporating a 16-input-9-output select logic structure, an accumulator, an activation function logic structure, and four flip-flops (FFs) used as memory storage to reuse the processed data from the S2 layer as data is inputted into the C3 layer. The 16-input-9-output select logic structure in each AS-unit chooses the multiplication results of the MAC operations using a 3 × 3 kernel from the connected 4 × 4 PM-units. This selection is guided by a 4-bit step control signal. The accumulator is composed of a shift register linked to nine scan-FFs and an 8-bit counter, which triggers at the rising edge of the clock signal. Data from the shift register is processed by a counting signal generator, which then permits the counter to count the number of high voltage level data stored in the shift register's scan-FFs. The number of high levels counted by the counter can be equated to the accumulation of the nine multiplied results. These results are stored in the scan-FFs in this counter. The multiplied results from the PM-units are then transferred to the AS-unit to continue processing the accumulation of the MAC operation. Subsequently, subsampling operations are sequentially processed in the same unit. A total of twenty-one FFs are utilized in a single AS-unit cell, comprising eleven FFs for the select logic structure, six FFs for the counter, and four FFs for storing outputs after subsampling.

Fig. 7 illustrates the schematic for the 9-step global-parallel processing scheme used for the convolution operation using a 3 × 3 kernel on the C1 layer in the proposed architecture. The first phase of global-parallel processing is represented in steps 1 through 3. In these steps, parallel processing using a 3 × 3 kernel scan is performed across the entire PM-units array in 2D, controlled by the shifting data from the 9-bit weight shift signal wire K_{11}–K_{33}. AS-units within a 3 × 3 kernel in each step switch to the ACTIVE mode, with the remaining units transitioning into STANDBY mode. Every PM-unit multiplies the image data and weight in the XNOR gate, transmitting the result to the ACTIVE mode AS-unit within the corresponding, orange-framed region to execute accumulation and subsampling operations. As the circuit progresses from step-1 to step-2, the 2D-expanded kernel moves one column to the right. Furthermore, different AS-units will select multiplication results from PM-units. Concurrently, each AS-unit's mode changes based on whether it remains within the orange frame. AS-units transitioning to STANDBY mode from ACTIVE mode retain accumulated results from the previous step, while AS-units shifting to ACTIVE mode from STANDBY mode continue the accumulation using results stored while in the previous STANDBY mode. This process repeats in step-3. Once step-3 is completed, the first phase of the

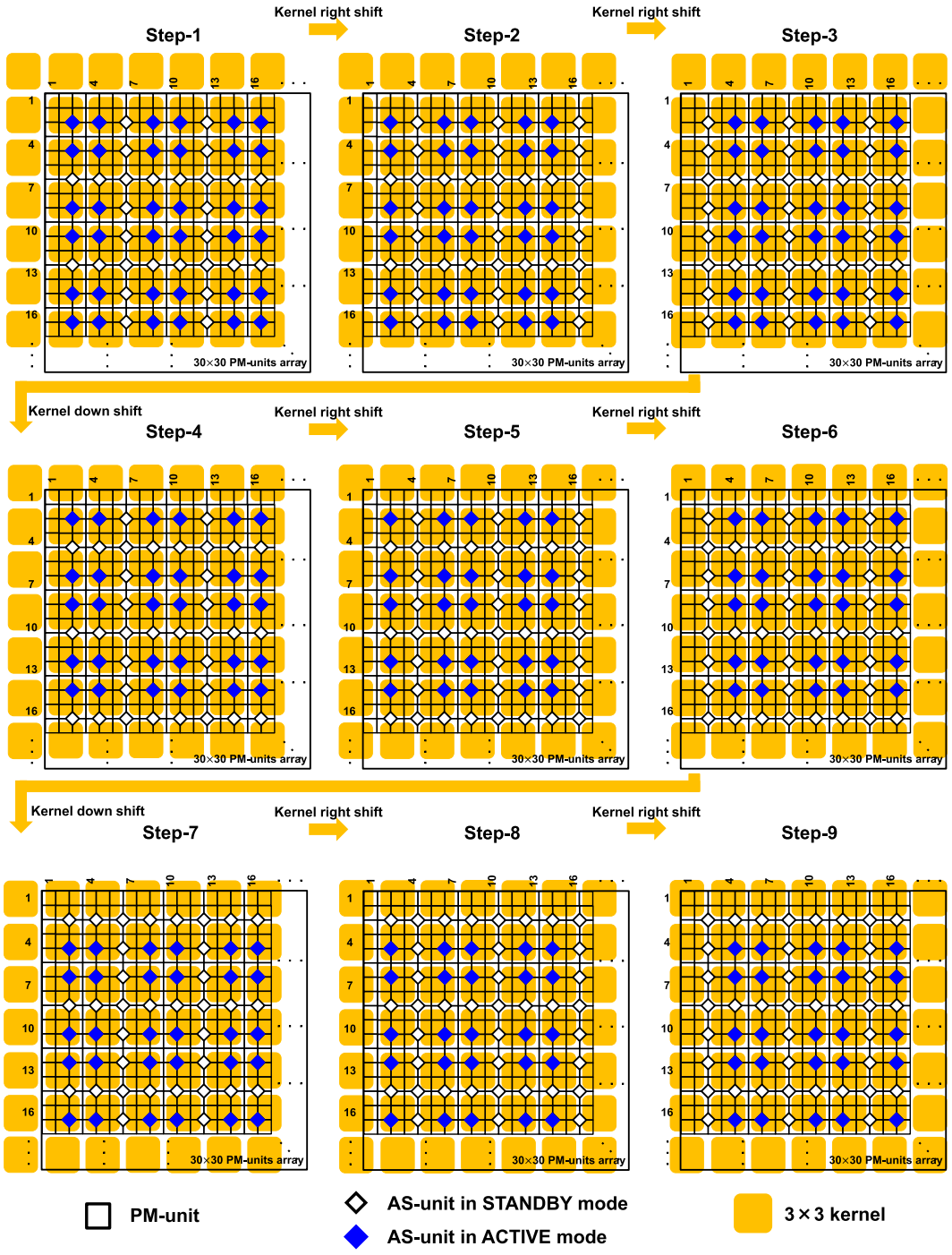

FIGURE 7. Nine-step global-parallel processing schematic using the proposed IIGP-BNN architecture.

proposed global-parallel processing is concluded. By this point, all of the results stored through kernels that scanned non-overlapping rows in the 30 × 30 image data array have been processed.

Steps 4 through 6 in Fig. 7 represent the second phase of global-parallel processing. During this phase, the 2D-expanded kernels shift down by one row relative to their position in the first phase, and they move right by one column at each step to execute MACs, just as in the first phase. Similarly, steps 7 through 9 indicate that the position of the kernels in the third phase shifts down by one row compared to the second phase, and the same parallel processing is performed. The completion of a convolution operation for a 3 × 3 kernel takes place within three phases in nine steps.

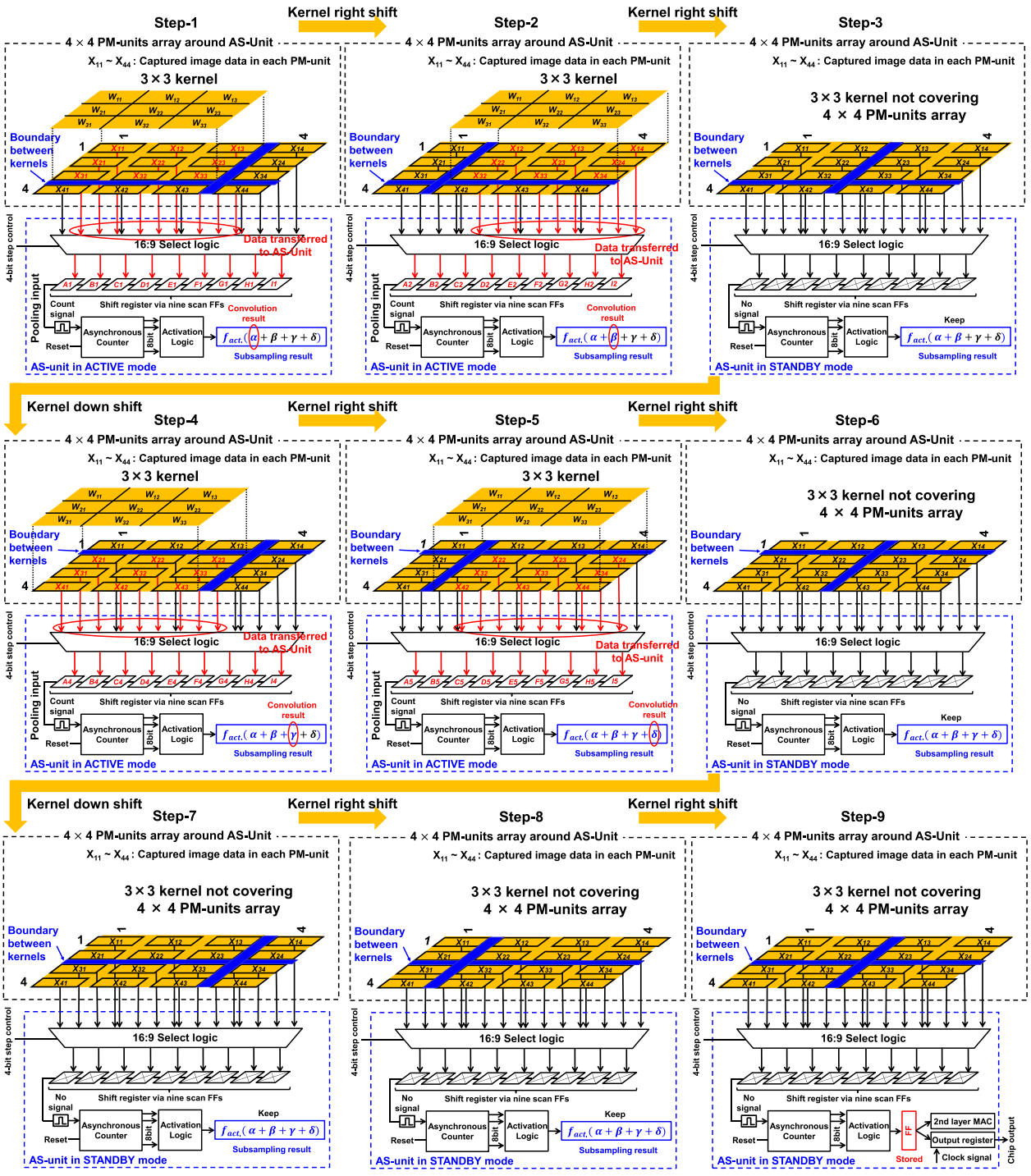

FIGURE 8. Examples of detailed data flow within sequential convolution and subsampling processing in proposed IIGP-BNN architecture.

If a 5 × 5 kernel is used, this would extend further to a five phases in twenty-five steps process, regardless of the size of the image data array.

The global-parallel processing approach in this paper also has the sequentiality of the subsampling process in the AS-unit, which eliminates the need to output convolution results to RAMs in 1D as shown in Fig. 1 (a). The underlying principle can be summarized as follows: The binary CNN model employed in this study, as depicted in Fig. 3, utilizes mean-pooling and a binary tanh activation function to complete the proposed sequential convolution and subsampling processing using the same AS-unit circuit as shown in Fig. 6. Contrary to max-pooling, which computes the maximum of each 2 × 2 convolution result from the C1

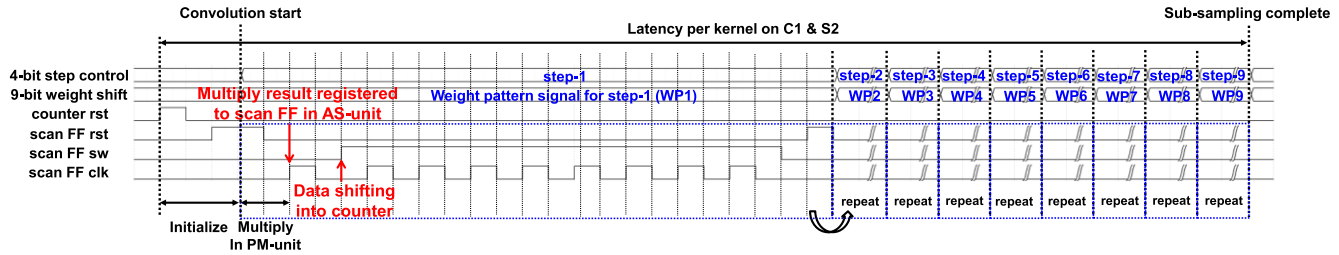

FIGURE 9. Timing chart when processing on the first convolutional layer via a single kernel as in Fig. 8.

layer, mean-pooling calculates the average, as illustrated in Fig. 3. Suppose the 2 × 2 data, which represents convolution results after nine rounds of accumulation, are denoted as α, β, γ, and δ. The output resulting from mean-pooling processing can then be represented by the following equation:

$$out_{pooling} = (\alpha + \beta + \gamma + \delta)/4. \tag{1}$$

Simultaneously, a binary tanh activation function, as opposed to the ReLU function, is utilized in the proposed architecture:

$$f_{act.}(x) = \begin{cases} +1 & if\ x \geq 0 \\ -1 & if\ x < 0. \end{cases} \tag{2}$$

The correlation between the output of the activation function and the convolution results α, β, γ, and δ can be calculated by combining equations (1) and (2):

$$f_{act.}(\alpha, \beta, \gamma, \delta) = \begin{cases} +1 & if\ (\alpha+\beta+\gamma+\delta)/4 \geq 0 \\ -1 & if\ (\alpha+\beta+\gamma+\delta)/4 < 0. \end{cases} \tag{3}$$

Equation (3) results in +1 or −1 depending on whether the numerator of equation (1) is greater than zero or not. Therefore, equation (3) can be simplified into the below equation with no accuracy loss of the model in this work:

$$f_{act.}(\alpha, \beta, \gamma, \delta) = \begin{cases} +1 & if\ (\alpha + \beta + \gamma + \delta) \geq 0 \\ -1 & if\ (\alpha + \beta + \gamma + \delta) < 0. \end{cases} \tag{4}$$

In this manner, the mean-pooling operation can be implemented using the same accumulator in the AS-unit through the non-resetting accumulation of α, β, γ, and δ, which results in 36 repetitions of accumulation using the counter shown in Fig. 6.

This paper utilizes the 4 × 4 PM-units, connected to an AS-unit from PM-unit position (1, 1) to (4, 4) in Fig. 4 (b), as an example to explain the proposed sequential convolution and subsampling processing framework in Fig. 8. This figure illustrates the data flow details and how the subsampling operations are sequentially managed in the proposed IIGP-BNN. The shift register, connected via scan FFs, exclusively receives 3 × 3 multiplication outcomes from 4 × 4 PM-units through a 16-in-9-out select logic. Subsequently, it serially produces a counter input into the counter in the AS-unit. The orange frame signifies a 3 × 3 kernel scanning with weight W_{11}–W_{33} on the PM-units array, shifting under the control of the 9-bit weight shift signal wires K_{11}–K_{33}, as illustrated in Fig. 4 (b). The 16-in-9-out select logic turns the AS-unit to the ACTIVE mode if a 3 × 3 kernel fully encompasses this AS-unit, which is under the command of the 4-bit step control signal. The AS-unit solely conducts accumulation and subsampling in the ACTIVE mode, preserving the results in the STANDBY mode. Steps 1 to 3 in Fig. 8 elucidate the data flow of the 4 × 4 array in the initial phase. In step 1, a 3 × 3 kernel fully envelops these 4 × 4 PM-units, and nine multiplication outcomes from PM-unit position (1, 1) to (3, 3) are selected and transmitted by the select logic; they can be calculated as:

$$\begin{bmatrix} A1 \\ B1 \\ C1 \\ D1 \\ E1 \\ F1 \\ G1 \\ H1 \\ I1 \end{bmatrix} = \begin{bmatrix} W_{11} & & & & & & & & \\ & W_{12} & & & & & & & \\ & & W_{13} & & & & & & \\ & & & W_{21} & & & & & \\ & & & & W_{22} & & & & \\ & & & & & W_{23} & & & \\ & & & & & & W_{31} & & \\ & & & & & & & W_{32} & \\ & & & & & & & & W_{33} \end{bmatrix} \times \begin{bmatrix} X_{11} \\ X_{12} \\ X_{13} \\ X_{21} \\ X_{22} \\ X_{23} \\ X_{31} \\ X_{32} \\ X_{33} \end{bmatrix} \tag{5}$$

In this stage, the 3 × 3 kernel entirely covers the 4 × 4 PM-units, and the AS-unit transitions into the ACTIVE mode. The nine results ranging from A1 to I1 are individually stored in the scan-FFs, creating an input signal for the counter. This can be interpreted as the convolution result α as per equation (4):

$$\alpha = \sum_{N=A\sim I} N1. \tag{6}$$

In the second step, the 3 × 3 kernel shifts one column to the right, yet remains within the scope of the 4 × 4 PM-units array. The product results from PM-unit positions (2, 1) to (4, 3) are relayed to the select logic, keeping the AS-unit in the ACTIVE mode. The nine outcomes, A2 through I2, are

sequentially dispatched to the counter, and can be computed in a similar fashion:

$$\begin{bmatrix} A2 \\ B2 \\ C2 \\ D2 \\ E2 \\ F2 \\ G2 \\ H2 \\ I2 \end{bmatrix} = \begin{bmatrix} W_{11} & & & & & & & & \\ & W_{12} & & & & & & & \\ & & W_{13} & & & & & & \\ & & & W_{21} & & & & & \\ & & & & W_{22} & & & & \\ & & & & & W_{23} & & & \\ & & & & & & W_{31} & & \\ & & & & & & & W_{32} & \\ & & & & & & & & W_{33} \end{bmatrix} \times \begin{bmatrix} X_{12} \\ X_{13} \\ X_{14} \\ X_{22} \\ X_{23} \\ X_{24} \\ X_{32} \\ X_{33} \\ X_{34} \end{bmatrix} \tag{7}$$

The convolution result, β, is derived after nine consecutive counting instances utilizing A2 through I2. Following step-2, the counter's cumulative result becomes the sum of α and β.

$$\beta = \sum_{N=A\sim I} N2. \tag{8}$$

In step-3, the kernel shifts one more column to the right, resulting in no 3 × 3 kernel covering this 4 × 4 array. The select logic thereby switches the AS-unit to the STANDBY mode, leaving the result in the counter unaltered.

In a similar manner, during steps-4 and -5, as illustrated in Fig. 8, the AS-unit returns to the ACTIVE mode. The multiplication results A4 through I4 in step-4, and A5 through I5 in step-5, are conveyed to the counter as the convolution results γ and δ, respectively

$$\gamma = \sum_{N=A\sim I} N4 \tag{9}$$

$$\delta = \sum_{N=A\sim I} N5. \tag{10}$$

In this 4 × 4 PM-units example, the pooling operation for this kernel concludes in step-5, with the pooling result being the summation of α, β, γ, and δ. This AS-unit switches to the STANDBY mode from step-6 through step-9, as displayed in Fig. 8. In the 30 × 30 PM-units array, the residual AS-units remain operational until the completion of step-9. Upon the completion of step-9, the outcomes of the subsampling process, derived from equation (4), are stored in the flip-flops (FFs) within the AS-units. These results can be returned to the PM-units as input X for the C3 layer, as illustrated in Fig. 3. Similar operations are executed in the C3 layer using a 3 × 3 × 4 kernel within the same AS-unit, as depicted in Fig. 6. The solitary difference is that the counter is required to increment up to 144 times to complete the convolution in the C3 layer. Post subsampling processing in the S4 layer, the results are preserved in the scan-FFs within the AS-units and sequentially outputted from the circuit. At this point, computations by the proposed IIGP-BNN accelerator reach completion.

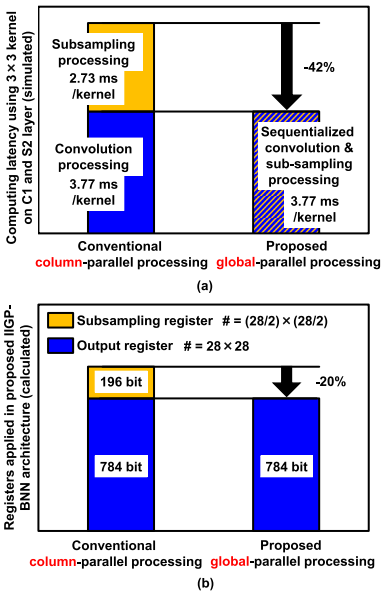

FIGURE 10. Reductions in (a) computing latency and (b) memory demand on registers in conventional column-parallel processing and proposed global-parallel processing architecture.

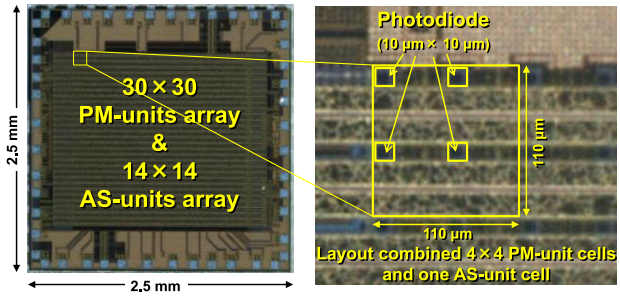

FIGURE 11. Proposed IC with integrated imager and proposed IIGP-BNN accelerator.

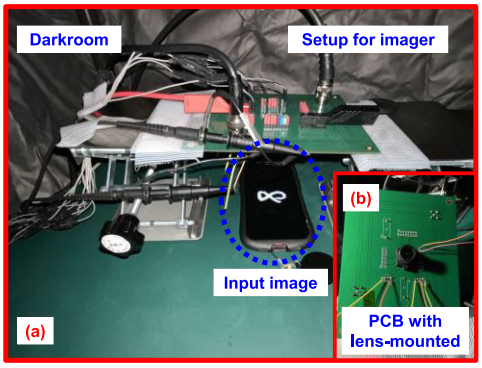

FIGURE 12. (a) Setup inside the darkroom platform for imager measurement. (b) PCB board mounted with lens.

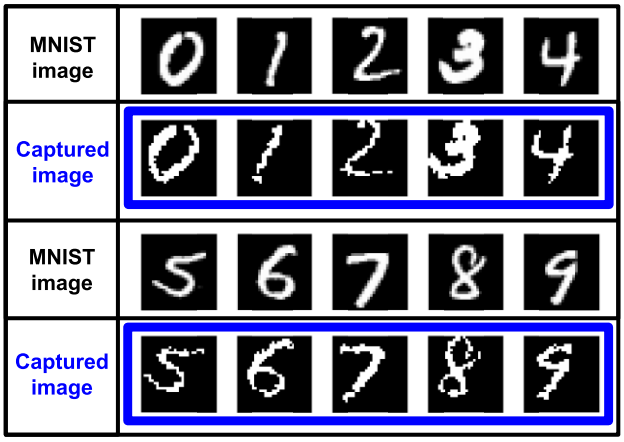

FIGURE 13. Original images and captured binary images.

Fig. 9 exhibits the timing diagram of the convolution operation for a single kernel as in Fig. 8. The duration from the onset of step-1 to the termination of step-9, as dictated by the clock, is interpreted as the latency for the convolution of a single kernel. The counter resets following the completion of the initialization phase, and the 4-bit step control signal alongside the 9-bit weight shift signal are initiated during the reset of the scan-FFs. The multiplication outcome is transferred into scan-FFs, subsequently introducing a clock into the counter as the scan-FFs transform into a shift register. This pattern repeats for the ensuing nine steps without resetting the counter. Following step-9, the activation outcomes are stored in FFs within each AS-unit. This timing diagram can be applied to the prototypical 30 $\times$ 30 IIGP-BNN circuit and is also adaptable for larger-scale circuits.

Figures 10 (a) and (b) depict the simulated computing latency and the memory capacity demand on registers for accelerators, employing both the traditional column-parallel processing approach and the newly proposed global-parallel processing concept, respectively. The application of the proposed architecture results in a 42% reduction in computing latency, and a 20% decrease in the memory capacity demand on registers.

III. MEASUREMENT RESULTS

The proposed 30 $\times$ 30 integrated circuits (IC), combining the integrated imager and the IIGP-BNN accelerator, was fabricated using a 0.18 μm mixed-signal CMOS process. Fig. 11 presents the chip microphotograph. The die size is 2.5 mm $\times$ 2.5 mm. The imager circuit, incorporated into the accelerator IC, employs pixels of dimensions 55 μm $\times$ 55 μm. The layout of a single AS-unit is nested within four PM-units, enabling them to equally share a 110 μm $\times$ 110 μm space. To validate the functionality of the proposed IIGP-BNN architecture, a darkroom platform was established, demonstrating the imager and digital computing circuits respectively. The measurement in this work applied ADCMT 6240A as the DC voltage source, Keithley

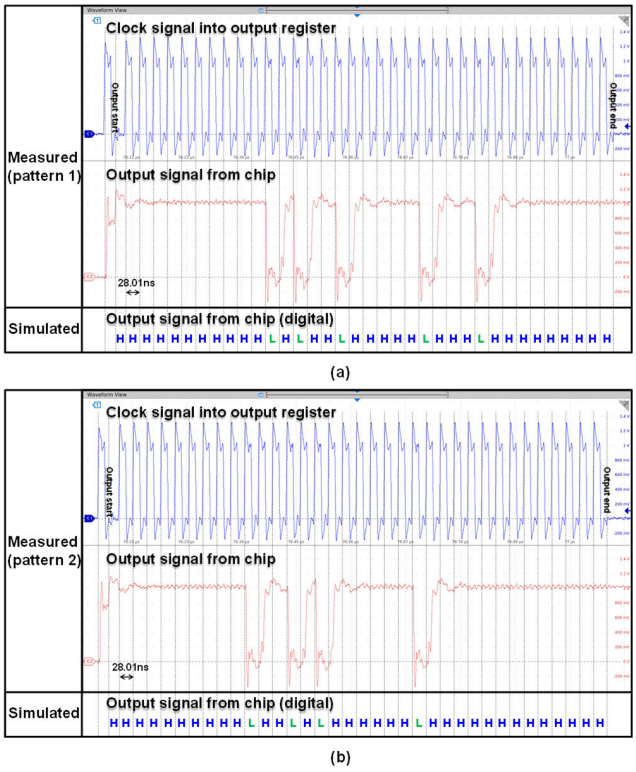

FIGURE 14. Measured output waveforms from a chip under (a) input pattern 1 and (b) input pattern 2.

2000 multimeter as the ammeter, and Tektronix MSO58 as the oscilloscope.

A. MEASUREMENT OF IMAGER CIRCUIT IN PROPOSED IC

A darkroom setup has been established to validate the imager circuit applied in the proposed IIGP-BNN accelerator IC, as depicted in Fig. 12. Images representative of numerals 0 through 9, derived from the MNIST dataset, have been stored in PNG format and displayed on a smartphone screen (Fig. 12 (a)). The printed circuit board (PCB) side, on which a camera lens (screw size: M12; focal length: 2.8mm; angle of view: 115 degrees) is mounted over the chip, is directed towards the screen (Fig. 12 (b)). The separation between the lens and the screen can be appropriately adjusted by altering the height of both two stands. The imager outputs raw data serially via the shift register, connected by FFs of the PM-units, and these data are captured and analyzed using a pattern analyzer. Fig. 13 exhibits the original and binary images of numerals 0 to 9, as captured by the imager circuit. This imager circuit (refer to Fig. 5), integrated into the proposed fully digital IIGP-BNN accelerator, was tested under a supply voltage (V_{DD}) of 1V, pixel exposure time of 20 ms, and a reference voltage (V_{REF}) of 0.843V. Binary images of numbers from 0 to 9 can be captured successfully at a frame rate of 50 fps. It's worth noting that the imager circuit utilized in this work could be substituted with any other imager circuit without impacting the functionality of the proposed IIGP-BNN accelerator circuit.

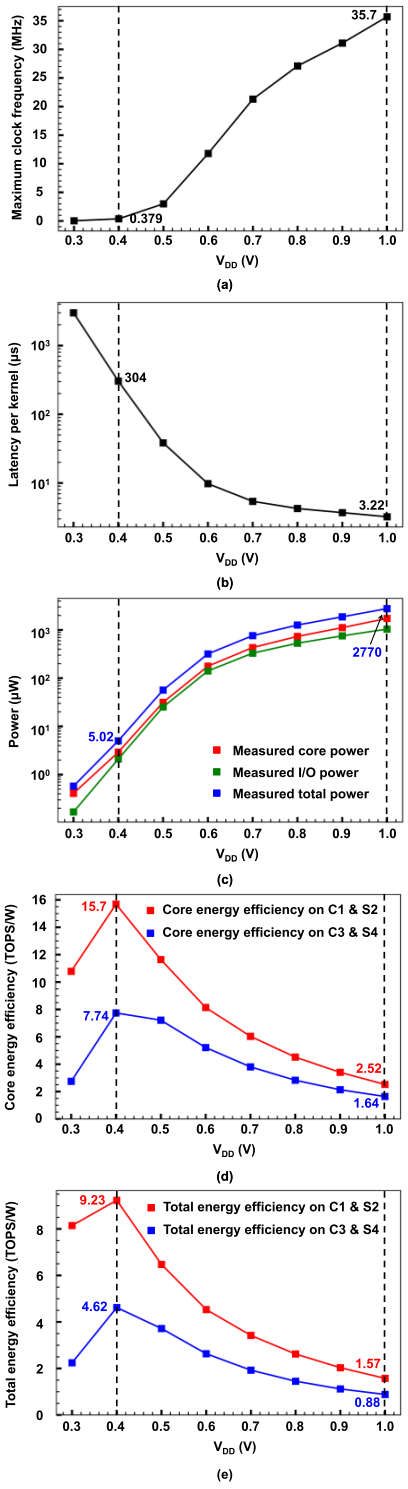

FIGURE 15. Measured V_{DD} dependencies of (a) maximum clock frequency, (b) latency per kernel, (c) power consumption, (d) core energy efficiency, and (e) total energy efficiency of proposed IIGP-BNN accelerator IC.

B. MEASUREMENT OF COMPUTING CIRCUIT IN PROPOSED IC

The digital signal inputs (including weight signal inputs) and outputs received from the chip were generated and assessed using the PXIe-6570 digital pattern generator from National Instruments. The chip's output signal was captured, then analyzed using the waveform capture function on Digital Pattern Editor software, and then verified through Python programs. Fig. 14 (a) and (b) display two sample waveforms recorded by the oscilloscope under different input signal patterns. The waveforms were evaluated with a V_{DD} of 1V, as well as a peak clock frequency of 35.7 MHz. The output signal data is distinguished by the clock signal on the output shift register, which represents computed results after the S4 layer in Fig. 3. The output signal levels (0 or 1) from the chip are in complete match with the simulated results, which indicates the validity of the proposed IIGP-BNN accelerator IC.

Figs. 15 (a) through (e) showcase the measured V_{DD} (ranging from 0.3V to 1V) dependencies of the maximum clock frequency, the latency per kernel, power consumption, core energy efficiency, and total energy efficiency (inclusive of I/O power) of the proposed IIGP-BNN accelerator IC, respectively. At a V_{DD} of 0.4V and a peak frequency of 379 kHz, the latency per kernel on the C1 and S2 layers was calculated to be 304 μs. The energy efficiency on the C1 and S2 layers peaked at 15.7 TOPS/W (core) and 9.23 TOPS/W (total). Similarly, the energy efficiency on the C3 and S4 layers reached peak values of 7.74 TOPS/W (core) and 4.62 TOPS/W (total) at the same V_{DD}. The circuit's power was measured to be 5.02 μW, and the circuit's throughput was calculated to be 46.33 MOPS. At a V_{DD} of 1V, the energy efficiency reached 2.52 TOPS/W (core) and 1.57 TOPS/W (total) on the C1 and S2 layers, with a maximum frequency of 35.7 MHz, a power of 2770 μW, and a throughput of 4.36 GOPS. The latency of the convolution per kernel was calculated as 3.22 μs, an 35.6% reduction compared with state-of-the-art in/near-imager-computing work [12]. The energy efficiency reached 1.64 TOPS/W (core) and 0.88 TOPS/W (total) on the C3 and S4 layers.

IV. DISCUSSION

Considering the information asymmetry on the latency of macro's I/O caused by the huge gap in quantity of data movement between input data and weight data when comparing in-sensor-computing and in-memory-computing due to their feature of input stationary and weight stationary [30], [32], to provide a comprehensive comparison, Table 1 focuses on the results of recent works on in/near-imager-computing CNN accelerator circuits for image processing. In this work, we proposed a novel global-parallel processing concept to achieve ultralow latency by processing convolution operations in 2D. This concept is successfully verified by the first IIGP-BNN prototyped IC. The prototyped IC completes the first convolutional layer using a 3 $\times$ 3 kernel in only 3.22 μs, reducing the latency/kernel by 35.6% on the first convolutional layer compared with state-of-the-art in/near-imager-computing work [12].

The latency of the BNN model in this work was analyzed under the environment of Anaconda 3 Jupyter Notebook; CUDA version 11.3; CPU i7-11700 @ 2.50 GHz; GPU

TABLE 1. Comparison of Previous In/Near-Imager-Computing Accelerators and Our Accelerator.

Reference	Chen TCAS-I 2019 [11]	Lefebvre ISSCC 2021 [9]	Eki ISSCC 2021 [13]	Hsu JSSC 2021 [12]	Xu TCAS-I 2022 [10]	Abedin JXCDC 2022 [14]	This work	
Architecture	Near imager	In imager	Near imager	Near imager	In imager [h]	In imager	**In imager**	
	Column-parallel	Column-parallel	Column-parallel	Column-parallel	Column-parallel	Column-parallel	**Global-parallel**	
Computing type	Analog	Analog	Digital	Analog	Analog	Analog	Digital	
Technology (nm)	180	65	65/22	180	180	65	180	
Applying 3D stack	No	No	Yes	No	No	No	No	
Application	Binary CNN	CNN	CNN	CNN, etc.	BNN	QNN	Binary CNN	
Input/weight bit accuracy	1 b / 1 b	N/A / 1.5 b	8 b ~ 32 b	3 b / 3 b	Analog / 1 b	Analog / 2b	1 b / 1 b	
Memory for MAC	Yes	Yes	Yes	No	Yes	Yes	**No**	
V_{DD} (V)	2	N/A	2.6 / 1.7 / 0.8	0.5	0.8 ~ 1.8	1.2	0.3 ~ 1	
Power (μW)	1800 @ 2 V	53.2	243460 @ 0.8 V	117 @ 0.5 V	0.148 @ 0.8 V	88	5.02 @ 0.4 V [b]	2770 @ 1 V [b]
Throughput (GOPS) [a c]	0.98 @ 2 V	0.194	1210 @ 0.8 V	N/A	0.003 @ 0.8 V	0.166	0.05 @ 0.4 V	4.36 @ 1 V
Energy efficiency (TOPS/W) [c]	0.545 @ 2 V	3.64	4.97 @ 0.8 V	N/A	17.3 @ 0.8 V	1.89	9.23 @ 0.4 V	1.57 @ 1 V
Accuracy	N/A [d]	RMSE = 4.1% [i]	TOP-1: 70.5% [j]	95.9% [g] @ ResNet-18, CIFAR-10	93.8% @ Binarized MLP model, MNIST	97.26 @QNN model, MNIST	96.0% @ Customized LeNet-5, MNIST	
Clock frequency (MHz)	10 @ 2 V	N/A	262.5 @ 0.8 V	N/A	N/A	N/A	0.379 @ 0.4 V	35.7 @ 1 V
Latency/kernel (μs) [c e]	16.5	2028	N/A	5	15324	N/A	304 @ 0.4 V	**3.22 @ 1 V**
Number of operations/kernel [c f]	16200 @ 3 × 3 kernel, stride = 1	393216 @ 32 × 32 kernel, stride = 8	N/A	285768 @ 3 × 3 kernel, stride = 1	39200 @ 5 × 5 kernel, stride = 1	N/A	14112 @ 3 × 3 kernel, stride = 1	
Array size	32 × 32	160 × 128	4056 × 3040	128 × 128	32 × 32	256 × 256	30 × 30	
Pixel size (μm^2)	40 × 40	9 × 9	1.5 × 1.5	7.6 × 7.6	35 × 35	6 × 6	55 × 55	

a. Throughput is calculated using the function 'Throughput = Energy efficiency × Power'.
b. The power consumption calculation within this paper does not include the power consumption of the digital pattern generator.
c. The comparison was conducted only on the first convolutional layer.
d. The analog circuit computation accuracy accumulates errors no more than 0.25% of full scale.
e. Latency is calculated using the function 'Latency = Number of operations/Throughput'.
f. Number of operations is calculated by the function '1 MAC = 2 ops'. For instance, in Chen's work [11], the calculation on number of operations goes to: Number of operations = (32-3+1) × (32-3+1) × (3 × 3) × 2 = 16200.
g. Hsu's work used a total of 30 000 edge images with six categories that contain five gestures and one random object as the data set.
h. The architecture in Xu's work processes more globally than conventional column-parallel processing works, but also operates differently from the global-parallel processing concept proposed in this paper.
i. No classification accuracy is reported. RMSE is obtained by comparing measured convolution results with software results.
j. Used dataset is reported. The evaluated model is Mobilenet_V1_1.0_224_quant.

GeForce RTX 3060. The MNIST test dataset was applied in the analysis by performing 10 rounds of inference, with each round consisting of 1000 images. The average latency results were obtained as 0.38907 s on the first layer (0.25256 s for the C1 layer and 0.13651 s for the S2 layer), 0.18933s on the second layer (0.08618 s for the C3 layer and 0.10315 s for the S4 layer), 0.00174 s for flatten and 0.028365 s for fully connected (FC) layers. Considering that the form of the data output of this circuit has completed flatten process, the total latency of the complete model can be approximated as 12.88 μs + 202.56 μs + 28365 μs = 28580.44 μs $\approx$ 28.58 ms, (608.505 ms in GPU) which achieves the latency reduction of 99.96% on the convolutional layers (215.44 μs in this work from 578.4 ms in GPU) and 95.30% on the overall system (28.58 ms in this work from 608.505 ms in GPU) compared to GPU.

Another comparison between systems that adopt the concept of processing convolution two-dimensionally instead of row-by-row is presented in TABLE 2. Compared to Eyeriss's works, the prototyped architecture is not compatible with more models at this stage. However, it still shows potential in ultralow latency processing of models with larger maps as well as image-capturing functionality in the same circuit. As the in-sensor-computing architectures serve better as the

TABLE 2. Comparison of Previous Accelerators Employed Concept of Processing Convolution in 2D and Our Accelerator.

Architecture	Eyeriss [27]	Eyeriss v2 [28]	IIGP-BNN (This work)
Computing type	Digital	Digital	Digital
Technology (nm)	65	65	180
Application	AlexNet, VGG-16, MobileNet	sparse AlexNet, sparse MobileNet	BNN (customized LeNet-5)
Chip size	4 mm × 4 mm	N/A	2.5 mm × 2.5 mm
On-chip SRAM (kB)	181.5	246	**No**
Number of PEs	168	192	30 × 30 PM-units, 14 × 14 AS-units
Supply voltage (V)	Core: 0.82 ~ 1.17, I/O: 1.8	N/A	0.3 ~ 1
Max core frequency (MHz)	200	200	35.7
Peak throughput (GOPS)	33.6 ~ 84.0	153.6	4.36
Peak energy efficiency (TOPS/W) [a]	0.1662	0.5817 [b]	9.23 @ 0.4 V
Bit precision	16b	8b	1b
Supported CNN shapes	Programmable	Programmable	30 × 30 sized image, 3 × 3 sized kernel
Processing latency [c]	AlexNet CONV1:16.5 ms	N/A	LeNet-5 CONV1: 12.88 µs
	AlexNet CONV1:39.2 ms	N/A	LeNet-5 CONV2: 202.56 µs
In/near-imager	No	No	**Yes**
Processing dataflow	Row stationary	Row stationary	Input stationary
Pixel size (μm^2)	N/A	N/A	**55 × 55**

a. Number of operations is calculated by the function '1 MAC = 2 ops'.
b. The parameter is inferred and calculated from the following statement in paper of Eyeriss v2: "Overall, Eyeriss v2 with sparse MobileNet is 12.6× faster and 2.5× more energy efficient than Eyeriss v1 with MobileNet."
c. The comparison of processing latency in this context is not an apples-to-apples comparison, considering the differences in models and depth of the kernels in the first and second convolutional layers.

interface between sensing and other high-level processing units due to one of its key features of input stationary [7], [30], the possibility of combining these works by utilizing 3D stack technology is under consideration, which could additionally reduce the latency associated with both image data load and data processing. The proposed IIGP-BNN architecture was all-digital instead of mixed-signal or all-analog in this work, which is capable to fabricate the circuits using the finer CMOS process to achieve better performance on both latency and energy efficiency.

Our global-parallel processing concept processes input image data globally in 2D, postulating that this method could theoretically manage 4K or even 8K image data with identical convolution latency per kernel on the initial convolutional layer. Additionally, when employing this global-parallel processing concept, the throughput is estimated to experience a meteoric increase by over 10000× and over 40000× when processing 4K and 8K resolution image data, respectively. This innovative feature guarantees ultralow latency for convolutional processing, even at ultrahigh resolution.

In the experimental phase, a total of 10,000 images from the MNIST dataset were utilized to evaluate the processing accuracy of the algorithm implemented in the proposed system. By employing fixed-size 3 × 3 kernels, mean-pooling, and a binary tanh activation function, the proposed model achieved a maximum accuracy of 96.0%. This was made possible by using three full-precision fully connected layers with 200, 120, and 10 nodes, as illustrated in Fig. 3.

The prototyped IC in this work was evaluated and designed to apply a simple BNN model because of the limitations of the TSMC 0.18 μm CMOS process and single chip size within 2.5 mm × 2.5 mm. The circuit size was minimized to solely operate binary convolutional processing while ensuring the verification of the two key elements, namely in-imager (II) and global-parallel processing (GP) in this work. The pixel circuit in this work performs binary quantization on the output, which limits the applicability of the prototypical IC to BNN models merely. Applying the prototyped IC directly to certain scenarios in the case of colored and grayscale images would result in significant accuracy loss [33]. Therefore, based on the successful verification of the both two key elements (II and GP), the envisioned expansion of the proposed IIGP-BNN architecture includes incorporating compatibility with pixel circuits that are capable of multi-bit precision output. Meanwhile, the proposed architecture is also projected to be expanded with multibit precision processing and deeper convolutional layers to cater to more intricate CNN models in future research endeavors including applying 3D-stack technology [24], [34], [36]. Considering the complexity of wiring caused by a large number of multipliers and multi-bit precision processing for higher resolutions such as 1080p and 4k, trade-offs may be done in terms of fill factor. However, this compromise can be significantly mitigated by the finer CMOS process as mentioned above. An apparent trade-off exists between on-chip memory demand and chip size, which we foresee being resolved by leveraging 3D-stack technologies to stack memory layers [7], [34], [35]. Our proposed accelerator currently supports convolution operations using fixed-size 3 × 3 kernels exclusively, and subsampling operations are also applied in a fixed function. This restricts the

diversity and accuracy of models compatible with our accelerator. Hence, the development of a programmable architecture for both kernel size and subsampling function is a key objective for future research.

V. CONCLUSION

This study presents a prototyped IC including a CMOS image sensor and a binary convolutional neural network accelerator implemented inside the pixel array in a 0.18-μm CMOS technology. The main novelty of this work is the proposal of the global-parallel processing concept, which parallelize the convolution operation not only at the level of several rows of pixels, but at the level of the whole pixel array, therefore leading to a latency decreased by 35.6% compared to the state of the art [12]. Importantly, our IIGP-BNN architecture could theoretically be extended to high-resolution image processing such as 4K or even 8K while maintaining the same latency. This suggests a potential reduction in computing latency of over 99.9% compared with conventional architectures, as discussed in relation to Fig. 2. Such performance makes it feasible to meet the requirements for real-time image processing, thus presenting a significant advancement in the field.

REFERENCES

[1] K.-C. Chen, Y.-W. Huang, G.-M. Liu, J.-W. Liang, Y.-C. Yang, and Y.-H. Liao, "A hierarchical K-means-assisted scenario-aware reconfigurable convolutional neural network," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 1, pp. 176–188, Jan. 2021.

[2] S. D. Manasi, F. S. Snigdha, and S. S. Sapatnekar, "NeuPart: Using analytical models to drive energy-efficient partitioning of CNN computations on cloud-connected mobile clients," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 8, pp. 1844–1857, Aug. 2020.

[3] B. Murmann, "Mixed-signal computing for deep neural network inference," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 1, pp. 3–13, Jan. 2021.

[4] Y. Chen, L. Lu, B. Kim, and T. T. Kim, "Reconfigurable 2T2R ReRAM architecture for versatile data storage and computing in-memory," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 12, pp. 2636–2649, Dec. 2020.

[5] Z. Lin, H. Zhan, X. Li, C. Peng, W. Lu, X. Wu, and J. Chen, "In-memory computing with double word lines and three read ports for four operands," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 5, pp. 1316–1320, May 2020.

[6] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 μ J/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, Jan. 2019.

[7] F. Zhou and Y. Chai, "Near-sensor and in-sensor computing," *Nature Electron.*, vol. 3, no. 11, pp. 664–671, Nov. 2020.

[8] L. Bose, P. Dudek, J. Chen, S. Carey, and W. Mayol-Cuevas, "A camera that CNNs: Towards embedded neural networks on pixel processor arrays," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1335–1344.

[9] M. Lefebvre, L. Moreau, R. Dekimpe, and D. Bol, "A 0.2-to-3.6TOPS/W programmable convolutional imager SoC with in-sensor current-domain ternary-weighted MAC operations for feature extraction and region-of-interest detection," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 118–120.

[10] H. Xu, N. Lin, L. Luo, Q. Wei, R. Wang, C. Zhuo, X. Yin, F. Qiao, and H. Yang, "Senputing: An ultra-low-power always-on vision perception chip featuring the deep fusion of sensing and computing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 1, pp. 232–243, Jan. 2022.

[11] Z. Chen, H. Zhu, E. Ren, Z. Liu, K. Jia, L. Luo, X. Zhang, Q. Wei, F. Qiao, X. Liu, and H. Yang, "Processing near sensor architecture in mixed-signal domain with CMOS image sensor of convolutional-kernel-readout method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 2, pp. 389–400, Feb. 2020.

[12] T.-H. Hsu, Y.-R. Chen, R.-S. Liu, C.-C. Lo, K.-T. Tang, M.-F. Chang, and C.-C. Hsieh, "A 0.5-V real-time computational CMOS image sensor with programmable kernel for feature extraction," *IEEE J. Solid-State Circuits*, vol. 56, no. 5, pp. 1588–1596, May 2021.

[13] R. Eki, S. Yamada, H. Ozawa, H. Kai, K. Okuike, H. Gowtham, H. Nakanishi, E. Almog, Y. Livne, G. Yuval, E. Zyss, and T. Izawa, "A 1/2.3inch 12.3 Mpixel with on-chip 4.97TOPS/W CNN processor back-illuminated stacked CMOS image sensor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 154–156.

[14] M. Abedin, A. Roohi, M. Liehr, N. Cady, and S. Angizi, "MR-PIPA: An integrated multilevel RRAM (HfOx)-based processing-in-pixel accelerator," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 8, pp. 59–67, 2022.

[15] T. Zhang, "Toward automated vehicle teleoperation: Vision, opportunities, and challenges," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11347–11354, Dec. 2020.

[16] T. Burger. (Feb. 9, 2015). *How Fast is Realtime? Human Perception and Technology*. [Online]. Available: https://www.pubnub.com/blog/how-fast-is-realtime-human-perception-and-technology/

[17] S. Tiku, P. Kale, and S. Pasricha, "QuickLoc: Adaptive deep-learning for fast indoor localization with mobile devices," *ACM Trans. Cyber-Phys. Syst.*, vol. 5, no. 4, pp. 1–30, Oct. 2021.

[18] J. Cong and B. Xiao, "Minimizing computation in convolutional neural networks," in *Proc. Int. Conf. Artif. Neural Netw.*, 2014, pp. 281–290.

[19] S. S. Lee, T. D. Nguyen, P. K. Meher, and S. Y. Park, "Energy-efficient high-speed ASIC implementation of convolutional neural network using novel reduced critical-path design," *IEEE Access*, vol. 10, pp. 34032–34045, 2022.

[20] A. Ardakani, C. Condo, M. Ahmadi, and W. J. Gross, "An architecture to accelerate convolution in deep neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1349–1362, Apr. 2018.

[21] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "14.5 envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm FDSOI," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2017, pp. 246–247.

[22] C. Zhang and V. Prasanna, "Frequency domain acceleration of convolutional neural networks on CPU-FPGA shared memory system," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, Feb. 2017, pp. 35–44.

[23] F. Sahin, "Long-range, high-resolution camera optical design for assisted and autonomous driving," *Photonics*, vol. 6, no. 2, p. 73, Jun. 2019.

[24] T. Yamazaki, H. Katayama, S. Uehara, A. Nose, M. Kobayashi, S. Shida, M. Odahara, K. Takamiya, Y. Hisamatsu, S. Matsumoto, L. Miyashita, Y. Watanabe, T. Izawa, Y. Muramatsu, and M. Ishikawa, "A 1ms high-speed vision chip with 3D-stacked 140GOPS column-parallel PEs for spatio-temporal image processing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 82–83.

[25] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, Apr. 2017.

[26] A. Sayal, S. S. T. Nibhanupudi, S. Fathima, and J. P. Kulkarni, "A 12.08-TOPS/W all-digital time-domain CNN engine using bi-directional memory delay lines for energy efficient edge computing," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 60–75, Jan. 2020.

[27] 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.

[28] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," 2018, *arXiv:1807.07928*.

[29] P. J. Koopman, Jr., "The future of stack computers," in *Stack Computers: The New Wave*. Chichester, U.K.: Ellis Horwood, 1989, pp. 171–182.

[30] J. Emer, V. Sze, and Y.-H. Chen, "DNN accelerator architectures," in *Proc. Int. Symp. Comput. Archit. (ISCA) Tutorial*, 2019, pp. 15–43. [Online]. Available: https://www.rle.mit.edu/eems/wp-content/uploads/2019/06/Tutorial-on-DNN-05-DNN-Accelerator-Architectures.pdf

[31] Z. Zhang, X. Zhao, X. Zhang, X. Hou, X. Ma, S. Tang, Y. Zhang, G. Xu, Q. Liu, and S. Long, "In-sensor reservoir computing system for latent fingerprint recognition with deep ultraviolet photo-synapses and memristor array," *Nature Commun.*, vol. 13, no. 1, p. 6590, Nov. 2022.
[32] X. Yang, Y. Hou, and H. He, "A processing-in-memory architecture programming paradigm for wireless Internet-of-Things applications," *Sensors*, vol. 19, no. 1, p. 140, Jan. 2019.
[33] R. Ding, H. Liu, and X. Zhou, "IE-Net: Information-enhanced binary neural networks for accurate classification," *Electronics*, vol. 11, no. 6, p. 937, Mar. 2022.
[34] J. Gao, J. Zhu, K. Nie, and J. Xu, "An image inpainting method for interleaved 3D stacked image sensor," *IEEE Sensors J.*, vol. 19, no. 24, pp. 12253–12260, Dec. 2019.
[35] T. Haruta, T. Nakajima, J. Hashizume, T. Umebayashi, H. Takahashi, K. Taniguchi, M. Kuroda, H. Sumihiro, K. Enoki, T. Yamasaki, K. Ikezawa, A. Kitahara, M. Zen, M. Oyama, H. Koga, H. Tsugawa, T. Ogita, T. Nagano, S. Takano, and T. Nomoto, "A 1/2.3inch 20Mpixel 3-layer stacked CMOS image sensor with DRAM," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 76–77.
[36] T. Miura, M. Sakakibara, H. Takahashi, T. Taura, K. Tatani, Y. Oike, and T. Ezaki, "A 6.9 μm pixel-pitch 3D stacked global shutter CMOS image sensor with 3M cu-cu connections," in *Proc. Int. 3D Syst. Integr. Conf. (3DIC)*, Oct. 2019, pp. 1–2.

RUIZHI WANG received the B.S. degree in optoelectronic information science and engineering from Beijing Jiaotong University, Beijing, China, in 2017, and the M.Eng. degree in electronic engineering from The University of Tokyo, Japan, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include integrated circuits for high energy efficiency and high-speed computing.

CHENG-HSUAN WU received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2017, and the M.S. degree in electronic engineering from The University of Tokyo, Japan, in 2020. In 2020, he joined the Research and Development Division, TSMC, Hsinchu, where he worked in the field of SRAM IP design. In 2021, he joined the IP Division of Novatek, Hsinchu, where he is currently working in the field of analog front-end (AFE) circuit design of high-speed interface IPs.

MAKOTO TAKAMIYA (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from The University of Tokyo, Japan, in 1995, 1997, and 2000, respectively. In 2000, he joined NEC Corporation, Japan, where he was engaged in the circuit design of high speed digital LSI's. He joined The University of Tokyo, in 2005, where he is currently a Professor with the Institute of Industrial Science. From 2013 to 2014, he stayed with the University of California at Berkeley, Berkeley, as a Visiting Scholar. His research interests include the digital gate driver and sensor ICs for power electronics and the integrated power management circuits for automotive and industrial applications. He was an elected member of Administrative Committee of IEEE Solid-State Circuits Society, from 2023 to 2025. He is a member of the Technical Program Committee of IEEE Symposium on VLSI Technology and Circuits, IEEE Asian Solid-State Circuits Conference, and IEEE International Symposium on Power Semiconductor Devices and ICs. He formerly served on the technical program committees for IEEE Custom Integrated Circuits Conference, from 2006 to 2011, and IEEE International Solid-State Circuits Conference (ISSCC), from 2015 to 2020. He was a Far East Regional Chair in ISSCC 2020. He was a Distinguished Lecturer of IEEE Solid-State Circuits Society, from 2019 to 2020. He received 2009 and 2010 IEEE Paul Rappaport Awards and the Best Paper Award, in 2013, and IEEE Wireless Power Transfer Conference.