2025 IEEE International Symposium on Circuits and Systems (ISCAS) | 979-8-3503-5683-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISCAS56072.2025.11043165

A Real-Time Memory-Less In-Sensor Time-Domain Convolution Processor with Programmable Kernel for Feature Extraction

Harshit Naman, Gourab Barik, Shreyas Sen
School of Electrical and Computer Engineering, Purdue University, West Lafayette, USA
Email: hnaman@purdue.edu

***Abstract*—With the growing demand for artificial intelligence (AI) and the Internet of Things (IoT), there is an increasing need for smart vision sensors and cameras with energy-efficient computing capabilities. While previous works have explored in-sensor computing for feature extraction, they often rely on memory-based weight storage or fixed kernels, limiting their flexibility and energy efficiency. This paper introduces a low-power, programmable, memory-less convolution engine designed for feature extraction in the analog domain. The proposed engine utilizes a linear large-signal voltage-to-current converter-based Time-Domain (TD) multiply-accumulate (MAC) cell, employing time pulses as weights. A 32-phase subsampling phase-locked loop (SS-PLL) is implemented to generate 5-bit Time Domain weights for a programmable 3x3 kernel employed for feature extraction. The in-sensor convolution engine achieves an energy efficiency of 0.4 pJ/pixel at a data rate of 300 MSps, making it suitable for resource-constrained, battery-operated devices.**

***Index Terms*—Artificial intelligence (AI), convolution, feature extraction, multiply-accumulate (MAC), processing in sensor (PIS), in-sensor computing, time domain computing**

INTRODUCTION

Rapid advancement of deep neural networks (DNN) and artificial intelligence (AI) has led to a wide array of applications in image recognition and classification, now integral to smart surveillance, autonomous vehicles, and medical diagnostics as shown in Fig. 1(a). This has driven the development of sophisticated image processing techniques demanding high computational power and efficiency. Although high-quality image sensors have improved resolution and quality, the resulting surge in data has led to exponential increases in the energy consumption of analog-to-digital conversion (ADC) [1]. The vast data access required for multiply accumulate (MAC) operations in computing algorithms results in considerable power consumption and latency, creating critical challenges for integrating intelligent networks into power-constrained Internet of Things (IoT) devices [2] [3].

To address these issues, the concept of in-sensor feature extraction is gaining traction. As illustrated in Fig. 1(c), this approach contrasts with traditional computation by utilizing Time-Domain in-sensor Feature Extraction (TD-FE). Instead of employing high-resolution ADCs and transmitting full-resolution data at high frame rates, TD-FE technology leverages early-stage image processing for feature extraction,

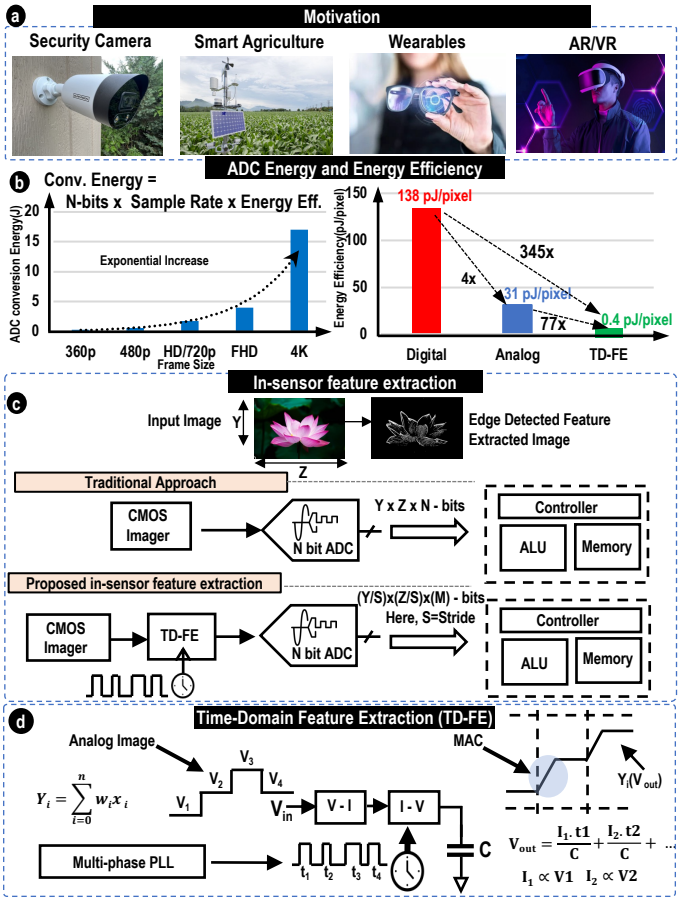

Fig. 1. Overview of the analog-to-digital conversion energy challenges and the benefits of Time-Domain in-sensor Feature Extraction (TD-FE).

thereby reducing data transmission and computational workload. For example, a gray-level image sensor with dimensions $Y \times Z$ and N-bit output typically requires $Y \times Z \times N$-bit data transfer for a single kernel convolution in conventional architectures. In contrast, a TD-FE circuit that executes a 3x3 convolution reduces the output data size to $\frac{Y}{S} \times \frac{Z}{S} \times M$ bit (where the stride $= S$ with an M bit ADC), achieving bandwidth and energy savings by a factor of $S \times S \times 2^{N-M}$

[4].

Furthermore, modern DNNs require billions of MAC operations per inference. Given that these computations demand relatively low precision, analog computing becomes feasible, offering greater efficiency than digital methods in low signal-to-noise ratio (SNR) regimes. Hence, TD-FE based on computing in the analog time domain shows better energy efficiency results than SOA analog, digital equivalent circuits, as shown in Fig. 1(b) [5].

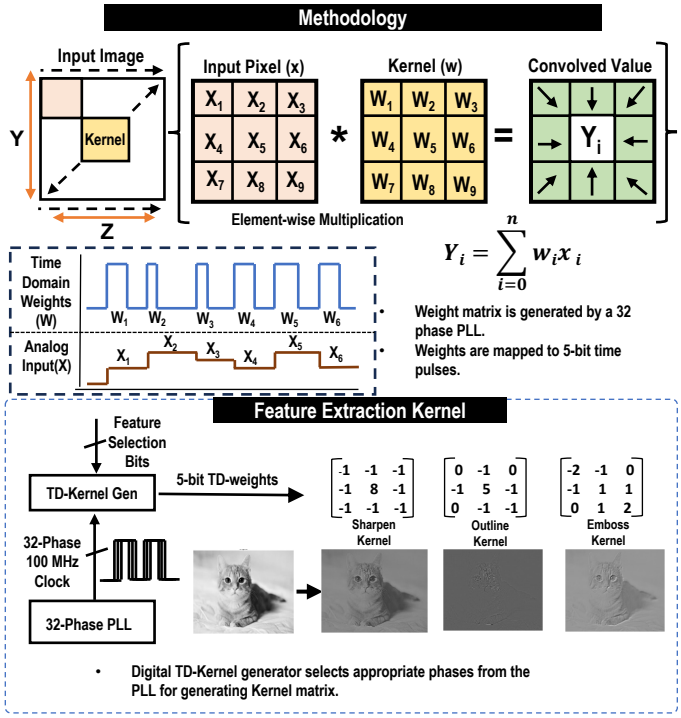

Fig. 2. Methodology and Illustration of various kernel matrices used for different applications in time-domain kernel generation.

I. Methodology

A. Background on Feature Extraction with Convolution

Feature extraction using convolution is a fundamental technique in image processing and computer vision, particularly in the context of convolutional neural networks (CNNs). Convolution involves sliding a filter or kernel across an image to produce a feature map. This process helps identify and extract relevant features, such as edges, textures, and patterns, from input data [6].

B. Time-Domain Kernel Generation

The weight matrix, or kernel, is selected based on specific application requirements, as illustrated in Fig. 2, which shows various kernels for different applications. These kernel matrices are normalized and mapped to time pulses ranging from 310 ps to 5 ns with 5-bit accuracy using the multiphase clock generated by a 100-MHz phase-locked loop (PLL). The time-domain kernel generation circuit selects the appropriate clock phases from the multiphase PLL and utilizes the phase differences between these clocks to generate time pulses that are mapped to the kernel matrix. This kernel matrix can be adjusted according to the specific feature that needs to be extracted. The time pulses are generated in real-time and used directly for multiply-accumulate (MAC) operations. This approach is faster and more energy efficient compared to storing weights in memory and fetching them for MAC operations.

C. Multiphase PLL for Efficient Weight Generation

A key innovation in this design is the utilization of a low-power (60 µW) 32-phase phase-locked loop (PLL) to generate weights in the time domain to create the kernel matrix. This approach offers significant advantages over traditional SRAM-based weight storage.

1) Energy Efficiency: By eliminating the need to retrieve weights from memory for each multiply-accumulate (MAC) operation, the system achieves substantial energy savings.
2) Improved speed: The direct generation of weights through phase differences eliminates the latency associated with memory access, resulting in a faster overall operation. The weights are generated and used for convolution directly; the PLL provides the weights with a frequency of 100 MHz, enhancing the overall speed of the system.
3) Real-time Adaptability: The PLL-based approach allows for dynamic weight adjustment, enabling the system to adapt to different kernel requirements in real-time using a digital FSM.

II. Circuit Implementation

A. TD-MAC Unit Cell Design and Operation

The unit cell TD-MAC (Time-Domain Multiply-Accumulate), as depicted in Fig. 3, is an analog computing element designed for efficient multiplication and accumulation operations. At its core, the unit cell employs a large signal Voltage-to-Current (V-to-I) converter implemented using a source-degenerated native NMOS device (threshold voltage $\sim$ 0mV).

The source degeneration technique enhances linearity by introducing negative feedback. The larger the degeneration resistor, the greater is the linearity improvement. However, resistor sizing involves a careful trade-off between linearity and input dynamic range.

The converted current is reflected using a current-mirror configuration. The M3 transistor plays a crucial role in increasing the output resistance seen by the capacitor, making the reflected current less sensitive to voltage fluctuations across the capacitor. The native input device (M0) is intentionally small in size to minimize input capacitance and facilitate easy cascading of multiple MAC units.

Switches S0 and S1 gate the current of the MAC unit, activating it only during the integration phase. This design choice ensures that the circuit remains off for the majority of the MAC operation, resulting in high energy efficiency.

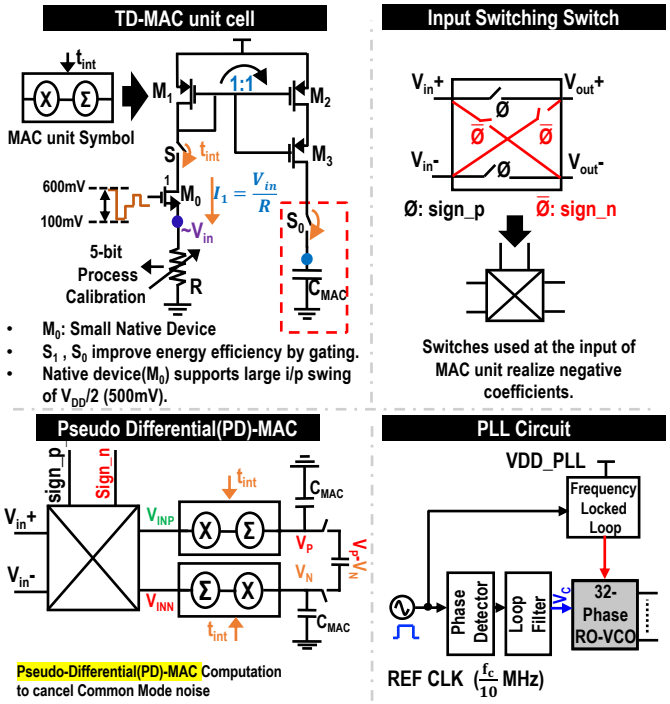

Fig. 3. Design and operation of the TD-MAC unit cell, highlighting the pseudo-differential architecture for handling positive and negative weights.

To compensate for process variations, the resistor is equipped with a 5-bit trimming control. This allows for fine-tuning of the resistor value to maintain consistent performance across process corners.

The TD-MAC unit cell operates as follows: The analog input voltage (V_{in}) is converted to a current (V_{in}/R) by the V-to-I converter. This current is then integrated into the capacitor C_{MAC} for a specific integration time (t_{int}), performing the multiplication. The process is repeated for n cycles to realize n accumulations. Each clock cycle, a new input V_{in} is fetched from the image sensor for MAC operation.

The output voltage (V_{out}) of the TD-MAC unit cell can be expressed by the following equation:

$$V_{out} = \frac{V_{in1} \cdot t_{int1} + V_{in2} \cdot t_{int2} + V_{in3} \cdot t_{int3} + \cdots}{R \cdot C_{MAC}} \quad (1)$$

This equation demonstrates how the TD-MAC unit cell performs multiple accumulations over time, with each term representing a single integration cycle. The output voltage is the sum of these individual integrations, where R and C_{MAC} remain constant, while V_{in} and t_{int} may vary for each cycle.

B. Pseudo-Differential Architecture for Negative Weights

To accommodate kernel matrices with both positive and negative weights, the TD-MAC unit employs a pseudo-differential architecture, as illustrated in Fig. 3 (bottom left). This design allows for the realization of negative integration(for negative kernel weights) and offers additional benefits in noise reduction.

For positive weights, the input voltage is applied to V_{in+}, while V_{in-} is connected to a common mode voltage. In contrast, for negative weights, the connections are reversed: V_{in-} receives the input voltage, and V_{in+} is tied to the common-mode voltage. This configuration enables the circuit to handle both positive and negative weights effectively.

The pseudo-differential architecture offers two significant advantages:

1) Facilitates the cancellation of common-mode noise, improving the signal-to-noise ratio of the system.
2) It helps mitigate common-mode clock feedthrough, improving the overall accuracy of MAC operations.

The final output of the TD-MAC unit is obtained by taking the differential voltage between the two branches:

$$V_{out} = V_P - V_N \quad (2)$$

A 32-phase PLL is used to generate the multi-phase clock for integration time generation.

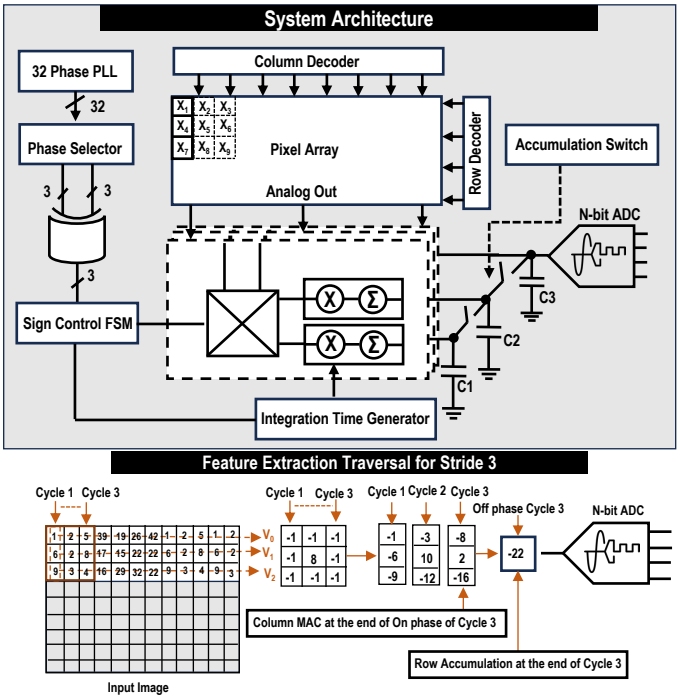

Fig. 4. System architecture of the proposed time-domain feature extraction circuit, including the role of the 32-phase PLL in integration time generation.

III. System Architecture

The proposed architecture, depicted in Fig. 4, features a time domain feature extraction circuit (TD-FE) that utilizes a 32-phase phase-locked loop (PLL) as a 5-bit memory. This design leverages the phase differences between various clocks to store kernel values. The phase selector chooses the appropriate phases from the PLL, which are then used by the integration time generator to produce integration pulses. These pulses facilitate multiply accumulate (MAC) operations

directly, enhancing speed and energy efficiency compared to traditional methods that store weights in memory.

The sign-control finite state machine (FSM) manages input switching to account for both positive and negative weights in the kernel. The sign of each weight determines the input connection to the pseudo-differential MAC. The capacitors C_1, C_2, and C_3 store accumulation results for each row of the kernel. During each clock cycle's positive phase, these capacitors accumulate multiplication results corresponding to elements in a column of the kernel. At the end of the third clock cycle, the capacitors are shorted in the off-phase of the clock, resulting in the accumulation of individual row MACs. This process is illustrated in Fig. 4(bottom), showcasing the feature extraction calculation.

The pixel values are fed into the convolution engine column-wise, with the column size determined by the kernel size (e.g., size 3 for a 3x3 kernel, size 5 for a 5x5 kernel). The analog voltages corresponding to these pixel values are sent to the time-domain multiply-accumulate (MAC) units. These units multiply the analog voltage by the time pulse and store the result in capacitors. Once the last column of the kernel is processed and accumulated, the capacitors are shorted to perform a row-wise accumulation. The final value of the analog voltage is then sent to the analog-to-digital converter (ADC) for digitization.

IV. Results and Comparison

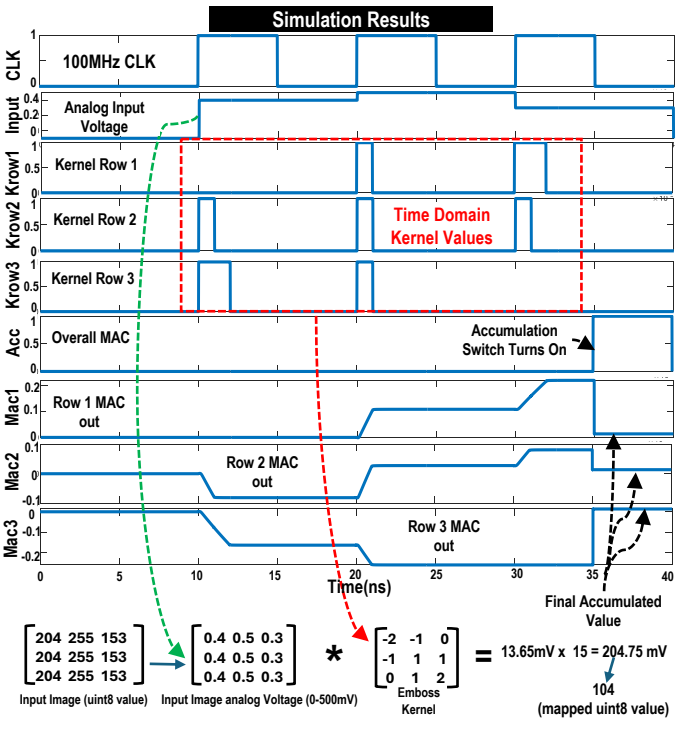

Fig. 5. Simulation results showing the 100 MHz clock, analog input image, kernel time pulses, and MAC operations.

The proposed architecture is simulated using an input analog image, as illustrated in Fig. 5. The simulation results include a 100 MHz clock shown in the first part of the figure, followed by the analog input image. Kernel-time pulses and corresponding multiply-accumulate (MAC) operations are also highlighted. The final voltage values are normalized to $\frac{1}{15}$, which can be adjusted by a shift and add operation after the analog-to-digital conversion (ADC). The ideal output for the simulated kernel is 13.33 mV, while the circuit achieves an output of 13.65 mV.

Comparison Table

	ISSCC '17[4]	VLSI '18[5]	VLSI '17[6]	ISSCC '17[7]	JSSC '21[2]	This Work
Feature	Haar-like filtering	Edge detection	Spatial-temporal processing	Spatial-temporal processing	ED, Sharpening	ED, Sharpening, Embossing, Outline
Power	23.8uW	12.7uW	29.94uW	363mW	117uW	120uW
Energy Efficiency	309 pJ/pixel	31pJ/pixel	134pJ/pixel	286pJ/pixel	14.8pJ/pixel	0.4pJ/pixel
Process	65nm	180nm	180nm	60nm	180nm	65nm
Memory	Yes	Yes	Yes	Yes	No	No
Data Rate	77 KSps	409 KSps	223 KSps	1.26 GSps	8 MSps	300 MSps
Weight	Fixed	Fixed	Fixed	Fixed	Programmable	Programmable

Fig. 6. Comparison table.

This architecture supports features such as edge detection, sharpening, embossing, and outlining. It achieves an energy efficiency of 0.4 pJ/pixel, significantly lower than comparable works, making it suitable for energy-constrained applications. The programmability of weights offers flexibility in adapting to various tasks.

The proposed architecture offers a highly energy-efficient and programmable solution for real-time feature extraction, achieving significant improvements in power consumption and flexibility compared to existing methods [7] [8] [9] [10].

V. Conclusion

The proposed architecture presents a novel real-time in-sensor time-domain convolution processor with a programmable kernel for feature extraction, offering significant advancements in energy efficiency and computational flexibility. By utilizing a time-domain approach with a linear large-signal voltage-to-current converter-based MAC cell and a 32-phase subsampling PLL, the proposed architecture achieves an energy efficiency of 0.4 pJ/pixel at a 300 MSps data rate. The system's programmability allows for a variety of feature extraction tasks, including edge detection, sharpening, embossing, and outlining. Compared to existing methods, the proposed solution provides substantial improvements in power consumption and adaptability, making it suitable for energy-constrained applications such as IoT devices and smart vision sensors.

REFERENCES

[1] K Gaurav Kumar, Gourab Barik, Baibhab Chatterjee, Sumon Bose, Shovan Maity, and Shreyas Sen. A 65 nm 2.02 mW 50 Mbps Direct Analog to MJPEG Converter for Video Sensor Nodes using low-noise Switched Capacitor MAC-Quantizer with automatic calibration and Sparsity-aware ADC. In *2023 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–2, 2023.

[2] Ningyuan Cao, Saad Bin Nasir, Shreyas Sen, and Arijit Raychowdhury. Self-Optimizing IoT Wireless Video Sensor Node With In-Situ Data Analytics and Context-Driven Energy-Aware Real-Time Adaptation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(9):2470–2480, 2017.

[3] Ningyuan Cao, Baibhab Chatterjee, Jianbo Liu, Boyang Cheng, Minxiang Gong, Muya Chang, Shreyas Sen, and Arijit Raychowdhury. A 65 nm Wireless Image SoC Supporting On-Chip DNN Optimization and Real-Time Computation-Communication Trade-Off via Actor-Critical Neuro-Controller. *IEEE Journal of Solid-State Circuits*, 57(8):2545–2559, 2022.

[4] Tzu-Hsiang Hsu, Yi-Ren Chen, Ren-Shuo Liu, Chung-Chuan Lo, Kea-Tiong Tang, Meng-Fan Chang, and Chih-Cheng Hsieh. A 0.5-V Real-Time Computational CMOS Image Sensor With Programmable Kernel for Feature Extraction. *IEEE Journal of Solid-State Circuits*, 56(5):1588–1596, 2021.

[5] Boris Murmann. Mixed-Signal Computing for Deep Neural Network Inference, year=2021. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(1):3–13.

[6] Manjunath Jogin, Mohana, M S Madhulika, G D Divya, R K Meghana, and S Apoorva. Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 2319–2323, 2018.

[7] Kyeongryeol Bong, Sungpill Choi, Changhyeon Kim, Sanghoon Kang, Youchang Kim, and Hoi-Jun Yoo. 14.6 A 0.62mW ultra-low-power convolutional-neural-network face-recognition processor and a CIS integrated with always-on haar-like face detector. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 248–249, 2017.

[8] Xiaopeng Zhong, Qian Yu, Amine Bermak, Chi-Ying Tsui, and May-Kay Law. A 2PJ/Pixel/Direction MIMO Processing Based CMOS Image Sensor for Omnidirectional Local Binary Pattern Extraction and Edge Detection. In *2018 IEEE Symposium on VLSI Circuits*, pages 247–248, 2018.

[9] Kyuseok Lee, Seokjun Park, Sung-Yun Park, Jihyun Cho, and Euisik Yoon. A 272.49 pJ/pixel CMOS image sensor with embedded object detection and bio-inspired 2D optic flow generation for nano-air-vehicle navigation. In *2017 Symposium on VLSI Circuits*, pages C294–C295, 2017.

[10] Tomohiro Yamazaki, Hironobu Katayama, Shuji Uehara, Atsushi Nose, Masatsugu Kobayashi, Sayaka Shida, Masaki Odahara, Kenichi Takamiya, Yasuaki Hisamatsu, Shizunori Matsumoto, Leo Miyashita, Yoshihiro Watanabe, Takashi Izawa, Yoshinori Muramatsu, and Masatoshi Ishikawa. 4.9 A 1ms high-speed vision chip with 3D-stacked 140GOPS column-parallel PEs for spatio-temporal image processing. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 82–83, 2017.