

Single-Slope ADC with Embedded Convolution Filter for Global-Shutter CMOS Image Sensors

Miron Kłosowski, Yichuang Sun, *Senior Member, IEEE*, Waldemar Jendernalik, Grzegorz Blakiewicz, Jacek Jakusz and Stanisław Szczepański

Abstract—This paper presents an analog-to-digital converter (ADC) suitable for acquisition and processing of images in the global-shutter mode at the pixel level. The ADC consists of an analog comparator, a multi-directional shift register for the comparator states, and a 16-bit reversible binary counter with programmable step size. It works in the traditional single-slope mode. The novelty is that during each step of the reference ramp, neighboring pixels can exchange status information. During the conversion, the direction and step size of the counter are set globally to realize the corresponding coefficient of a convolution kernel. This technique does not slow down the conversion when used for small kernels (3×3) and does not significantly increase sensor noise. Convolution windows of arbitrary size can be implemented. The concept was verified in an experimental 64×64 imaging array implemented in 180 nm CMOS technology.

Index Terms—CMOS image sensor, global shutter, focal-plane processing, pixel-level processing, single-slope analog-to-digital converter, vision chip, energy efficient convolution filter.

I. INTRODUCTION

SINGLE-SLOPE analog-to-digital converters (ADCs) are used in CMOS image sensors (CISs) due to their simple electrical topology and compact layout. These features allow a large number of converters to be integrated together with a pixel array to create efficient, parallel video processing architectures. There are three such architectures, namely the classic column-parallel where one converter handles one column of a pixel array [1]–[5], group-parallel in which one converter serves a pixel sub-array [6], [7], and the pixel-parallel (massively-parallel) where each pixel is integrated with its own converter [8]–[15] (Fig. 1(a)). The arrangement of Fig. 1(a) has the advantage that even slow ADCs can acquire thousands of image frames per second in global

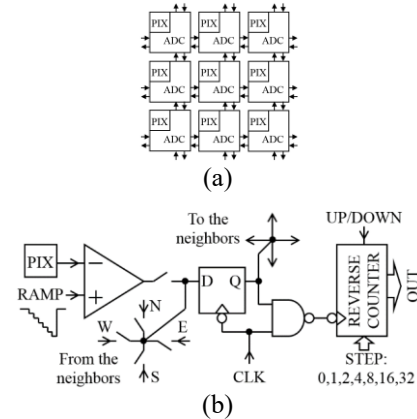


Fig. 1. Massively-parallel imager: (a) portion of a pixel-parallel ADC array, (b) the proposed ADC idea.

shutter mode. Moreover, it is possible to establish local data-exchange connections and to process the images already in ADCs. In [1], [2] and [8] it was shown that single-slope ADCs can perform image pre-conditioning such as compensation of dark-signal-nonuniformity via digital CDS [1], [2] in global shutter mode [8], as well as photo-response-nonuniformity compensation through a special clock-stopping technique [8], [13]. In this paper, a single-slope ADC solution is proposed, which provides more advanced image processing, namely convolution filtering in global shutter mode.

The pixels of the proposed ADC are interconnected as shown in Fig. 1(a). This 2-D shift register allows for a quick exchange of comparator states between pixels before changing the RAMP level. Thus, different kernel coefficients can be realized by pulsing the clock with the count direction signals (UP/DOWN) set to the sign of the current coefficient and the STEP signals set to its absolute value (Fig. 1(b)). The final content of the counter holds the result of the convolution. The advantage of the proposed convolution implementation is the lack of latency, because it is executed while waiting for the analog path to process the next ramp level. Moreover, no analog signals are transmitted between pixels, so the convolution operation is not affected by extra noise. The shifts may be performed to any distance and direction, so that kernels of arbitrary size can be established.

II. DETAILS OF THE PIXEL ADC

A detailed schematic diagram of the pixel ADC is depicted in Fig. 2(a). The analog front-end (PG, TG, RG transistors and comparator) is a typical circuit used in classic pixels with single-slope ADCs and MOS photosensors [8], [13]–[15]. A two-stage comparator with a differential pair has been used.

Manuscript received xx November 2022; revised xx Xxx 202X; accepted xx Xxxx 202x. Date of publication xx Xxxx 202x; date of current version xx Xxxx 202x.

This work was supported in part by the National Science Centre of Poland under Grant 2016/23/B/ST7/03733. (Corresponding author: M. Kłosowski).

Miron Kłosowski, Waldemar Jendernalik, Grzegorz Blakiewicz, Jacek Jakusz and Stanisław Szczepański are with the Faculty of Electronics Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland (e-mail: miron.klosowski@pg.edu.pl).

Yichuang Sun is with the School of Engineering and Computer Science, University of Hertfordshire, Hatfield, Herts AL10 9AB, United Kingdom (e-mail: y.sun@herts.ac.uk).

Color versions of one or more of the figures in this article are available online at <https://doi.org/10.1109/TCSII.202X.xxxxxxx>

Digital Object Identifier 10.1109/TCSII.202X.xxxxxxx

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

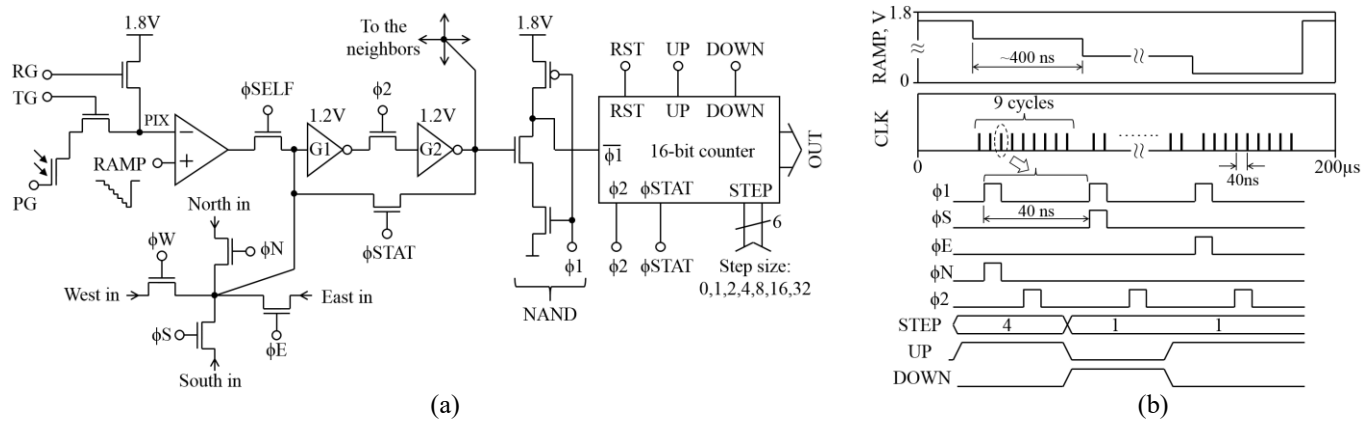


Fig. 2. Proposed “convolutional” ADC: **(a)** detailed diagram, **(b)** waveforms for clock pulses 3, 4, and 5 in Table II.

The comparator output drives a dynamic latch based on the inverters G1 and G2. This latch works with a two-phase clock. The first phase can be generated by one of the five global clock signals: ϕ_{SELF} , ϕ_{N} , ϕ_{E} , ϕ_{S} , or ϕ_{W} . The second phase is generated by the ϕ_2 clock. The clock signal pulsed in the first phase determines the stage of the kernel processing. At the beginning, ϕ_{SELF} is pulsed, which captures the current state of the comparator. Then, the ϕ_2 and ϕ_{STAT} clocks are activated to get rid of any metastable state. Now, the inverter G2 holds the representation of the center pixel. The logic 1 is captured at the beginning of AD conversion (after a pulse on RG or TG), and when logic 0 is captured (after change of the RAMP voltage) the counter stops, representing the center pixel value. Other kernel coefficients are calculated using the comparator states from the neighboring pixels. This data is acquired by quickly shifting the latch value to/from the neighboring pixels (the RAMP voltage is constant during those shifts) using ϕ_{N} , ϕ_{E} , ϕ_{S} and ϕ_{W} global clocks. For the quick shift operation, the latches work in a dynamic mode i.e. ϕ_{STAT} is not used. After every shift the counter is pulsed using ϕ_1 clock, and when logic 1 is captured the counter adds the contribution of the next kernel coefficient. Hence, during each RAMP step, a full “walk” in the pixel window is carried out, along with appropriate clock and counter action.

Suppose the pixel window (**P**) and the kernel mask (**K**) are

$$\mathbf{P} = \begin{bmatrix} P_{NW} & P_N & P_{NE} \\ P_W & P_{SF} & P_E \\ P_{SW} & P_S & P_{SE} \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \end{bmatrix} \quad (1)$$

hence, the convolution is

$$\mathbf{P} \times \mathbf{K} = P_{NW}k_9 + P_Nk_8 + P_{NE}k_7 + P_Wk_6 + P_{SF}k_5 + \dots \quad (2)$$

The sign of kernel coefficients k_i ($i = 1 \dots 9$) is represented by the activity of the global signal UP (for $k_i > 0$) and DOWN (for $k_i < 0$). The selected values of $|k_i|$ can be realized simply in a single clock pulse using the STEP signals (Fig. 3(a)).

$$|k_i| = n \text{ if } \text{STEP}_n = 1 \quad n \in \{1, 2, 4, 8, 16, 32\} \quad (3)$$

For the other k_i values, the counting step must be repeated, but without moving the shift registers. For example, if $|k_i|$ is 5,

it can be realized in two counter steps: 4 and 1. For $k_i = 0$ all global STEP signals can be set to 0, but it is advised to try to bypass zero coefficients during the shift process.

The realization of the 3×3 convolution (2) typically takes 9 clock cycles in each RAMP step (Fig. 2(b)). The clock cycle consists of a pulse on ϕ_{SELF} , ϕ_{N} , ϕ_{E} , ϕ_{S} or ϕ_{W} followed by a pulse on ϕ_2 (ϕ_1 followed by ϕ_2 for the counter). If there are zero coefficients, the number of shifts can be optimized to reduce the number of clocks. If there are coefficients whose values are not countable with a single counter step, subtraction can also be utilized for optimization, e.g. 7 can be implemented as STEP_8/UP and $\text{STEP}_1/\text{DOWN}$.

The shift and counter clock frequency can be higher than a clock driving RAMP DAC (which is limited by a response time of analog ramp distribution network and in-pixel comparators). Thus, the shifting and counting processes can be fast enough not to slow down the AD conversion. The optimal RAMP step time was experimentally determined to be about 400 ns (shortening this time deteriorates the picture quality). The number of RAMP steps is 512. All digital inputs are 1.8V.

The counter (Fig. 3(a)) consists of sixteen identical stages shown in Fig. 3(b). The signals ϕ_2 , ϕ_{STAT} , RST, UP and DOWN are common to all stages. The BYPASS signal is only used in the first five stages.

The implementation of dead pixel compensation in the presented imager is difficult because it requires additional programmable connections between pixels.

A. CDS

The linearity of the convolution (2) allows for digital CDS realization in the reversible counter using the superposition rule [1], [2], [8]. First, after resetting the PIX sense node by RG pulse, the mask with negated coefficients ($-\mathbf{K}$) is realized, then after a pulse on TG, the mask \mathbf{K} is applied.

$$\text{Convol and CDS} \equiv \mathbf{P}_{\text{reset}} \times (-\mathbf{K}) + \mathbf{P}_{\text{photo}} \times \mathbf{K} \quad (4)$$

III. CONVOLUTION PROGRAM AND OPTIMIZATION

An example of the successive calculation for the sharpen kernel is presented in Table I. In this example, no optimization has been performed and the program takes 11 clock cycles.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

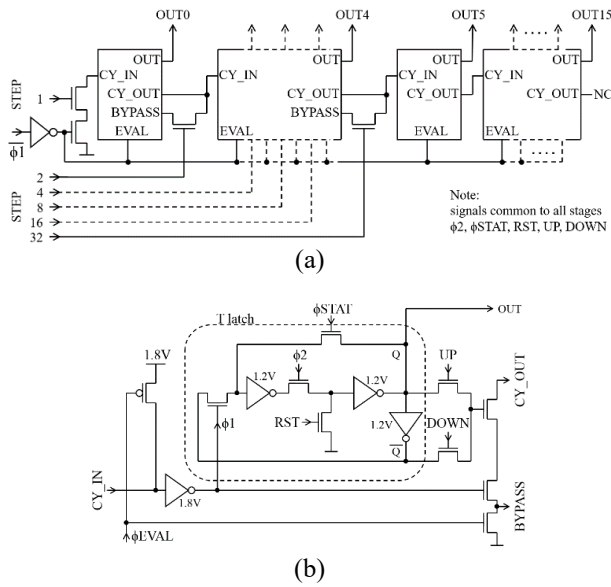


Fig. 3. 16-bit counter: (a) stage-level diagram, (b) the stage.

Note that to obtain a pixel shift in N direction, ϕS must be pulsed and vice versa. The same applies to the E-W direction. Also, there is 1 clock latency between the shift and the count operation realized on the result of that shift.

In Table II the number of steps has been reduced thanks to the observation that we can visit the center pixel more than once and the additional step needed for realization of coefficient 5 can be performed by the way (clock pulse 5). Additionally, if one pixel shift of the resulting image is not a problem, we can start from other than the center pixel, which allows for a further reduction of clocks (clock pulse 2). If the shifting of the final image is not allowed, one more clock pulse (9 pulses in total) will be needed.

A. Edge handling

The convolution operation presented in the paper requires special treatment of signals passing through the edges of the pixel matrix. There is no such treatment in the implemented imager. The edge crossing output signals are not used (open) and the edge crossing input signals are grounded. This interferes with the shift operation (data is lost) and therefore the resulting image is distorted close to the boundaries.

One possible solution is to use dummy pixels with only the shift register functionality (G1, G2 and switches only). Analog circuits and the counter are not needed. The $\phi SELF$ switch can be connected to the circuit defining the level of the dummy pixel. Fig. 4 presents schematic diagrams of dummy pixels set to black level (a) or edge level (extend method) (b).

In this paper, crop edge handling method has been used, therefore the presented images are smaller (60×60) than the full array size (64×64).

IV. EXPERIMENTAL RESULTS

The proof-of-concept 64×64 pixel-parallel ADC array was realized in the integrated circuit (Fig. 5(a)) in the standard $0.18\text{-}\mu\text{m}$ 1P6M CMOS process of ams AG (austriamicrosystems). The pixel ADC size is $55\text{ }\mu\text{m} \times 55\text{ }\mu\text{m}$ (Fig. 5(b)).

TABLE I
THE PROGRAM FOR THE SHARPEN KERNEL (NAÏVE)

Clock pulse	Shift direction	Count step	Count direction	Convolution progress	Abstract walk ^(*)
1	SELF	0	don't care	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix}$
2	no shift	4	UP	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
3	N (ϕS)	1	UP	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
4	W (ϕE)	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
5	S (ϕN)	0	don't care	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
6	S (ϕN)	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
7	E (ϕW)	0	don't care	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
8	E (ϕW)	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
9	N (ϕS)	0	don't care	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
10	N (ϕS)	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
11	don't care	0	don't care	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$

^(*) ■ A pixel currently counted (a pixel at the output of G2)

TABLE II
THE PROGRAM FOR THE SHARPEN KERNEL (OPTIMIZED)

Clock pulse	Shift direction	Count step	Count direction	Convolution progress	Abstract walk ^(*)
1	SELF	0	don't care	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix}$
2	S (ϕN)	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \blacksquare & \square \end{bmatrix}$
3	S (ϕN)	4	UP	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
4	N (ϕS)	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 4 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
5	W (ϕE)	1	UP	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 5 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
6	E (ϕW)	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
7	E (ϕW)	0	don't care	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$
8	don't care	1	DOWN	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \square & \square & \square \\ \square & \blacksquare & \square \\ \square & \square & \square \end{bmatrix}$

^(*) ■ A pixel currently counted (a pixel on the output of G2)

Fig. 6 (a)–(d) presents the results of image processing using the following kernels, respectively:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \quad (5)$$

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

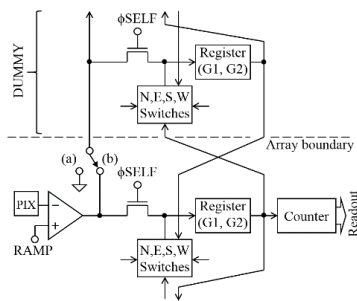


Fig. 4. Example of edge handling.

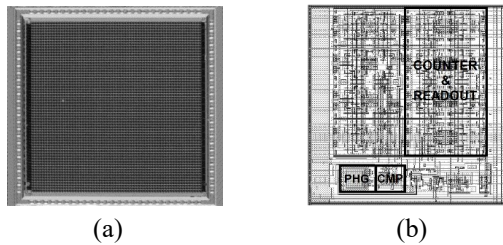


Fig. 5. The chip: (a) photo, (b) pixel layout (Cadence).

Measurements of the imaging array with variable uniform irradiation were performed. The average pixel response of the imager is presented in Fig. 7. The graph shows that the sensor is linear in a wide range and the response does not depend on the selected kernel. For the sharpen kernel the response is distorted near saturation because of excessive noise.

In Fig. 8 the measured fixed-pattern noise (FPN) of the imager is shown. The FPN depends on the type of the kernel and the intensity of illumination, as expected. In addition, the FPN measured for the identity kernel is slightly higher than the noise of the imager working with no convolution. This proves that the shift clocks ($\phi_{N,E,S,W}$) have little effect on the FPN increase.

The measured energy per conversion for the digital part of a single pixel in dark conditions is: 17.6 pJ for the conversion without convolution, 41 pJ for the identity kernel, 135 pJ for the blur kernel, and 281 pJ for the sharpen kernel. The same measured in bright conditions (250 mW/m^2 , 625 nm) is: 35.2 pJ for the conversion without convolution, 64.5 pJ for the identity kernel, 281 pJ for the blur kernel and 393 pJ for the sharpen kernel. The analog circuits consume a constant power of about 220 nW per pixel.

Convolution kernel size is not limited, but above 3×3 it may affect the maximum frame rate. E.g. for the 5×5 gaussian blur kernel, the ramp step duration must be increased to 1.06 μs . This reduces the maximum frame rate from 1000 fps to around 600 fps (including readout).

The identity, box blur, sharpen, and ridge 3×3 kernels need 2, 10, 8, and 10 clock cycles, respectively. The 5×5 gaussian blur kernel needs 35 clock cycles (period reduced to 30 ns).

A. Comparison

The traditional operation scheme, in which images are successively captured, digitized (ADCs), processed and sent from a chip, is intuitive and allows the individual operations to be independently optimized. However, data stream of digitized images is large and implies common problems, namely data

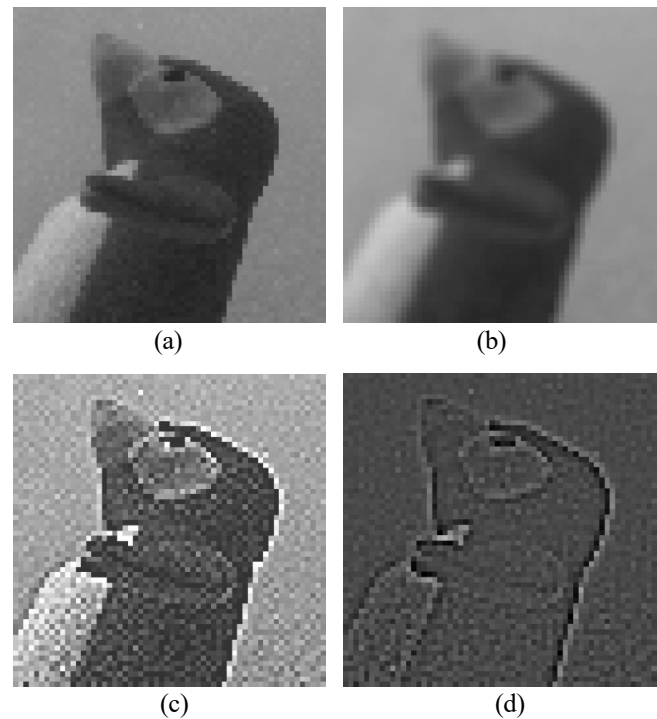


Fig. 6. Images obtained with 3×3 convolution kernels: (a) identity, (b) box blur, (c) sharpen, (d) ridge (edge detection).

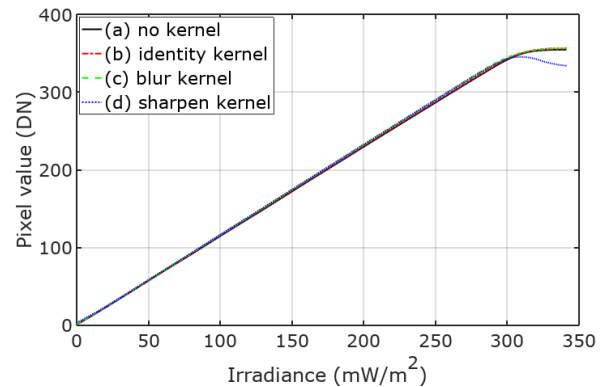


Fig. 7. Pixel response measured for image acquisition with no convolution (a) and for image acquisition with 3×3 convolution kernels: identity (b), Gaussian blur (c), sharpen (d).

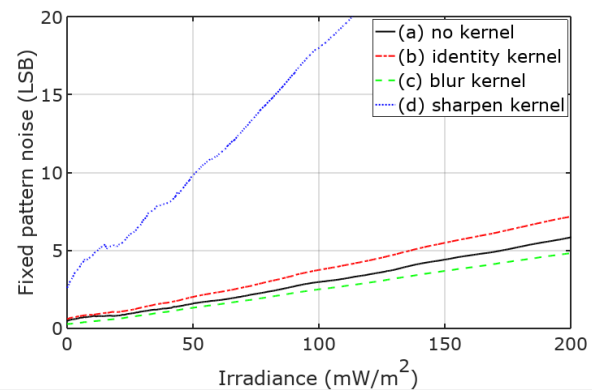


Fig. 8. FPN measured for image acquisition with no convolution (a) and for image acquisition with 3×3 convolution kernels: identity (b), Gaussian blur (c), sharpen (d).

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

throughput bottleneck and/or excessive power consumption. As a result, a frame rate for image processing may be even twice smaller than for acquisition only (e.g. the frame rate of [7] in Table III). To maintain a high frame rate, the capturing, digitization and processing stages can be “overlapped” on a timeline, as in [6]. However, such solution needs the expensive technology of stacked chips which physically separates the noisy digital processors from the sensor’s analog front-end. In [12], digital bottleneck was overcome by not digitizing the images before processing. Processing is performed directly on analog samples, which allows for a very high frame rate up to 100,000 fps at a single watt of power. The convolutional sensors [4] and [16] also do not digitize images, however here the goal was the maximum reduction in power, below 1 mW. This was achieved by the direct processing of photocurrents, nevertheless such approach limits the frame rate to 100–250 fps. This work uses a different approach, namely image digitization and convolutional processing are totally mixed—are performed at the same time and on the same hardware. The solution is a compromise giving 1000 fps at less than 10 mW. The fill factor is low (10.9%) because the designed pixel (Fig. 5(b)) contains extra circuits intended for other project. Without them, the pixel size can be reduced to $35\ \mu\text{m} \times 35\ \mu\text{m}$, resulting in the 26% fill factor, better than other non-stacked imager solutions.

TABLE III
COMPARISON OF COMPUTATIONAL IMAGE SENSORS

	[7]	[6]	[12]	[4]	[16]	This work
Technology	0.13 μm	0.13 μm stacked	0.18 μm	65 nm	0.35 μm	0.18 μm
Supply, V	2.5/1.2	1.2	1.8/1.5	1.2/0.8	3.3	1.8/1.2
Pix array	80×64	1024×768	256×256	160×128	64×64	64×64
Shutter	global	rolling	global	rolling	global	global
CDS	digital	ana. DS	–	–	–	digital
Grayscale	8b	9b at 5.5 kfps	8b	8b	analog	9b
DR, dB	–	54	–	47.1	58	49
Parallelism	group	group	full	column	full	full
FPS: with process.	545–808	5500 ⁽¹⁾	100k	24–268	10–100	1000
acq. only	906	5500	–	–	–	1000
Power, mW	36	720	1230	0.20	0.28	6.75 (2.3 ⁽³⁾)
pJ/pixel/frame	7000	2618	188	2.5–104	684	1647(557 ⁽³⁾)
Pix pitch, μm	39.6	12	32.3	9	35	55 (35 ⁽²⁾)
Fill factor, %	12	75	6.2	12.9	23	10.9 (26 ⁽²⁾)

⁽¹⁾ Using the on-timeline “overlap” of operations (Mode 1) at 0.05 Mpix.

⁽²⁾ Expected after layout optimization (explained in the text). ⁽³⁾ Array (core) only.

V. CONCLUSION

The proposed solution is suitable for single-chip and stacked-chip global-shutter CISs and vision chips, especially low power. The presented CIS can be used in many applications, such as intelligent image sensors, IoT devices, security and surveillance systems, etc. For example it can be used at the first stage of deep neural networks for image processing. If CDS is implemented in analog circuitry it is also possible to perform multiple convolutions on the same image by repeating the RAMP signal run after array readout without reacquiring the image from the photosensors.

REFERENCES

- [1] T. Toyama *et al.*, “A 17.7Mpixel 120fps CMOS image sensor with 34.8Gb/s readout,” *2011 IEEE International Solid-State Circuits Conference*, 2011, pp. 420–422, doi: 10.1109/ISSCC.2011.5746379.
- [2] S. Son, S. Jeon, S. Namgung, J. Yoo and M. Song, “A one-shot digital correlated double sampling with a differential difference amplifier for a high speed CMOS image sensor,” *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 1054–1057, doi: 10.1109/ISCAS.2015.7168818.
- [3] T. Yamazaki *et al.*, “A 1ms high-speed vision chip with 3D-stacked 140GOPS column-parallel PEs for spatio-temporal image processing,” *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 82–83, doi: 10.1109/ISSCC.2017.7870271.
- [4] 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,” *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, 2021, pp. 118–120, doi: 10.1109/ISSCC42613.2021.9365839.
- [5] S. Okura *et al.*, “A 3.7 M-Pixel 1300-fps CMOS Image Sensor With 5.0 G-Pixel/s High-Speed Readout Circuit,” in *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 1016–1024, April 2015, doi: 10.1109/JSSC.2014.2387201.
- [6] L. Millet *et al.*, “A 5500-frames/s 85-GOPS/W 3-D Stacked BSI Vision Chip Based on Parallel In-Focal-Plane Acquisition and Processing,” in *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1096–1105, April 2019, doi: 10.1109/JSSC.2018.2886325.
- [7] J. A. Schmitz, M. K. Gharzai, S. Balkir, M. W. Hoffman, D. J. White and N. Schemm, “A 1000 frames/s Vision Chip Using Scalable Pixel-Neighborhood-Level Parallel Processing,” in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 556–568, Feb. 2017, doi: 10.1109/JSSC.2016.2613094.
- [8] M. Kłosowski, W. Jendernalik, J. Jakusz, G. Blakiewicz and S. Szczepański, “A CMOS Pixel With Embedded ADC, Digital CDS and Gain Correction Capability for Massively Parallel Imaging Array,” *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 64, no. 1, pp. 38–49, Jan. 2017, doi: 10.1109/TCSI.2016.2610524.
- [9] M. -W. Seo *et al.*, “2.45 e-RMS Low-Random-Noise, 598.5 mW Low-Power, and 1.2 kfps High-Speed 2-Mp Global Shutter CMOS Image Sensor With Pixel-Level ADC and Memory,” in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1125–1137, April 2022, doi: 10.1109/JSSC.2022.3142436.
- [10] M. Sakakibara *et al.*, “A 6.9- μm Pixel-Pitch Back-Illuminated Global Shutter CMOS Image Sensor With Pixel-Parallel 14-Bit Subthreshold ADC,” in *IEEE Journal of Solid-State Circuits*, vol. 53, no. 11, pp. 3017–3025, Nov. 2018, doi: 10.1109/JSSC.2018.2863947.
- [11] C. Liu *et al.*, “A 4.6 μm , 512×512, Ultra-Low Power Stacked Digital Pixel Sensor with Triple Quantization and 127dB Dynamic Range,” *2020 IEEE International Electron Devices Meeting (IEDM)*, 2020, pp. 16.1.1–16.1.4, doi: 10.1109/IEDM13553.2020.9371913.
- [12] S. J. Carey, A. Lopich, D. R. W. Barr, B. Wang and P. Dudek, “A 100,000 fps vision sensor with embedded 535GOPS/W 256×256 SIMD processor array,” *2013 Symposium on VLSI Circuits*, Kyoto, Japan, 2013, pp. C182–C183.
- [13] M. Kłosowski, “A Power-Efficient Digital Technique for Gain and Offset Correction in Slope ADCs,” *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 67, no. 6, pp. 979–983, June 2020, doi: 10.1109/TCSII.2019.2928183.
- [14] M. Kłosowski, “Hybrid-mode single-slope ADC with improved linearity and reduced conversion time for CMOS image sensors,” *International Journal of Circuit Theory and Applications*, vol. 48, no. 1, pp. 28–41, 2020, doi: 10.1002/cta.2713.
- [15] M. Kłosowski, Y. Sun, “Fixed Pattern Noise Reduction and Linearity Improvement in Time-Mode CMOS Image Sensors,” *Sensors*, vol. 20, no. 20, pp. 5921, 2020, doi: 10.3390/s20205921.
- [16] W. Jendernalik, G. Blakiewicz, J. Jakusz, S. Szczepański and R. Piotrowski, “An Analog Sub-Miliwatt CMOS Image Sensor With Pixel-Level Convolution Processing,” in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 2, pp. 279–289, Feb. 2013, doi: 10.1109/TCSI.2012.2215803.