

Focal-Plane Algorithmically-Multiplying CMOS Computational Image Sensor

Alireza Nilchi, *Student Member, IEEE*, Joseph Aziz, *Student Member, IEEE*, and Roman Genov, *Member, IEEE*

Abstract—The CMOS image sensor computes two-dimensional convolution of video frames with a programmable digital kernel of up to 8×8 pixels in parallel directly on the focal plane. Three operations, a temporal difference, a multiplication and an accumulation are performed for each pixel readout. A dual-memory pixel stores two video frames. Selective pixel output sampling controlled by binary kernel coefficients implements binary-analog multiplication. Cross-pixel column-parallel bit-level accumulation and frame differencing are implemented by switched-capacitor integrators. Binary-weighted summation and concurrent quantization is performed by a bank of column-parallel multiplying analog-to-digital converters (MADCs). A simple digital adder performs row-wise accumulation during ADC readout. A 128×128 active pixel array integrated with a bank of 128 MADCs was fabricated in a $0.35 \mu\text{m}$ standard CMOS technology. The $4.4 \text{ mm} \times 2.9 \text{ mm}$ prototype is experimentally validated in discrete wavelet transform (DWT) video compression and frame differencing.

Index Terms—Block-matrix image transform, CMOS image sensor, focal-plane image processing, multiplying algorithmic ADC.

I. INTRODUCTION

IN THE LAST decade CMOS image sensors have become the dominant video acquisition technology. Due to their moderate cost, CMOS imagers are an attractive choice for multi-sensor applications such as wireless sensor networks. In such applications the high amount of data generated by imagers is expensive to transmit or store. Video processing tasks such as image compression and pattern recognition reduce the output data rate, but are computationally expensive as they employ real-time block-matrix and convolutional image transforms. Various techniques for realizing such transforms in sensory systems have been developed.

Conventionally, dedicated digital signal processors (DSPs) perform spatial image transforms in hand-held digital cameras and camcorders on the focal plane [1], [2] or off it [3]–[6]. They rely on high-throughput architectures to compute spatial weighted sums needed in block-matrix and convolutional transforms. This comes at the cost of dissipated power or silicon area.

Manuscript received March 01, 2008; revised September 01, 2008. Current version published May 28, 2009. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Foundation for Innovation (CFI). Chips were fabricated through the Canadian Microelectronics Corporation (CMC) foundry service.

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4 Canada (e-mail: roman@eecg.toronto.edu).

Digital Object Identifier 10.1109/JSSC.2009.2016693

To overcome the problems associated with dedicated DSPs, block-matrix and convolutional transforms have also been implemented in the analog domain [7]–[11]. The intrinsic parallelism of analog imaging and signal processing architectures yields high computational energy efficiency and integration density, often beyond those of modern digital processors. Capacitor bank analog implementations use charge sharing to compute weighted sum and difference [7]–[9]. Current-mode weighted averaging implementations use zero-latency current-mode addition [10]. Current-mode vector-matrix multiplication [11] architectures employ floating-gate arrays for block-matrix storage and achieve high power efficiency. Purely analog focal-plane image transform implementations require an extra analog-to-digital converter (ADC) to provide the output in the convenient digital format and may suffer from limited accuracy.

A number of versatile digital or analog visual processors have been reported that perform general-purpose video processing, with the generality of computation beyond that of block-matrix and convolutional image transforms [12]–[15]. Digital general-purpose vision processor implementations generally utilize large peripheral silicon area. For example, the pixel array and the ADC in [12] occupy only approximately seven percent of the total die area. Analog implementations necessitate a large pixel size. The pixel pitch in the designs in [13]–[15] is over $70 \mu\text{m}$.

Mixed-signal CMOS imaging and spatial image processing combine the benefits of both analog and digital domains [16], [17]. Analog circuits perform area-efficient and low-power computation directly on the focal plane. Digital components provide the output in the digital format and sustain the accuracy and configurability of such systems. The mixed-signal VLSI implementation in [17] employs a current-mode processing unit and a pipelined MADC in order to efficiently implement block-matrix and convolutional transforms such as discrete cosine transform (DCT), low-pass filtering, and color interpolation and correction, but requires a large FIFO memory which adds a significant overhead to the power dissipation and integration area.

The presented CMOS mixed-signal image sensor computes block-matrix and convolutional transforms of each video frame with programmable digital kernels of up to 8×8 pixels as well as performs frame differencing [18]–[20], both directly on the focal plane, with a compact scalable VLSI implementation. Our approach combines digital-analog multiplication, accumulation and quantization in a single algorithmic analog-to-digital conversion cycle. Therefore, it makes focal-plane computing an intrinsic part of the quantization process and eliminates the

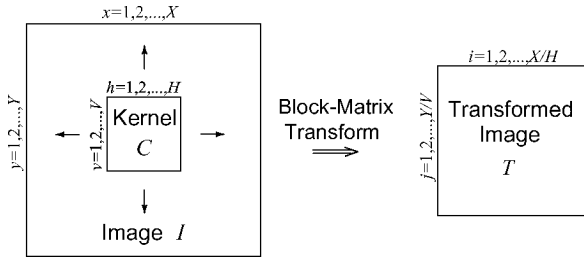


Fig. 1. An illustration of a block-matrix transform computation.

need for a peripheral DSP. In 8×8 block-matrix transforms, the mixed-signal approach yields three computations per pixel readout while power dissipation is almost equivalent to that of a conventional digital imager performing no computation.

The remainder of the paper is organized as follows. Section II gives an overview of block-matrix and convolutional transforms for image processing. Section III presents the top-level VLSI architecture of the algorithmically-multiplying CMOS computational image sensor. Sections IV and V discuss in detail the architectures to perform focal-plane computing in the first and second spatial dimensions. In Section VI the VLSI circuit implementation is presented. Section VII demonstrates the experimental results obtained from a $0.35 \mu\text{m}$ CMOS prototype of the computational image sensor. Section VIII demonstrates the benefits of the proposed approach over a digital imager employing conventional algorithmic ADCs. Section IX concludes the paper.

II. BLOCK-MATRIX AND CONVOLUTIONAL TRANSFORMS

Block-matrix and convolutional transforms correlate a segment of an image with a spatial kernel in order to extract certain features of the image such as horizontal, vertical and diagonal edges and identify statistical redundancies. In the example of image compression these redundancies are eliminated to obtain a lower imager output data rate. In pattern recognition applications the image features are utilized to form a more precise description of the object in order to enhance the classification performance.

To transform an image I into the transformed image T , the kernel C is tiled vertically and horizontally across the image. The kernel is tiled in non-overlapping or overlapping fashion corresponding to block-matrix and convolutional transforms, respectively. For the block-matrix transform case shown in Fig. 1, coefficients of T are obtained by computing the two-dimensional dot product of C and I at each tile location as follows:

$$T_{ij} = \sum_{h=1}^H \sum_{v=1}^V C_{hv} I_{xy}, \quad (1)$$

$$x = h + (i - 1)H, \quad i = 1, 2, \dots, \frac{X}{H}, \quad (2)$$

$$y = v + (j - 1)V, \quad j = 1, 2, \dots, \frac{Y}{V}, \quad (3)$$

where $C_{hv} \in \mathbb{Z}$ are the block-matrix coefficients comprising a spatial kernel; X and Y are the image horizontal and vertical dimensions, assumed for simplicity to be multiples of the kernel sizes H and V ; h and v are the horizontal and vertical block-

matrix indices, and i and j are the indexes of the block-transformed image. Convolutional transforms employ the same computation as in (1) with a different formulation of indexes in (2) and (3). For simplicity, in the remainder of this paper, we will focus on block-matrix transforms.

III. TOP-LEVEL ARCHITECTURE

The block-matrix transform of the form (1) can be decomposed as follows

$$T_{ij} = \sum_{h=1}^H \sum_{v=1}^V C_{hv} I_{xy} = \sum_{h=1}^H T_{ij,h} \quad (4)$$

with partial sums

$$T_{ij,h} = \sum_{v=1}^V C_{hv} I_{xy} \quad (5)$$

where the notation is consistent with that of (2), (3), and I_{xy} is the noise-compensated output of a pixel at location (x, y) .

The proposed mixed-signal VLSI architecture efficiently implements computations (5) and (4), in that order, as depicted in Fig. 2. An array of photodiode-based pixels performs the image acquisition. Row control logic supplies the digital signals necessary for pixel timing and readout. The pixel array is described in detail in Section VI-A. Image fixed-pattern noise (FPN) is suppressed by column-parallel difference circuits.

The computation in (5) corresponds to the most computationally intensive part of the block-matrix transform in (4). It involves pixel-wise signed multiplication and cross-pixel addition in the vertical dimension. This computation is efficiently performed in the mixed-signal VLSI domain in a column-parallel fashion by a sign unit, a binary-analog multiplier, an accumulator and an algorithmic MADC. The sign unit, the binary-analog multiplier, and the accumulator reuse the amplifier of the difference circuit resulting in an area-efficient implementation. Computation and quantization are intrinsically interleaved within a single algorithmic ADC cycle. The MADC output $\hat{T}_{ij,h}$ in Fig. 2 is the digital representation of $T_{ij,h}$ in (5).

The switch matrix routes the kernel coefficients and their corresponding sign values bit-serially from two sets of shift registers, with a sequence period of V values and spatial period of H columns, synchronously with the image readout clock *RowScan* to the multiplication and sign units, respectively. The operation of the switch matrix is discussed in Section V-A.

The summation in the horizontal dimension in (4) is of low computational complexity and is implemented in the digital domain. A simple digital delay-adder loop performs spatial accumulation over H adjacent MADC outputs during readout to yield \hat{T}_{ij} , which is the digital representation of T_{ij} in (4). The digital accumulation is discussed in more detail in Section V-B.

IV. ONE-DIMENSIONAL COMPUTING

This section describes the procedure to compute the inner product in the vertical dimension described by (5), the most expensive computation in the block-matrix transform in (4), on the focal plane of the image sensor in a column-parallel fashion. To simplify the notation, in this section we consider a single image

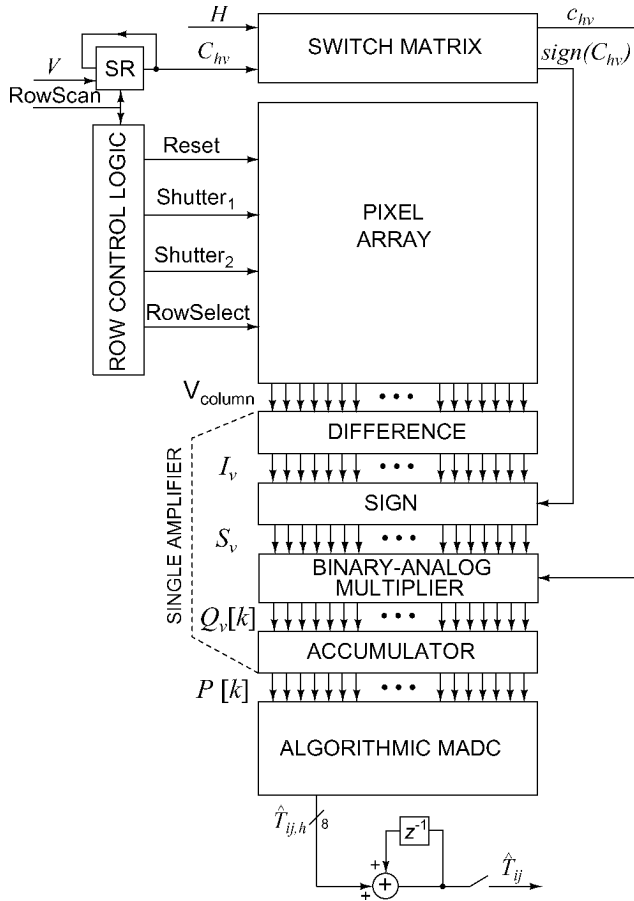


Fig. 2. Top-level architecture of the focal-plane algorithmically-multiplying CMOS computational image sensor. (Digital accumulation is implemented off-chip.)

column segment and the corresponding column of the spatial kernel (i.e., for given i , h and x). This simplifies (5) to

$$T = \sum_{v=1}^V C_v I_v. \quad (6)$$

The signed coefficients C_v in (6) have arbitrary values with a resolution of K bits and can be represented in the binary format as

$$C_v = \text{sign}(C_v) \sum_{k=1}^K 2^{-k} c_v[k]. \quad (7)$$

Substituting (7) into (6) and changing the summation order results in

$$T = \sum_{k=1}^K 2^{-k} P[k] \quad (8)$$

where

$$P[k] = \sum_{v=1}^V Q_v[k], \quad (9)$$

$$Q_v[k] = S_v c_v[k], \quad (10)$$

and

$$S_v = \text{sign}(C_v) I_v. \quad (11)$$

In the equations above, S_v is the sign-transformed pixel output, $Q_v[k]$ is the pixel output multiplied by the corresponding coefficient bit $c_v[k]$, and the partial product $P[k]$ is the sum of the binary-analog multiplication outputs over V adjacent pixels in a column for the same binary weight, k . In the block diagram depicted in Fig. 2, S_v , $Q_v[k]$, and $P[k]$ are the outputs of sign, binary-analog multiplier and accumulator units, respectively. Quantization of the partial products $P[k]$ is efficiently performed by the algorithmic MADCs.

It is worth mentioning that the mixed-signal multiplication of the analog pixel values I_v by the respective digital coefficients C_v in (6) yields the same result as in the case where the pixel values are first quantized by an ADC and the multiplication is performed in the digital domain. The output resolution is equal to that of the MADC.

A. Algorithmic Multiplying ADC

To describe how multiplication in (6) is efficiently implemented in the mixed-signal VLSI domain, in this section we assume a single image row (i.e., $V = 1$) and unsigned coefficients, C . In this case, the transform in (6) simplifies to

$$T' = CI \quad (12)$$

with

$$C = \sum_{k=1}^K 2^{-k} c[k]. \quad (13)$$

Equation (8) becomes

$$T' = \sum_{k=1}^K 2^{-k} P[k] \quad (14)$$

with the partial products in (9) simplifying to

$$P[k] = I c[k]. \quad (15)$$

One approach to perform the digital-analog multiplication of (12) in the mixed-signal domain is to multiply I by $c[k]$, the bits in C , sequentially from MSB to LSB and generate a sequence of analog values with a decremting binary weight. This approach yields a simple VLSI implementation of a multiplier utilizing one two-input multiplexer. As the binary weight of the analog partial products has a descending order, a modified algorithmic quantization scheme is introduced to perform the binary-weighted accumulation in (14), concurrently with the quantization of the inner product in (12).

Fig. 3(a) shows the block diagram and the transfer characteristic of a conventional algorithmic ADC. It samples the input signal during the first cycle, extracts a single bit and feeds back the residue cyclically until all the required bits are extracted. Since each residue is one binary weight less than the previous

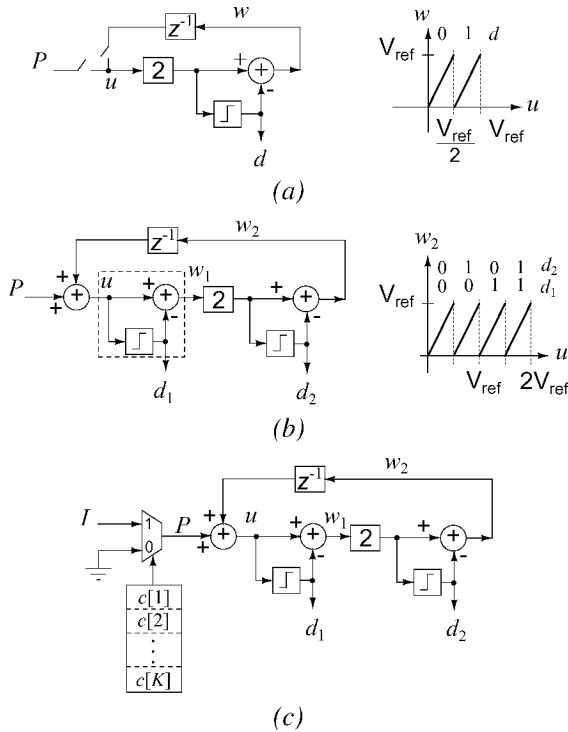


Fig. 3. Block diagram and transfer characteristics of (a), conventional algorithmic ADC; and (b), algorithmic multiplying ADC. (c) Implementation of the digital-analog multiplication and concurrent quantization utilizing the architecture in (b).

one, a gain-of-2 block is used to maintain the comparator input signal range.

The partial algorithmic ADC architecture shown in Fig. 3(b) was first proposed in [21]. In this architecture, binary-weighted discrete analog inputs are assumed, which limits the utility of the architecture. They are summed with the residue of the same binary weight. The most significant partial product $P[1]$ is sampled in the first cycle and one bit $d_2[1]$ is extracted. The residue $w_2[1]$ has half the binary weight of $P[1]$ and therefore can be directly summed with the second most significant partial product $P[2]$. This relationship holds for all the remaining conversion cycles. Adding the two signals causes the comparator input signal to have a doubled range. Hence, an additional residue modulator is introduced as highlighted by a dashed outline in Fig. 3(b) to maintain the signal range, and two bits are generated in every cycle.

In the proposed architecture, the multiplication of the analog input signal I by the digital kernel coefficient C is implemented by selective sampling of the input controlled by the bits in C sequentially fed in from MSB to LSB as depicted in Fig. 3(c). This results into a sequence of binary-weighted analog signals, P , and extends the partial ADC architecture in [21] to an algorithmic MADC architecture.

Assuming MADC resolution, N , equal to the binary depth of C , K , the residue in the first cycle is

$$w_2[1] = 2(P[1] - d_1[1]) - d_2[1]. \quad (16)$$

The residue in the remaining $(N - 1)$ cycles is

$$w_2[k] = 2(P[k] + w_2[k-1] - d_1[k]) - d_2[k], \quad k = 2, 3, \dots, N. \quad (17)$$

Binary-weighted summation of (16) and (17) over k results in

$$\sum_{k=1}^N 2^{-(k+1)} (2d_1[k] + d_2[k]) = \sum_{k=1}^N 2^{-k} P[k] - 2^{-(N+1)} w_2[N] \quad (18)$$

where

$$\sum_{k=1}^N 2^{-(k+1)} (2d_1[k] + d_2[k]) = \hat{T}' \quad (19)$$

is the digital representation of T' . This operation yields multiplication of the pixel output I with the unsigned digital coefficient C , with an output resolution of N bits

$$\hat{T}' = CI + q' \quad (20)$$

where

$$|q'| = \left| 2^{-(N+1)} w_2[N] \right| \quad (21)$$

is the quantization noise.

When all of the partial products are fed into the MADC, the conversion can be continued (i.e., if $N > K$) by operating on the signal residue as in a conventional algorithmic ADC until all the required bits are extracted. In this mode the input signal P is zero, and the output of the first comparator d_1 is zero.

B. Algorithmic Signed Weighted-Averaging ADC

To compute the inner product in (6), each column-parallel MADC is further extended to include signed multiplication and cross-pixel accumulation. As illustrated in Fig. 4, the outputs of V adjacent pixels in one column are multiplied by V coefficients of one column of the programmable spatial kernel stored in a shift register. First, the pixel outputs are multiplied by the MSB bits of the respective coefficients, $c_v[1]$. This binary-analog multiplication is performed by the two-input multiplexer at the input of the accumulator. Signed multiplication is achieved by controlling the sampling order of the pixel signal and reset values by the corresponding sign bits, $sign(C_v)$. The results of the signed binary-analog multiplications for V pixels are summed by the accumulator to form the most significant partial product, $P[1]$. This procedure is repeated for all the remaining bits from MSB-1 to LSB.

Binary-weighted summation and quantization of the partial products is performed by the MADC, as explained in Section IV-A. The two output bit streams d_1 and d_2 are combined utilizing two N -bit shift registers and an N -bit accumulator as shown in Fig. 4. This yields the signed weighted-average of the pixel outputs in the digital domain with N -bit resolution

$$\hat{T} = \sum_{v=1}^V C_v I_v + q, \quad (22)$$

where q is the quantization noise.

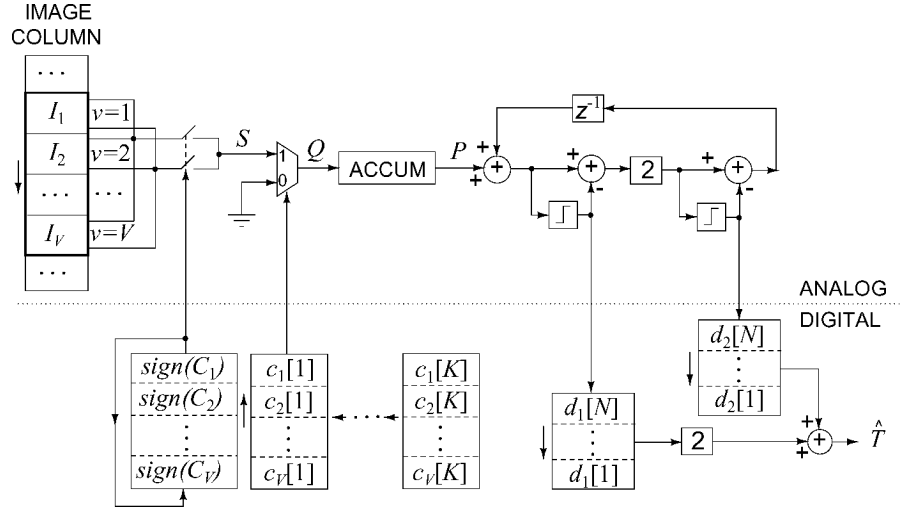


Fig. 4. Architecture of one column of the computational image sensor.

V. TWO-DIMENSIONAL COMPUTING

Section IV described an efficient implementation of (5) in the mixed-signal VLSI domain. This section presents a digital implementation of the addition in the second spatial dimension of the block-matrix transform described by (4).

A. Switch Matrix

Fig. 5 shows the block diagram of the switch matrix which routes the digital coefficients C to the respective columns. It takes the H block-matrix coefficients and their corresponding sign bits from two sets of shift registers and routes them to X/H groups of adjacent analog-multipliers and sign units, respectively. A total of V kernel coefficients each in a binary format of length K -bits are stored in the coefficient shift registers and shifted out, MSB first.

While sampling each pixel row, the corresponding sign bits are also synchronously applied to the sign units. To maintain the V -row time period, the sign and coefficients values are looped back to the input of the shift registers. Therefore, column-parallel correlation of the block C with the image I is realized sequentially in time. The switch matrix size and complexity have a linear dependence on H . The switch matrix integration area overhead scales linearly with the horizontal imager size X and becomes small for high resolution imagers.

B. Digital Accumulation

The quantized signed weighted average of pixel values in (22) can be expressed in the general form analogous to the analog formulation in (5) as

$$\hat{T}_{ij,h} = \sum_{v=1}^V C_{hv} I_{xy} + q_{ij,h}, \quad (23)$$

where $q_{ij,h}$ is the column-wise signed weighted-averaging quantization noise.

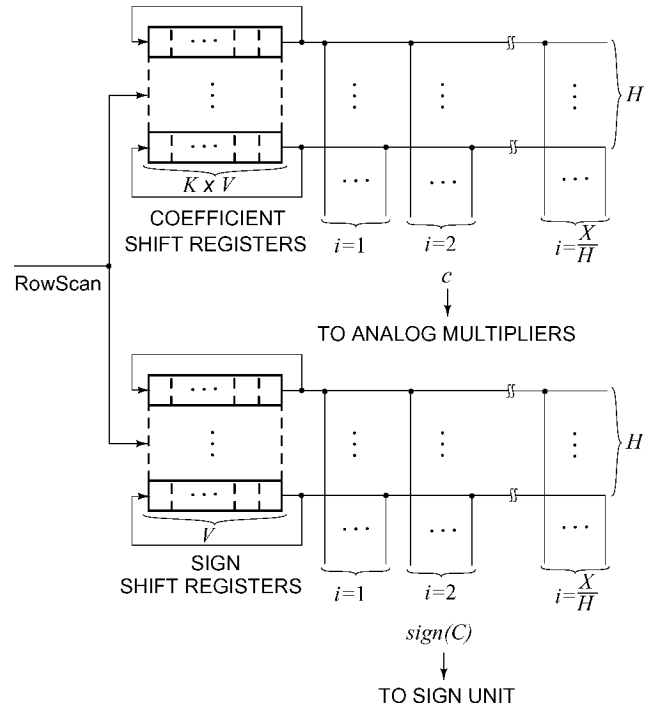


Fig. 5. Switch matrix block diagram.

Spatial accumulation over H adjacent ADC outputs is performed by a simple digital delay-adder loop during readout similar to the analog formulation in (4)

$$\hat{T}_{ij} = \sum_{h=1}^H \hat{T}_{ij,h} = \sum_{h=1}^H \sum_{v=1}^V C_{hv} I_{xy} + q_{ij}, \quad (24)$$

where q_{ij} is the transformed image quantization noise.

This two-dimensional mixed-signal VLSI implementation realizes any type of block-matrix or convolutional transform with $H, V \leq 8$.

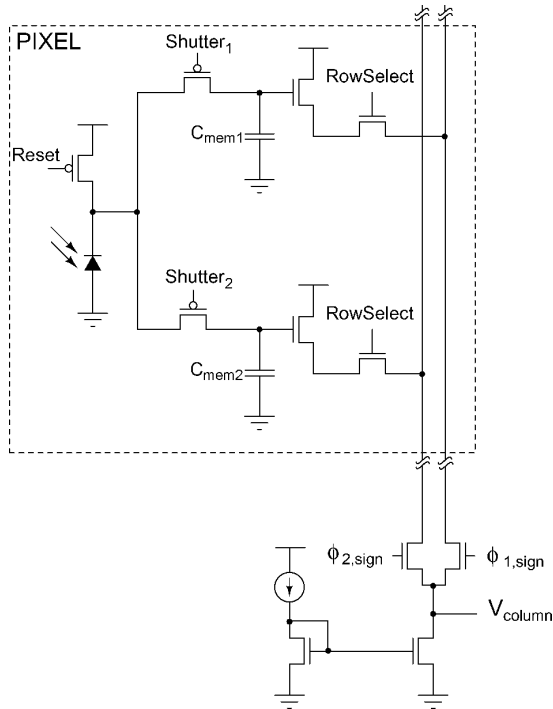


Fig. 6. Dual frame memory active pixel sensor circuit.

VI. VLSI CIRCUIT IMPLEMENTATION

A. Pixel Circuit

An array of 128×128 active pixel sensors acquires the video data. The pixel circuit is depicted in Fig. 6. Each pixel comprises a resettable n^+ -diffusion- p -substrate photodiode, and two signal paths consisting of two pMOS electronic shutter switches, two frame memories, two column-shared output source followers and readout switches [20]. Use of pMOS reset and shutter switches rather than nMOS increases the pixel voltage dynamic range by one V_{TH} of the nMOS transistor with body effect at the cost of slightly reduced fill factor. The reset and shutter switches are of the minimum size to reduce the channel charge injection and clock feedthrough errors.

The in-pixel dual frame memory enables multiple sampling of the same pixel, as required in the MADC architecture to compute spatial video transforms. This is achieved by storing the reset voltage and the integrated photocurrent on the two capacitors as illustrated in the timing diagram of Fig. 7(a), so that both are available for subsequent resampling. Non-destructive pixel readout enables not only non-overlapping (block-matrix) but also overlapping (convolutional) spatial image transforms. Two frame memories also allow for frame differencing which can be utilized in temporal video processing. The timing diagram for temporal difference mode is shown in Fig. 7(b). The in-pixel frame memories are implemented as MOS capacitors to allow for a higher density of integration inside the pixel and consequently a larger fill factor. The size of the MOS capacitors is chosen to strike a balance between achieving a higher pixel sensitivity and lowering the errors due to channel charge injection as well as junction leakage. In the experimental prototype the frame memories are approximately 15 fF.

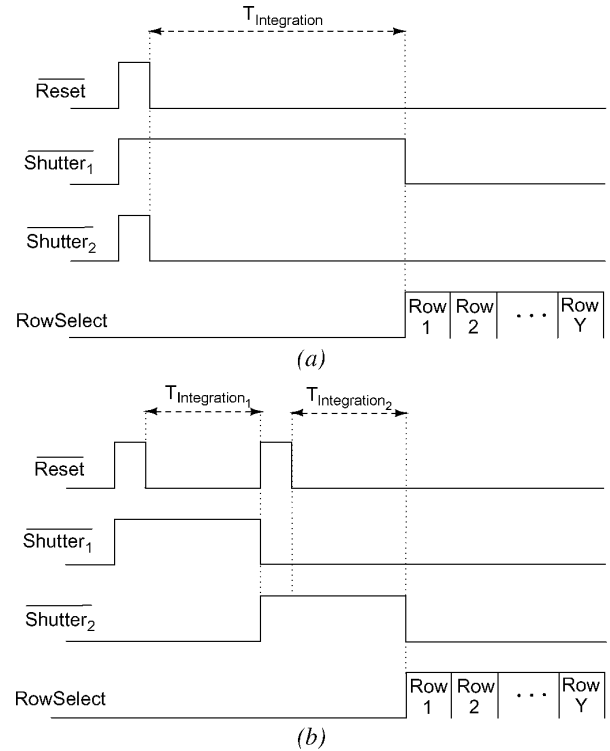


Fig. 7. Timing diagrams of (a) intraframe and (b) frame difference modes of operation.

A metal light shield covers the whole pixel area except the photodiode area in order to eliminate photo response from other regions of the pixel [22]. The pixel has a fill factor of 28 percent. FPN is reduced by taking the difference between the two pixel outputs. Through careful and symmetric pixel layout, the mismatch between the two signal paths is minimized as limited by local parameter variations. Digital signals *Reset*, *Shutter₁*, *Shutter₂*, and *RowSelect* are for pixel reset, exposure time control and readout timing control respectively, and are generated by the row control logic as described in Section III.

B. Difference-Sign-Multiplier-Accumulator Circuit

Fig. 8 depicts the circuit diagram of the column-parallel switched-capacitor gain stage following the pixel array. It combines four functionalities in a single amplifier: difference computation, sign transformation, binary-analog multiplication and accumulation.

The amplifier is a single-stage cascoded common-source nMOS amplifier. The clocks ϕ_1 and ϕ_2 are non-overlapping. The size of the input capacitor, C_s , is 125fF . Switches controlled by signals c_v , \bar{c}_v , ϕ_1 and *ResetAcc* are nMOS transistors. Transmission gates are used for switches G_1 , G_2 , G_4 and ϕ_2 to maintain the output signal swing. The amplifier bias current is $8\ \mu\text{A}$ and the simulated DC gain is approximately 77 dB.

The circuit computes the difference of the pixel signal and reset levels to suppress FPN. When the sensor is configured to perform spatial video processing, two signal paths are used. In this case, difference circuits reduce only FPN limited by the mismatch of the two signal paths. Thermal noise and source follower $1/f$ noise increase. When the sensor is utilized in the

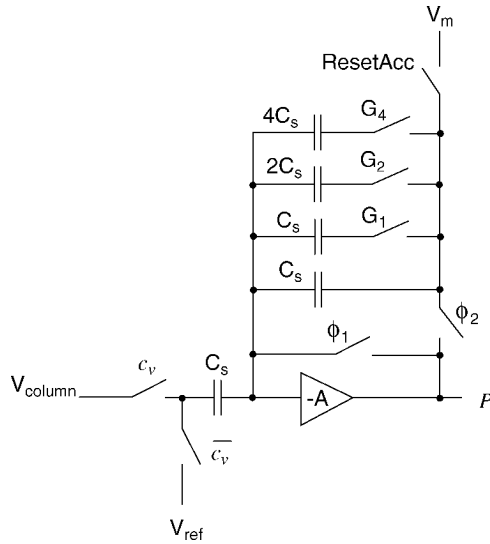


Fig. 8. Single-amplifier gain stage which performs difference computation, sign transformation, binary-analog multiplication, and accumulation over adjacent pixels in one column.

raw-image readout mode, the pixel values need to be sampled only once and one signal path can be used. In this case, difference circuits suppress FPN, 1/f noise, and reset noise.

Sign multiplication is performed by selecting the appropriate sequence order of $\phi_{1,sign}$ and $\phi_{2,sign}$ in Fig. 6 according to the corresponding sign bit $sign(C_v)$ to sample the differential pixel outputs. In the case of positive sign, $\phi_{1,sign}$ and in the case of negative sign $\phi_{2,sign}$ are slightly delayed with respect to ϕ_1 , so that charge injection errors are signal-independent and appear simply as an offset at the end of the sampling phase.

Binary-analog multiplication is performed by the two-switch multiplexer controlled by c_v and \bar{c}_v . When the bit c_v is one, the two pixel outputs are applied to the input capacitor during ϕ_1 and ϕ_2 (the pixel signal is multiplied by one). When the bit c_v is zero, the input capacitor is driven by a constant reference voltage during both phases (the pixel signal is multiplied by zero).

Accumulation over V adjacent pixels is performed on the programmable feedback capacitor bank. As voltage-domain outputs of pixels are sampled sequentially in time, the input of the accumulator is always of the full dynamic range. To maintain the output of the accumulator within the same dynamic range, the gain is programmed to be inversely proportional to the vertical size of the spatial kernel, V . In this case, the weighted average of V pixels achieves a maximum signal-to-noise ratio (SNR) improvement of $10\log V$, at the cost of lower spatial resolution. This implies that the pixel SNR can be less than the SNR of the accumulator by the same value.

C. Algorithmic Signed Weighted-Averaging ADC

Algorithmic ADC architectures require a small amount of analog circuitry as they employ the same stage circuitry to perform the quantization cyclically in time. This is also true for the modified algorithmic MADC architecture described in Section IV-A. Fig. 9 illustrates how the architecture of the MADC is optimized to reduce the number of gain stages for minimum power dissipation. The adders and multipliers are

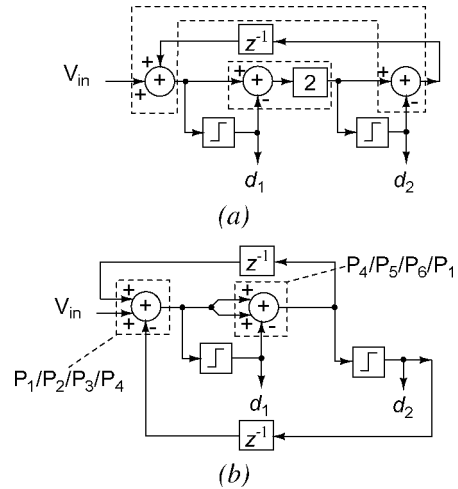


Fig. 9. (a) Amplifier merging in the multiplying ADC architecture. (b) Amplifier sharing in the multiplying ADC architecture.

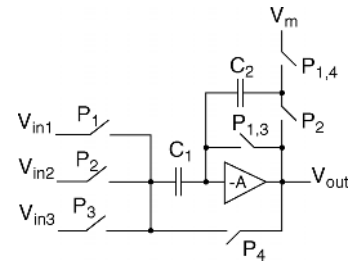


Fig. 10. Three-input adder circuit that does not rely on the capacitor ratio matching.

combined in two groups as shown with the dashed outlines in Fig. 9(a).

The active stages within each group are combined and implemented with a single-amplifier three-input switched-capacitor adder shown in Fig. 10, which is reused twice in the MADC architecture. The inverting amplifier is a cascoded common-source amplifier with a bias current of 15 μ A and a DC gain of approximately 78 dB. The size of capacitors C_1 and C_2 is chosen to be 300 fF.

To perform the addition over three inputs four clock phases are utilized. The four phases of operation for each of the ADC amplifiers are denoted in Fig. 9(b). Two of the eight phases needed for the two stages are shared which results in a total of six phases P_1 through P_6 .

Fig. 11 illustrates the four-phase clocking scheme that adds/subtracts three inputs. In phase one depicted in Fig. 11(a), V_{in1} is sampled on the input capacitor C_1 . In phase two shown in Fig. 11(b), the second input, V_{in2} , is applied to the input of the adder. Therefore, the difference of the two corresponding charges is integrated on the feedback capacitor C_2 , and $V_{in1} - V_{in2}$ appears on the output. In the third phase depicted in Fig. 11(c), V_{in3} is sampled on C_1 while the charge on C_2 is preserved. Finally, in phase four the charges on both capacitors are recombined on the input capacitor C_1 which is then connected in feedback as shown in Fig. 11(d).

A four-phase clock is commonly utilized to implement a two-input adder which is insensitive to the capacitor ratio mismatch

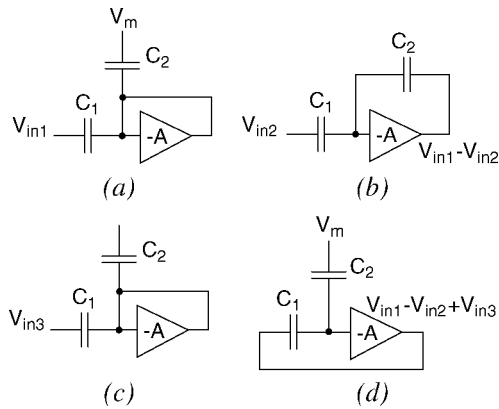


Fig. 11. The four-phase clocking scheme to add/subtract three inputs. (a) Sample the first input on C_1 while resetting C_2 . (b) Sample the second input and transfer the charge difference to C_2 . (c) Sample the third input on C_1 while preserving the charge on C_2 . (d) Dump the charge of C_2 back on C_1 and connect C_1 to the output.

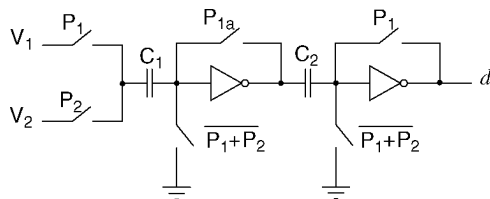


Fig. 12. The MADC comparator circuit.

[23]. Therefore, addition of three inputs has a zero power overhead over conventional two-input addition for ADCs with accuracy beyond capacitor matching where four-phase clock is utilized.

Another important block in the MADC architecture is the comparator. The two-stage comparator circuit is shown in Fig. 12. The two-stage implementation lowers the input-referred offset voltage of the comparator due to channel charge injection errors. The channel charge injection of the switch controlled by P_{1a} is stored on the coupling capacitor between the first and second stage and therefore is eliminated. The channel charge injection of the second stage is not cancelled; however when referred back to the input, it is effectively divided by the gain of the first stage [24]. The clock timing is such that the pulsewidth of P_{1a} is shorter than that of P_1 . Thus, the first stage leaves the reset (closed-loop) mode earlier than the second stage. The two switches at the inputs of the inverters ensure zero static power dissipation during the four phases of ADC operation when the comparator is not utilized.

VII. EXPERIMENTAL RESULTS

This section presents experimental results measured from a 128×128 -pixel integrated prototype fabricated in a $0.35 \mu\text{m}$ standard CMOS process. The micrograph of the $4.4 \text{ mm} \times 2.9 \text{ mm}$ computational image sensor die is shown in Fig. 13.

Fig. 14(a) shows a test image (portrait of Audrey) projected onto the imager pixel array through a lens mounted on the package. Fig. 14(b) shows the image acquired and digitized by the fabricated prototype when it was configured in the raw-image readout mode. The integration time used for image

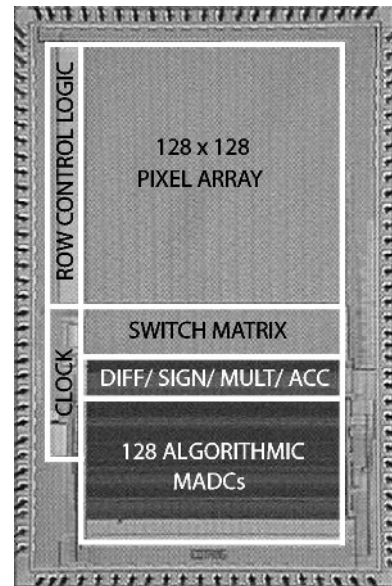


Fig. 13. Die micrograph of the focal-plane algorithmically-multiplying CMOS computational image sensor. The integrated $4.4 \text{ mm} \times 2.9 \text{ mm}$ prototype was fabricated in a $0.35 \mu\text{m}$ standard CMOS technology.

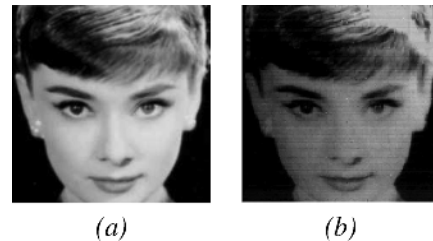


Fig. 14. (a) Test image (portrait of Audrey) projected onto the pixel array. (b) Digital output of the CMOS imager obtained with on-chip algorithmic ADC.

acquisition was 25 ms. Column-parallel FPN was removed by subtracting a dark frame from the acquired signal frame. Some temporal noise, due to interference from digital circuits on the test board, was observed and can be seen in Fig. 14(b).

The computational functionality of the image sensor was validated in on-chip spatial video compression. Focal-plane video compression yields an imager output data rate which is proportional to the mere information rate of the video, not the dimensions of the pixel array. Therefore, it relaxes the imager output bandwidth or storage capacity requirements in applications where a low-power wireless transmitter or small storage capacity are desired.

One approach to perform image compression is to filter out redundant and localized gradient values of the image according to a threshold bias. The threshold is set based on the required compression ratio and the reconstructed image quality specifications. Another compression technique is non-uniform quantization of the block-matrix transformed image, where the more significant low-frequency spatial information is quantized with a higher resolution compared to the less important high-frequency information. Discrete wavelet transform (DWT) and DCT are computationally intensive block-matrix transforms used in image and video compression algorithm standards such as JPEG, JPEG2000, H.261, and MPEG [25].

Two-dimensional one-, two-, and three-level Haar DWTs were computed on the chip. The spatial kernels used for the one-level Haar wavelet transform are

$$\Phi_1 = \frac{1}{4} \begin{pmatrix} +1 & +1 \\ +1 & +1 \end{pmatrix}, \quad (25)$$

$$\Psi_1^H = \frac{1}{4} \begin{pmatrix} +1 & +1 \\ -1 & -1 \end{pmatrix}, \quad (26)$$

$$\Psi_1^V = \frac{1}{4} \begin{pmatrix} +1 & -1 \\ +1 & -1 \end{pmatrix}, \quad (27)$$

$$\Psi_1^D = \frac{1}{4} \begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}, \quad (28)$$

where $1/4$ coefficient ensures the transformed image values stay within the image intensity range. Correlation of the above spatial kernels with the image can be expressed as the block-matrix transform of (1), where $C_{hv} \in \{-1, +1\}$ are the block-matrix coefficients. Correlation with ϕ_1 low-pass filters the acquired frame signal, while Ψ_1^H , Ψ_1^V and Ψ_1^D extract the relationship between intensities of neighboring pixels in the horizontal, vertical and diagonal directions, respectively. R -level Haar transform is obtained by using $3R + 1$ Haar wavelet kernels of size $H = V = 2^r$, with $r = 1, \dots, R$. For larger kernels where $r = 2, \dots, R$, the matrices are obtained by replacing each element of the first-level Haar matrices of (25)–(28) with square matrices of size 2^{r-1} corresponding to that element.

In order to compute the one-level Haar transforms, the size of the kernel window is set by the switch matrix to two. This requires the gain of the accumulator stage in front of the MADC to be $1/2$. Also, the accumulator has to be reset once every two rows of the pixel array have been scanned. For the experiments, the resolution of the coefficients of the kernels is set to be four bits. Therefore, analog partial products are applied to the MADCs during the first four conversion cycles. The MADC outputs within each block are combined off-chip to yield the appropriate Haar transform.

Fig. 15(a) shows the experimentally recorded outputs from the chip for the three levels of the transform. To achieve compression, the transformed image pixels are compared to a threshold value. Transformed image details which have a magnitude below the threshold are filtered out. Fig. 15(b) shows the corresponding off-chip reconstructed results. For the same threshold, a higher-level Haar transform yields more compression at the cost of reduced reconstructed image quality. The resulted compression ratios for the first, second and third-level transforms are 3.85, 14.63 and 30.12, respectively.

Fig. 16 demonstrates the experimentally measured trade-off between the peak signal-to-noise ratio (PSNR) and the compression ratio for the first-level Haar transform obtained by varying the compression threshold. The inset images correspond to the experimentally recorded images that were compressed on the chip and subsequently decompressed off-chip by computing the inverse Haar transform.

As stated in Section VI-A, the dual in-pixel frame memories enable non-destructive pixel readout, which is required by the MADC circuits to perform analog-digital multiplication. Although the focus of this architecture is on spatial video processing, the dual frame memories also allow the architecture to

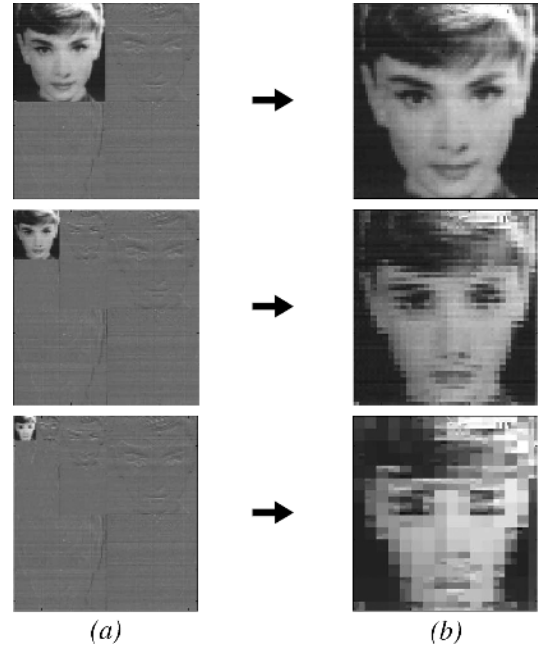


Fig. 15. (a) Experimentally recorded one-level (*top*), two-level (*center*), and three-level (*bottom*) Haar wavelet transforms computed on the CMOS computational image sensor chip. (b) Reconstructed images for one-level (*top*), two-level (*center*), and three-level (*bottom*) Haar wavelet transforms for the same compression threshold. Compression ratios from top to bottom are: 3.85, 14.63, and 30.12.

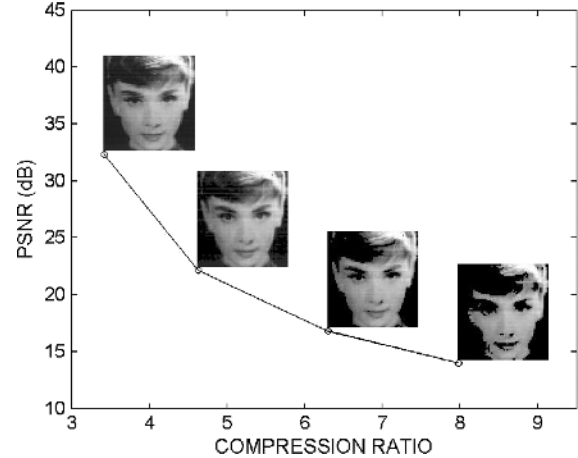


Fig. 16. Reconstructed images obtained by decompression of the experimentally computed one-level Haar discrete wavelet transform of the original image for varying compression thresholds.

perform forward frame differencing at no additional cost. This is achieved by storing two consecutive frame data on the two frame memories rather than the pixel reset and signal values. Frame differencing is a technique commonly employed in motion detection and temporal video compression algorithms [18]–[20].

To perform the frame difference experiments, an image of a rotating, radial disk was projected onto the pixel array. The integration time for each video frame in this mode was 12.5 ms. The difference circuit was biased such that only a positive difference was recorded, and the quantization was performed off-chip. Fig. 17(a) and (b) shows the still and rotating disk images

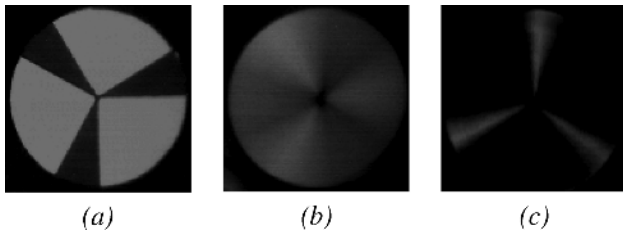


Fig. 17. (a) Still disk image in the raw-image readout mode. (b) Rotating disk image in the raw-image readout mode. (c) Rotating disk image in the frame-difference mode.

TABLE I
SUMMARY OF CHARACTERISTICS

Technology	0.35 μ m CMOS
Supply Voltage	3.3V
Die Area	4.4mm \times 2.9mm
Array Dimensions	128 \times 128 pixels
Pixel Size	15.4 μ m \times 15.4 μ m
Fill Factor	28%
Dark Current	36fA/pixel
Frame Rate	30fps
Output Resolution	8-bit
Kernel Size	2 \times 2 to 8 \times 8
Throughput @ HDTV 1080i	
Readout (30fps)	187MOPS
Maximum	20GOPS
Total Power (8 \times 8 Kernel)	26.2mW
Pixel Array	0.6mW
Accumulator and MADC	24.4mW
Digital	1.2mW

respectively acquired in the raw-image readout mode. Fig. 17(c) shows the disk image in the frame-differencing mode.

In some applications, spatial and temporal image processing can be combined, for example to reduce the imager output data rate and increase the video compression rate further.

Table I presents a summary of electrical and optical characteristics experimentally measured on the fabricated prototype. In a real-time single-scan block-matrix transform computation, the imager delivers three operations per pixel during readout. These three operations correspond to the frame difference, multiplication and addition. At HDTV 1080i imager resolution and 30 fps frame rate, the imager is projected to yield a readout computational throughput of 187MOPS. Based on a quantizer sampling rate of 52 ksp/s, the image sensor yields a maximum sustained computational throughput of 160MOPS which scales up to 20GOPS at HDTV 1080i resolution.

VIII. COMPARATIVE ANALYSIS

This section compares the presented computational imager architecture with a conventional digital imager where column-parallel algorithmic ADCs are utilized for raw-image quantization and no computation is performed. Comparison is made

for the case when the accumulator and MADC circuits perform focal-plane 8 \times 8 block-matrix transform computation.

In the conventional approach, algorithmic ADCs quantize the raw-image frames. The ADC sampling rate is directly proportional to the imager vertical resolution. For the MADC architecture performing 8 \times 8 block-matrix transform computation, the sampling rate of the MADCs is effectively reduced by a factor of 8, as there is one ADC sample per 8 pixels in any column. Assuming amplifier static power consumption is dominant in both ADCs' power, and the amplifiers in the two architectures drive equal load capacitances, this translates to a factor of 8 savings in the MADC power. Since the MADC amplifiers operate off a six-phase clock, there is a factor of 3 overhead in power compared to the conventional algorithmic ADC which utilizes a two-phase clock. Therefore, if the conventional algorithmic ADC power is denoted as P , the total MADC static power dissipation is equal to $3P/8$, divided equally between the two three-input adders.

The power consumption of the accumulator amplifier can also be estimated under the assumption of having equal capacitive loads for the accumulator and the MADC amplifiers. In the case of 8 \times 8 kernels, 8 pixels have to be sampled during each conversion cycle. This operation requires 8 clock cycles or 16 clock phases. Therefore, the accumulator amplifier has to be faster than each ADC amplifier by a factor of 16/6, which results in a power consumption of $P/2$ for the accumulator. Therefore, the total power consumption of one MADC channel in the computational imager is $7P/8$. Considering the small overhead power required to bias the in-pixel source followers during multiple pixel sampling, we can conservatively assume equal power dissipation for the two cases being compared.

As discussed in Section VII, the computational imager delivers three operations per pixel during readout, which corresponds to hundreds of millions of operations per second or higher computational throughput in high-resolution or high-frame-rate applications. In a single 8 \times 8 block-matrix transform, this throughput is delivered while dissipating approximately the same power as a conventional digital imager performing no computation. To perform the same image processing task, the conventional approach also requires a peripheral DSP. The equivalent throughput is delivered by a DSP at the cost of additional power and integration area. In the proposed approach, there is no need for such a DSP.

IX. CONCLUSION

A mixed-signal VLSI implementation of a digital CMOS imager is presented. It computes block-matrix and convolutional transforms as well as frame difference on the focal plane for real-time spatio-temporal video processing. The approach combines weighted spatial averaging and algorithmic quantization in a single analog-to-digital conversion cycle, making focal-plane computing an intrinsic part of the quantization process. In 8 \times 8 block-matrix transforms, the approach yields power dissipation almost equal to that of a conventional digital imager while the need for a peripheral DSP is eliminated. The experimental results obtained from a 0.35 μ m 128 \times 128-pixel CMOS prototype validate the utility of the design in large-scale focal-plane video processing.

REFERENCES

- [1] H. Yamashita and C. G. Sodini, "A 128×128 CMOS imager with 4×128 bit-serial column-parallel PE array," in *IEEE Int. Solid-State Circuits Conf. (ISSCC'01) Dig. Tech. Papers*, Feb. 5–7, 2001, pp. 96–436.
- [2] Y. Nishikawa, S. Kawahito, M. Furuta, and T. Tamura, "A high-speed CMOS image sensor with on-chip parallel image compression circuits," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC'07)*, pp. 833–836.
- [3] H. Yamasaki and T. Shibata, "A real-time image-feature-extraction and vector-generation VLSI employing arrayed-shift-register architecture," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 2046–2053, Sep. 2007.
- [4] B. Khailany, T. Williams, J. Lin, E. Long, M. Rygh, D. Tovey, and W. J. Dally, "A programmable 512 GOPS stream processor for signal, image, and video processing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC'07) Dig. Tech. Papers*, Feb. 11–15, 2007, pp. 272–602.
- [5] M. Nakajima, H. Noda, K. Dosaka, K. Nakata, M. Higashida, O. Yamamoto, K. Mizumoto, H. Kondo, Y. Shimazu, K. Arimoto, K. Saitoh, and T. Shimizu, "A 40GOPS 250 mW massively parallel processor based on matrix architecture," in *IEEE Int. Solid-State Circuits Conf. (ISSCC'06) Dig. Tech. Papers*, Feb. 5–9, 2006, pp. 410–662.
- [6] W. Hinrichs, J. P. Wittenburg, H. Lieske, H. Kloos, M. Ohmacht, and P. Pirsch, "A 1.3-GOPS parallel DSP for high-performance image-processing applications," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 946–952, Jul. 2000.
- [7] S. E. Kemeny, R. Panicacci, B. Pain, L. Matthies, and E. R. Fossum, "Multiresolution image sensor," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 4, pp. 575–583, Aug. 1997.
- [8] Q. Luo and J. G. Harris, "A novel integration of on-sensor wavelet compression for a CMOS imager," in *IEEE Int. Symp. on Circuits and Systems (ISCAS'02)*, Scottsdale, AZ, May 26–29, 2002.
- [9] S. Kawahito, M. Yoshida, M. Sasaki, K. Umehara, D. Miyazaki, Y. Tadokoro, K. Murata, S. Doushou, and A. Matsuzawa, "A CMOS image sensor with analog two-dimensional DCT-based compression circuits for one-chip cameras," *IEEE J. Solid-State Circuits*, vol. 32, no. 12, pp. 2030–2041, Dec. 1997.
- [10] V. Gruev and R. Etienne-Cummings, "Implementation of steerable spatiotemporal image filters on the focal plane," *IEEE T. Circuits and Systems II*, vol. 49, no. 4, pp. 233–244, Apr. 2002.
- [11] A. Bandyopadhyay, J. Lee, R. W. Robucci, and P. Hasler, "MATIA: A programmable $80 \mu\text{W}/\text{frame}$ CMOS block matrix transform imager architecture," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 663–672, Mar. 2006.
- [12] C. C. Cheng, C. H. Lin, C. T. Li, S. Chang, C. J. Hsu, and L. G. Chen, "iVisual: An intelligent visual sensor SoC with 2790 fps CMOS image sensor and 205GOPS/W vision processor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC'08) Dig. Tech. Papers*, Feb. 3–7, 2008, pp. 306–615.
- [13] P. Dudek and P. J. Hicks, "A general purpose processor-per-pixel analog SIMD vision chip," *IEEE T. Circuits and Systems I*, vol. 52, no. 1, pp. 13–20, Jan. 2005.
- [14] A. Rodriguez-Vazquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, and S. E. Meana, "ACE16k: The third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *IEEE T. Circuits and Systems I*, vol. 51, no. 4, pp. 851–863, May 2004.
- [15] R. C. Galan, F. Jimenez-Garrido, R. Dominguez-Castro, S. Espejo, T. Roska, C. Rekeczky, I. Petras, and A. Rodriguez-Vazquez, "A bio-inspired two-layer mixed-signal flexible programmable chip for early vision," *IEEE T. Neural Networks*, vol. 14, no. 9, pp. 1313–1336, Sep. 2003.
- [16] A. Olyaei and R. Genov, "Focal-plane spatially oversampling CMOS image compression sensor," *IEEE T. Circuits and Systems I*, vol. 49, no. 1, pp. 26–34, Jan. 2007.
- [17] A. Graupner, J. Schreiter, S. Getzlaff, and R. Schuffny, "CMOS image sensor with mixed-signal processor array," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 948–957, Jun. 2003.
- [18] V. Gruev and R. Etienne-Cummings, "A pipelined temporal difference imager," *IEEE J. Solid-State Circuits*, vol. 39, no. 3, pp. 538–543, Mar. 2004.
- [19] Y. M. Chi, U. Mallik, M. A. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, "CMOS Camera with in-pixel temporal change detection and ADC," *IEEE J. Solid-State Circuits*, vol. 42, no. 10, pp. 2187–2196, Oct. 2007.
- [20] S. Y. Ma and L. G. Chen, "A single-chip CMOS APS camera with direct frame difference output," *IEEE J. Solid-State Circuits*, vol. 34, no. 10, pp. 1415–1418, Oct. 1999.
- [21] R. Genov and G. Cauwenberghs, "Algorithmic partial analog-to-digital conversion in mixed-signal array processors," in *IEEE Int. Symp. on Circuits and Systems (ISCAS'03)*, Bangkok, Thailand, May 25–28, 2003.
- [22] S. K. Mendis, S. E. Kemeny, R. C. Gee, B. Pain, C. O. Staller, Q. Kim, and E. R. Fossum, "CMOS active pixel image sensor for highly integrated imaging systems," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 187–197, Feb. 1997.
- [23] P. W. Li, M. J. Chin, P. R. Gray, and R. Castello, "A ratio-independent algorithmic analog-to-digital conversion technique," *IEEE J. Solid-State Circuits*, vol. 19, no. 6, pp. 828–836, Dec. 1984.
- [24] R. Poujois and J. Borel, "A low-drift fully integrated MOSFET operational amplifier," *IEEE J. Solid-State Circuits*, vol. 13, no. 4, pp. 499–503, Aug. 1978.
- [25] H. Yamauchi, S. Okada, K. Taketa, T. Ohyama, Y. Matsuda, T. Mori, T. Watanabe, Y. Matsuo, Y. Yamada, T. Ichikawa, and Y. Matsushita, "Image processor capable of block-noise-free JPEG2000 compression with 30 frames/s for digital camera applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC'03) Dig. Tech. Papers*, Feb. 9–13, 2003, pp. 46–477.



Alireza Nilchi (S'01) received the B.Sc. degree (with honors) in Electrical and Electronics Engineering from the University of Tehran, Tehran, Iran, in 2005, and the M.A.Sc. degree in Electrical Engineering from the University of Toronto, Toronto, ON, Canada, in 2008. He is currently pursuing the Ph.D. degree at the University of Toronto.

He has held internship positions at the Universität der Bundeswehr München, Munich, Germany, in 2004, and Iran Telecommunication Research Center, Tehran, Iran, in 2003. His current research interests are on scaled CMOS analog/mixed-signal integrated circuits and very low power data converters for sensor applications.



Joseph Aziz (S'05) received the B.Sc. degree in Electronics and Electrical Communications Engineering from Cairo University, Giza, Egypt, in 2003, and the M.A.Sc. degree in Electrical Engineering from the University of Toronto, Toronto, ON, Canada, in 2007.

He is currently with Broadcom Corporation, Irvine, CA. His current research interests focus on the design of mixed-signal integrated circuits.



Roman Genov (S'96-M'02) received the B.S. degree (first rank) in Electrical Engineering from Rochester Institute of Technology, NY in 1996, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Johns Hopkins University, Baltimore, MD in 1998 and 2002 respectively.

Dr. Genov held engineering positions at Atmel Corporation, Columbia, MD in 1995 and Xerox Corporation, Rochester, NY in 1996. He was a visiting researcher in the Laboratory of Intelligent Systems at Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland in 1998 and in the Center for Biological and Computational Learning at Massachusetts Institute of Technology, Cambridge, MA in 1999. He is presently an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Toronto, Canada. His research interests include analog and digital VLSI circuits, systems and algorithms for energy-efficient signal processing with applications to electrical, chemical and photonic sensory information acquisition, biosensor arrays, brain-silicon interfaces, parallel signal processing, adaptive computing for pattern recognition, and implantable and wearable biomedical electronics.

Dr. Genov received Canadian Institutes of Health Research (CIHR) Next Generation Award in 2005, and DALSA Corporation Componentware/CAD Award in 2006. He is an Associate Editor of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS and IEEE SIGNAL PROCESSING LETTERS.