A General-Purpose Processor-per-Pixel Analog SIMD Vision Chip

Piotr Dudek, *Member, IEEE,* and Peter J. Hicks, *Member, IEEE*

***Abstract*—A smart-sensor VLSI circuit suitable for focal-plane low-level image processing applications is presented. The architecture of the device is based on a fine-grain software-programmable SIMD processor array. Processing elements, integrated within each pixel of the imager, are implemented utilising a switched-current analog microprocessor concept. This allows the achievement of real-time image processing speeds with high efficiency in terms of silicon area and power dissipation. The prototype 21 × 21 vision chip is fabricated in a 0.6 μm CMOS technology and achieves a cell size of 98.6 μm × 98.6 μm. It executes over 1.1 giga instructions per second (GIPS) while dissipating under 40 mW of power. The architecture, circuit design and experimental results are presented in this paper.**

***Index Terms*—Analog processor array, CMOS imager, smart sensor, vision chip.**

I. Introduction

MANY computer vision applications can benefit from the integration of image sensing and image processing functions onto a single solid-state circuit, to form a so-called "vision chip" [1]. A processor-per-pixel arrangement, as shown in Fig. 1, is of particular interest. Low-level image processing tasks (such as filtering, edge detection, feature extraction, etc.), while being computationally intensive, are inherently pixel-parallel in nature (identical, localized operations are performed on every pixel). Pixel-parallel processing architectures can thus enable real-time processing speeds, required in many applications, to be achieved. At the same time, the processor-per-pixel integration ensures that data is processed adjacent to the pixel from which it originated, so that no extra resources are wasted on long-distance data transfers. This eliminates the I/O bottleneck between the sensor and the processor and reduces the power dissipation, size, and cost of the system.

Systems employing vision chips are somewhat similar to mammalian visual systems, where preliminary image processing is performed directly on the retina, before the preprocessed information is passed on to the higher levels of the visual cortex in the brain. Yet, while the information processing in the retina is based on neural circuitry performing various operations in continuous-time, it is difficult to replicate this behavior in silicon. Although it is relatively easy to implement simple operations (e.g., convolutions with 3 × 3 kernels, some nonlinear filters, motion detection, etc.) using continuous-time circuitry [2]–[4], nevertheless it is difficult to implement a large number of different operations in the limited silicon area available to accommodate the processing circuitry in a processor-per-pixel array. The majority of computer vision algorithms, however, require a number of different low-level image processing operations to be executed on a single image. Furthermore, from the system design perspective, it is beneficial to ensure complete programmability of the system, so that various algorithms can be implemented using the same hardware.

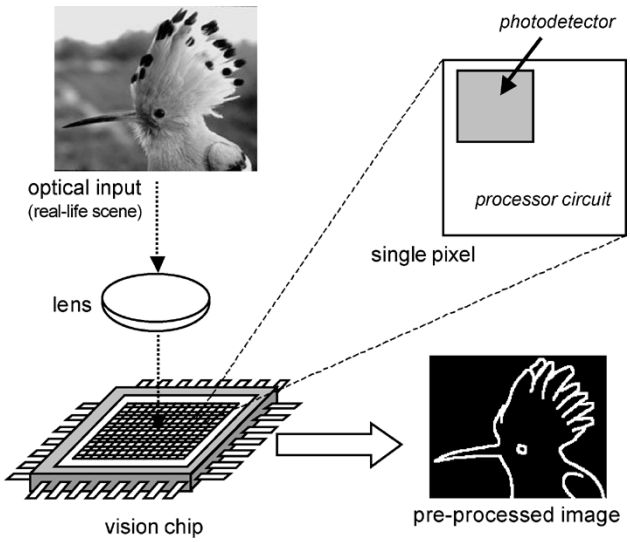


Fig. 1. A vision chip with a processor-per-pixel array.

Both of the aforementioned goals can be achieved by departing from trying to build "artificial retinas" which operate in a biologically inspired continuous-time way, and building instead a system based on the single instruction multiple data (SIMD) computer paradigm. In the SIMD computer, a single controller issues instructions that are executed by an array of processing elements (PEs). Each PE is an algorithmically programmable entity, operating in a discrete-time fashion. As compared with application-specific hardware implementations, the execution time may be increased as a result of the software implementation, but the complexity of a processor is reduced, since the same hardware resources can be reused for various purposes during the execution of the algorithm. Furthermore, the software-based system is general purpose, capable of executing an infinite number of algorithms (limited in practice by hardware resources such as the amount of available memory, etc.).

Several implementations of SIMD vision chips have been described in the literature, illustrating various approaches to

Manuscript received August 26, 2003; revised August 10, 2004. This work was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant GR/R52688/01. This paper was recommended by Associate Editor G. Cauwenberghs.

The authors are with the Department of Electrical Engineering and Electronics, University of Manchester Institute of Science and Technology (UMIST), Manchester M60 1QD, U.K. (e-mail: p.dudek@umist.ac.uk).

Digital Object Identifier 10.1109/TCSI.2004.840093

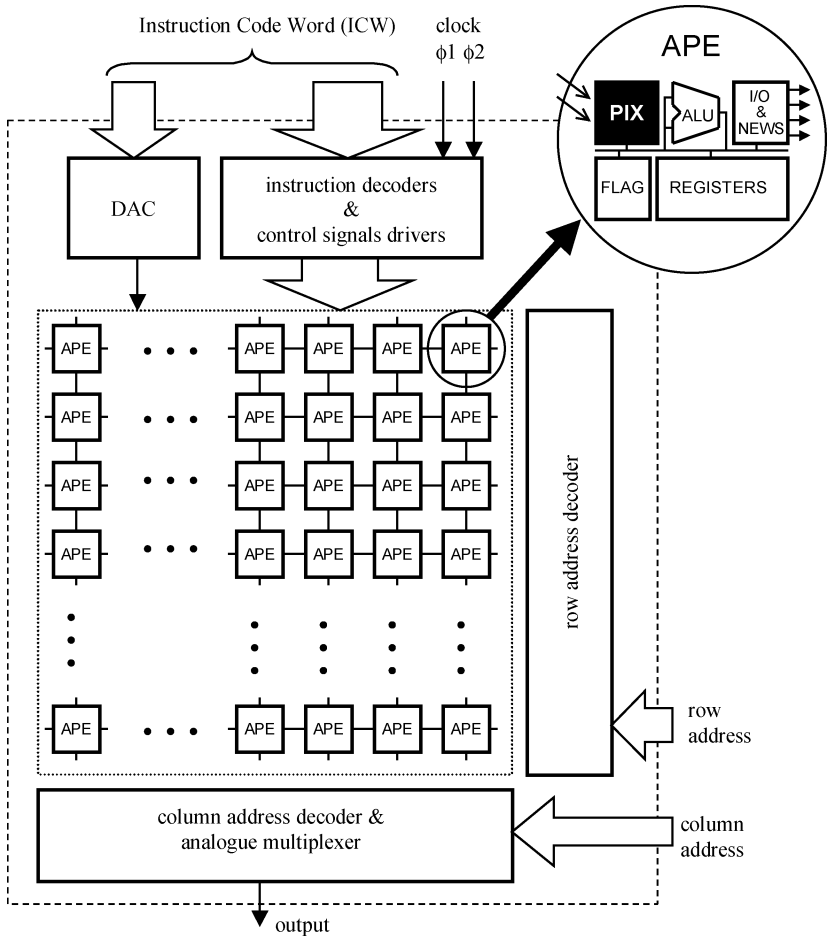


Fig. 2. SCAMP chip architecture.

the problem of designing compact, efficient, software-programmable PEs. Some of the chips use single-bit processors [5], [6], however, due to their limited capabilities (a few bits of memory per processor) they can be hardly considered "general purpose." Vision chips with more complex digital processors have also been developed [7], but these processors occupy a relatively large amount of silicon area, have limited memory and require many clock cycles to perform basic operations due to their bit-serial architecture.

It has to be noted that a PE is an embodiment of a Universal Turing Machine, and whereas this paradigm has been the foundation of digital computing, it can also be used to implement analog, sampled-data systems. We have proposed a switched-current "analog microprocessor" [8] and demonstrated that it can outperform its digital equivalent, not only in terms of cell area, but also performance and power dissipation. The concepts of sampled-data analog SIMD processing have been recently applied to the design of application-specific processor arrays [9] and vision chips with linear processor arrays [10]. Vision chips have also been reported, that combine the analog SIMD approach with the cellular neural network (CNN) mode of operation [11].

In this paper we present our approach to the design of a massively parallel, general purpose SIMD vision chip [12]–[14], which is characterized by small cell area, low power dissipation and the ability to execute a variety of image processing algorithms in real-time. In Section II, the architecture of the chip is outlined. In Section III, the circuit implementation is described, together with some measurement results. In Section IV, experiments with implementing image processing algorithms are presented. In Section V, the performance of the chip is discussed and compared with other approaches and finally some conclusions are presented in Section VI.

II. SCAMP Architecture

The architecture of our vision chip, named SCAMP (SIMD current-mode analog matrix processor), is shown in Fig. 2. The processing core is a mesh-connected array of processors, which are called analog processing elements (APEs). This name reflects the fact that the data is represented and manipulated inside the APEs using analog samples, however, the architecture and operation of the APE is similar to that of a digital microprocessor. Each APE comprises six general purpose "analog registers" ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{H}$, and $\mathbf{K}$), some special purpose registers ($\mathbf{NEWS}$, $\mathbf{M}$, $\mathbf{PIX}$, $\mathbf{FLAG}$) and I/O blocks, all connected to an internal analog data bus. The APE supports an instruction set comprising register-transfer operations, arithmetic operations, I/O operations, neighbor communication and conditional instructions. In a way akin to an 8-bit digital microprocessor, which operates on 8-bit data values, this "analog microprocessor" operates on analog data samples.

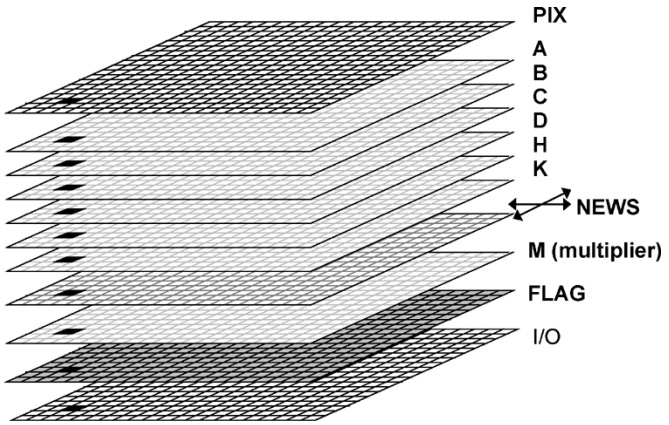


Fig. 3. Register array planes in the SCAMP architecture (single APE is marked).

The APEs in the array execute identical instructions on their local data in an SIMD fashion. As the processor array size corresponds to the image size, i.e., instructions are performed on an entire array at once, it is convenient to represent the array architecture as shown in Fig. 3, consisting of several register array planes (a single APE is thus formed from a vertical set of nodes corresponding to a unique row and column address in each of the planes). Each register array can hold a grey-level image or another array variable. Transfer instructions (for example $\mathbf{A} \leftarrow \mathbf{B}$) represent the transfer of an image from one array to the other. Similarly, arithmetic operations (e.g., $\mathbf{A} \leftarrow \mathbf{B} + \mathbf{C}$) perform pixel-wise arithmetic operations on the data arrays. The SCAMP chip supports inversion and summation of any number of arguments in a single instruction, executed in a single clock cycle. Multiplication (scaling) is performed via a special purpose multiplier register $\mathbf{M}$.

Communication between four nearest neighbors is facilitated via a special purpose $\mathbf{NEWS}$ (North, East, West, South) register. The array also supports random-access input and output. Additionally, entire rows, columns or indeed the entire array can be addressed for readout, resulting in a global summation operation. Image acquisition is supported via a special purpose register array $\mathbf{PIX}$. The value held in this register array corresponds to the state of the photodetector array, which works in an integration mode. Nondestructive readout ensures that multiple exposure times are possible, which can be used to extend the dynamic range of the image sensor.

As in the majority of SIMD processor arrays, local autonomy is supported by the "activity-flag" register ($\mathbf{FLAG}$). This register can be set or reset depending on the result of a comparison operation. If the $\mathbf{FLAG}$ register is reset the PE does not respond to the broadcast instructions, and in this way conditional operations can be performed.

III. VLSI Implementation

A prototype SCAMP chip was fabricated in a standard three-metal single-poly 0.6-μm CMOS technology. The 10 mm^2 chip comprises a 21 $\times$ 21 array of APEs, random-access I/O logic, on-chip digital to analog converter (DAC), as well as instruction decoder and control signal drivers. An external controller/sequencer is required to store and execute the programs that provide a sequence of instructions to the SCAMP chip. These instructions are decoded and the control signals are distributed to the APEs using separate drivers for each row and column of the array, which makes it easier to scale-up the design to a larger array size. The chip microphotograph is shown in Fig. 4. Each APE contains 128 transistors and occupies a silicon area of 98.6 μm $\times$ 98.6 μm.

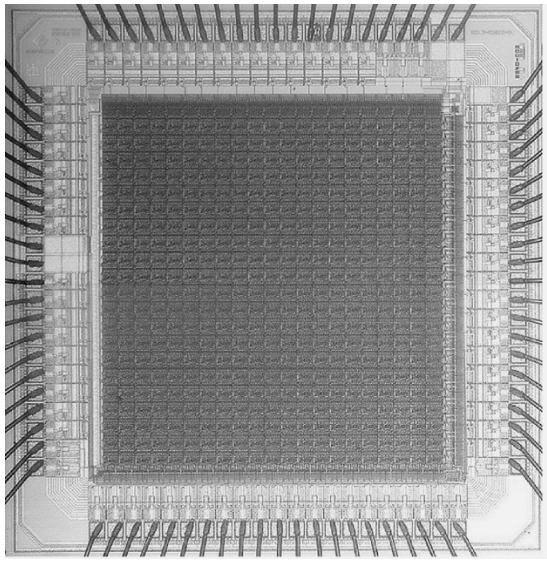


Fig. 4. Chip microphotograph.

A. Analog Processing Element

A major design consideration when designing a processor-per-pixel array is to minimize the silicon area occupied by a single processor. Another important design goal is low power dissipation. At the same time, acceptable levels of accuracy and speed of operation have to be maintained. We have achieved these goals using APEs, based upon our switched- current (SI) "analog microprocessor" concept [8]. A simplified schematic diagram of the APE is presented in Fig. 5. The APE is a discrete-time system in which data is represented as current samples flowing in and out of the analog bus (both positive and negative signals are possible).

General purpose registers are implemented as SI memory cells. Registers and other functional blocks are connected to the single wire analog bus by means of analog switches. Switches are also used to control other functions of the circuit. Register transfer and arithmetic operations are executed by closing appropriate switches, so that current samples are accordingly transferred from one register to another. For example, to execute the instruction denoted as $\mathbf{A} \leftarrow \mathbf{B} + \mathbf{K}$ we close switches $\mathrm{W_A}$, $\mathrm{S_A}$, $\mathrm{S_B}$ and $\mathrm{S_K}$, and following the operation of the SI cells we obtain $i_A = -(i_B + i_K)$, i.e., at the end of the operation the value stored in the register $\mathbf{A}$ is the *inverted* sum of the values stored in the registers $\mathbf{B}$ and $\mathbf{K}$ (note that the notation used here for the assignment operation is somewhat different than the conventional one, since it also implies the sign inversion).

It is worth noting that the arithmetic operations of addition and subtraction are performed with no need for explicit ALU

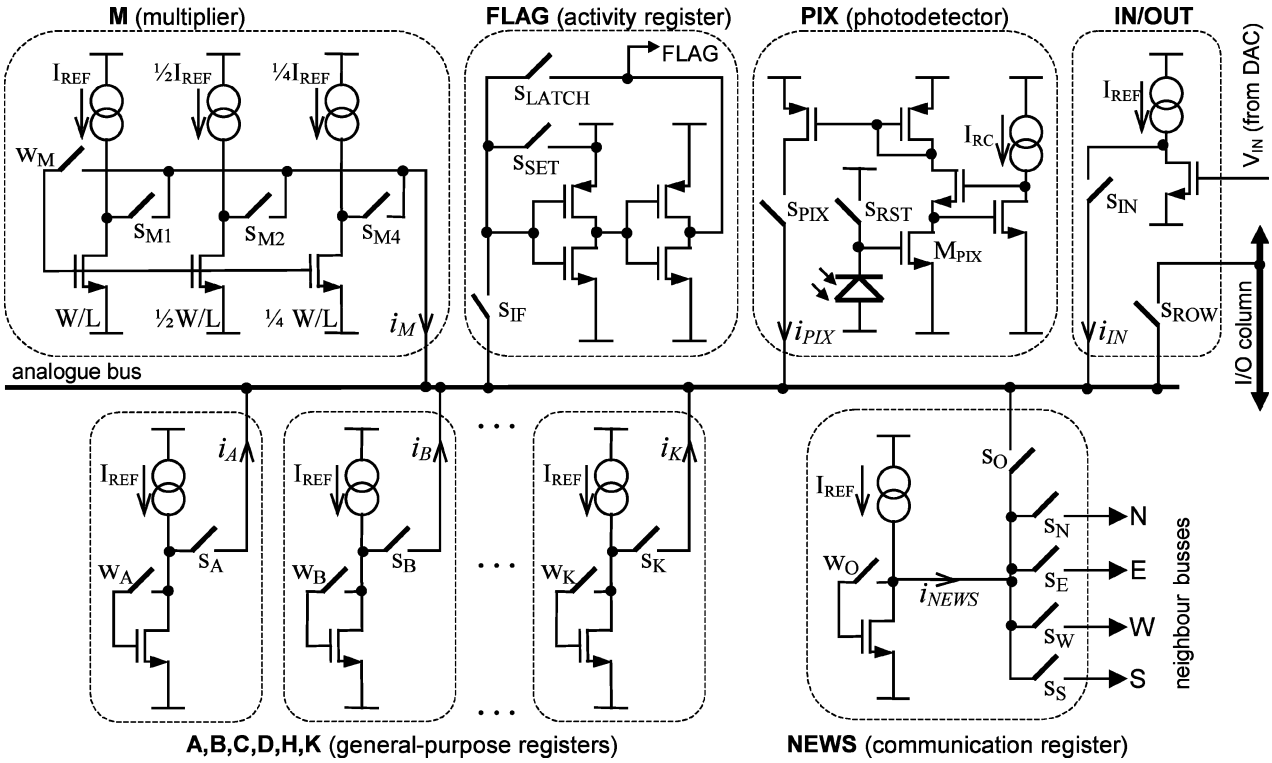


Fig. 5. Simplified schematic diagram of the analog processing element (APE).

circuitry (inversion is inherent in the SI memory cell and addition is performed directly on the analog bus using current summation). Multiplication by a digital constant is performed using a special purpose multiplier register **M**, which has binary scaled current-mirror outputs. A more detailed account of the basic operation of the switched-current analog processing element can be found in [8].

B. Register Cell

For clarity, basic SI cells are shown in Fig. 5, although in practice more complex cells are used in order to reduce processing errors (originating from charge injection and output conductance effects) and to reduce power dissipation. A full schematic diagram of a register cell is shown in Fig. 6. The "store" operation is performed in two phases ($\phi 1$ and $\phi 2$), according to the S^2I scheme [15]. The transistor M_{MEM} is used to store the initial value of the current during $\phi 1$. The transistor M_{REF} acts as a current source during $\phi 1$, and stores the error during $\phi 2$. Consequently the W-switches are implemented by transistors M_{W1} and M_{W1R} (closed during $\phi 1$) and M_{W2} (closed during $\phi 2$). The S-switch is implemented by transistor M_{SB}, while transistor M_{SP} is an additional switch, introduced to make sure that no dc current flows through the register when this register does not take part in the current instruction, thus reducing the power dissipation of the system. The register also contains logic, implemented by transistors M_{F1}, M_{F1R}, and M_{F2}, which is used to prevent the closing of the W-switches when the **FLAG** signal is not active. This conditionally disables the storage operation, which is required to implement the local autonomy feature.

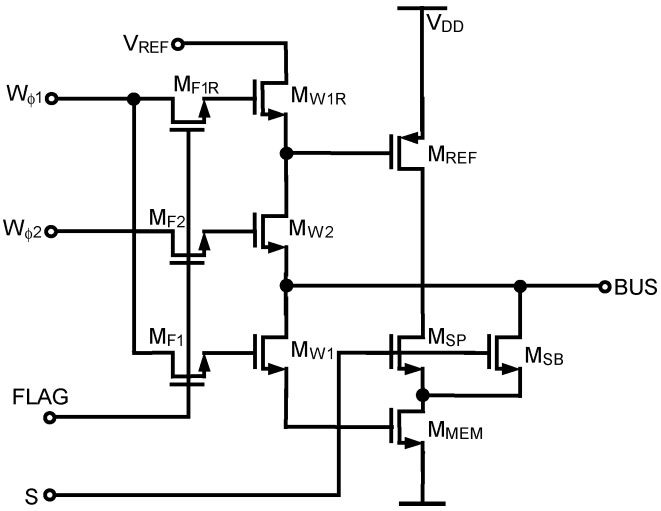


Fig. 6. Detailed schematic diagram of the register cell.

C. Accuracy

It has to be noted that unlike digital processors, where the accuracy of operations is limited by the chosen word length, analog processors have their accuracy limited by errors and noise inherent in the analog circuitry. The cells were carefully laid out to minimize parasitic capacitances and reduce the errors to a level that would be acceptable for low-level image processing algorithms. A measured systematic error associated with the storage operation in the SI register of the APE, as a function of the signal (i.e., stored data value), is plotted in Fig. 7. While the signal-independent error can be cancelled out algorithmically [8], the signal-dependent error will limit the accuracy. The magnitude of the signal-dependent error of a register transfer operation in the APE was measured to be equal to approximately 40 nA, that is 0.5% of the maximum signal level of 8 μA. In addition to the systematic error, each

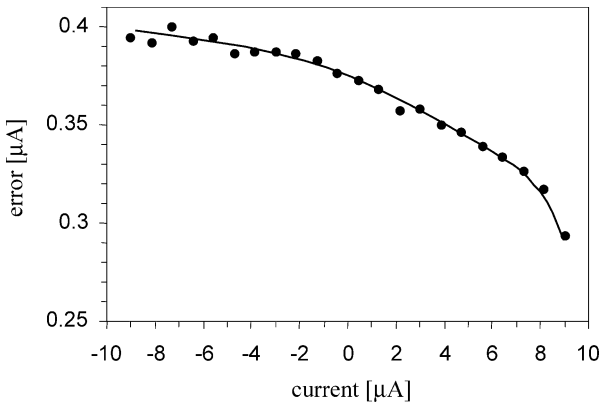


Fig. 7. Measured error characteristic of the register cell.

transfer also contributes a random error (noise) of 8.5nA rms, i.e., 0.11% of the maximum signal level.

D. Activity Flag

The flag register (see Fig. 5) is implemented as a D-latch. It can be set globally by instruction **ENDIF** (which closes switch $\mathrm{S_{SET}}$) or conditionally by a comparison instruction **IF** $\boldsymbol{X}$, where $\boldsymbol{X} \in \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathrm{etc}\ldots\}$. During the comparison instruction the current from a selected register $\boldsymbol{X}$ is routed to the analog bus, which is kept in the high-impedance state and connected to the input node of the flag register ($\mathrm{S_{IF}}$ and S_X closed). This node is consequently charged toward $\mathrm{V_{DD}}$ or discharged toward ground, depending on the sign of the current from the selected register (or a sum of currents if more registers are selected at once). Consequently, the sign of the current determines the comparison result and thus the logic level of the **FLAG** signal, which is latched by closing $\mathrm{S_{LATCH}}$.

E. Pixel

The photodetector (**PIX** circuit in Fig. 5) works in an integration mode. The voltage on the gate capacitance of $\mathrm{M_{PIX}}$ is reset by closing switch $\mathrm{S_{RST}}$ (instruction **RPIX**). Then $\mathrm{S_{RST}}$ is opened and the capacitance is discharged through the photodiode at a rate proportional to the incident light intensity. A regulated cascode output stage and a current mirror provide biasing of $\mathrm{M_{PIX}}$ in the ohmic region. As a result we obtain close-to-linear characteristic of the current i_{PIX} versus incident light intensity. After a specific integration time the current i_{PIX} can be readout to the analog bus (by closing $\mathrm{S_{PIX}}$) and sampled in one of the registers.

To reduce the fixed pattern noise (FPN), a correlated double sampling (CDS) technique can be implemented in software, by subtracting the reset level from the integration result. Having complete processors at each pixel it is relatively easily done using a following simple subroutine at the beginning of each video frame:

$\mathbf{A} \leftarrow \mathbf{PIX}$	sample integration result into A (also inverts!);
RPIX	reset photodetector;
$\mathbf{B} \leftarrow \mathbf{A} + \mathbf{PIX}$	calculate difference and store the output image in B.

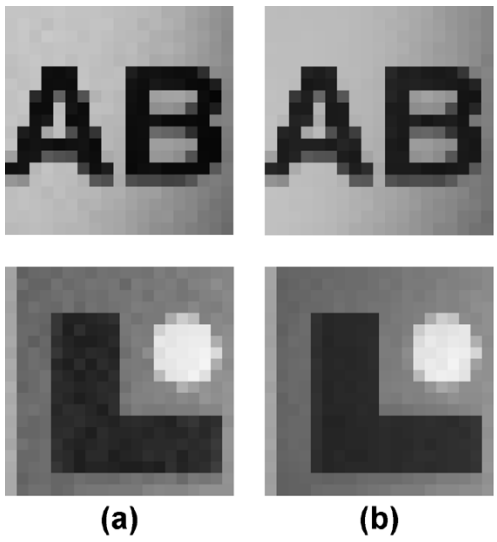


Fig. 8. Images acquired by the SCAMP chip (a) without FPN reduction (b) with FPN reduction.

The photodiode area is equal to 820 μm^2, which yields a fill factor of 8.4%, however the photodetector sensitivity is somewhat reduced by metal wires that pass over the photodiode area. With 1000 lux illumination level full-contrast images are obtained at 25 frames/s. Images obtained by the chip with and without the FPN reduction technique are shown in Fig. 8. The measured fixed pattern noise of the imager, with FPN reduction, is equal to 1% rms.

F. Neigbor Communication

A special purpose **NEWS** register is used to facilitate communication between the adjacent APEs in the array. The **NEWS** register can connect to the analog buses of four nearest neighbors, thus the current samples can be transferred from one APE to another via this register. For example, to load the register **B** of each APE with the value of the register **A** of its south neighbor the following instructions are performed:

$\mathbf{NEWS} \leftarrow \mathbf{A}$	close switches $\mathrm{W_O}$, $\mathrm{S_O}$ and $\mathrm{S_A}$;
$\mathbf{B} \leftarrow \mathbf{SOUTH}$	close switches $\mathrm{W_B}$, $\mathrm{S_B}$ and $\mathrm{S_N}$.

The layout of the **NEWS** register has been closely matched to the layout of the other registers, to ensure good signal-independent error cancellation of the nearest neighbor communication operations.

G. I/O Operations

To support random access analog I/O, the analog bus of an APE is connected to the array column bus via an access switch $\mathrm{S_{ROW}}$ controlled by a row-select signal. One column of the array is selected using an analog column-select multiplexer. A current from any register on the addressed APE can be thus readout off-chip. Additionally, column parallel analog outputs are available. Selecting multiple rows and/or columns is also allowed. It results in the summation of output currents from selected APEs, which provides a very useful operation of row-wise, column-wise, and global summation. This feature is very useful for monitoring the state of the entire array and also greatly

Code	Comments	Closed switches
	;IMAGE ACQUISITION	
H <- PIX	;sample image to H	W_H, S_H, S_{PIX}
RPIX	;reset photodetector	S_{RST}
A <- H + PIX	;subtract reset level from pixel level (CDS)	W_A, S_A, S_H, S_{PIX}
	;SMOOTH	
M <- A	;load multiplier register-array M with input image a_{xy}	W_M, S_{M1}, S_A
C <- M(0.25)	;load C (temporary variable) with $c_{xy}=¼a_{xy}$	W_C, S_C, S_{M4}
NEWS <- C	;this loads NEWS register-array with $¼a_{xy}$	W_O, S_O, S_C
B <- M(0.25) + EAST	;here we have $b_{xy}=¼a_{xy}+¼a_{(x+1)y}$	W_B, S_B, S_{M4}, S_W
C <- B		W_C, S_C, S_B
B <- M(0.25) + C + WEST	;and here we have $b_{xy} = ¼(2a_{xy}+a_{(x+1)y}+a_{(x-1)y})$	W_B, S_B, S_{M4}, S_C, S_E
M <- B	;so now M is loaded with $m_{xy} = ¼(2a_{xy}+a_{(x+1)y}+a_{(x-1)y})$	W_M, S_{M1}, S_B
C <- M(0.25)		W_C, S_C, S_{M4}
NEWS <- C	;and NEWS is now loaded with $¼m_{xy}$	W_O, S_O, S_C
B <- M(0.25) + NORTH	;here $b_{xy}=¼m_{xy}+¼m_{x(y-1)}$	W_B, S_B, S_{M4}, S_S
C <- B		W_C, S_C, S_B
B <- M(0.25) + C + SOUTH	;and finally, $b_{xy}=¼(2m_{xy}+m_{x(y+1)}+m_{x(y-1)})$	W_B, S_B, S_{M4}, S_C, S_E
	;THRESHOLD	
D <- B + IN(25)	;image is offset by 25 (threshold level)	W_D, S_D, S_B, S_{IN}; V_{IN}=v(25)
A <- D + IN(0)	;this reduces offset error of IN(x) by subtracting "zero"	W_A, S_A, S_D, S_{IN}; V_{IN}=v(0)
C <- IN(-100)	;load C with (100) – represents logic '0'	W_C, S_C, S_{IN}; V_{IN}=v(-100)
IF A	;for pixels above the threshold:	S_{IF}, S_A; open S_{LATCH}
C <- IN (100)	; load C with -100 (represents logic '1')	W_C, S_C, S_{IN}; V_{IN}=v(100)
ENDIF		S_{SET}
OUT C	;OUTPUT RESULT	S_C

Fig. 9. An example image processing algorithm performing image acquisition, smoothing and thresholding.

simplifies the design of global algorithms, such as histogramming.

To input a value in parallel to all the APEs (for example in order to generate an immediate argument for an instruction such as $\mathbf{A} \leftarrow \mathbf{25}$) a voltage $V_{\rm IN}$ is distributed globally and converted in each APE to a current $i_{\rm IN}$. The voltage $V_{\rm IN}$ is obtained from a DAC, common to all the APEs, so that the current $i_{\rm IN}$ can be set digitally with 7-bit resolution.

Digital output, random access digital input and analog input are also possible via a combination of the random access feature, immediate argument generation and conditional instructions [14].

IV. Image Processing Examples

The software-programmable architecture of the SCAMP chip allows the implementation of a variety of low-level image processing tasks. The availability of pixel-parallel operations makes the development of programs relatively easy, as low-level image processing algorithms are naturally expressed in a pixel-parallel fashion. To illustrate the programming of the SCAMP chip, consider the following simple example.

First, the image $\mathbf{a}$ is acquired (using software-based correlated double sampling). Then, the image $\mathbf{b}$ is obtained as a result of filtering, by convolving the image $\mathbf{a}$ with a smoothing kernel $\mathbf{k}$

$$b_{xy} = \sum_{j=1}^{3}\sum_{k=1}^{3} k_{jk} a_{x+j-2,y+k-2}; \quad \mathbf{k} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Finally, the binary output image $\mathbf{c}$ represents the image $\mathbf{b}$ segmented into two regions: pixels above the arbitrary threshold t are denoted by logic "1" and pixels below the threshold are denoted by logic "0"

$$c_{xy} = \begin{cases} 0, & \text{when } b_{xy} > t \\ 1, & \text{when } b_{xy} \le t \end{cases}.$$

The listing in Fig. 9 illustrates an implementation of the above algorithm in a machine-level language of the SCAMP array.

In a similar way many other early vision algorithms can be implemented. In Fig. 10 the results of sharpening, edge detection, and median filtering algorithms executed on the SCAMP chip are presented. The programs implementing these algorithms contain 15, 29, and 154 instructions, respectively. The APEs work with clock frequencies up to 2.5 MHz, executing one instruction per clock cycle. The execution times, for various algorithms we have implemented [14], are presented in Table I.

As analog operations are performed with an error (and the cumulative effects of errors degrade the overall performance below the equivalent 7-bits accuracy suggested by the single transfer error measured for a register cell) it is interesting to compare the experimental results with "ideal" results, obtained using numerical computations. For the images in Fig. 10 the rms differences between "ideal" and experimental results (allowing for linear brightness/contrast correction and ignoring border effects) are equal to 2.5%, 2.3%, and 1.2%, respectively. Even though the analog computations are performed with a limited accuracy, the end result should be satisfactory for many computer vision applications.

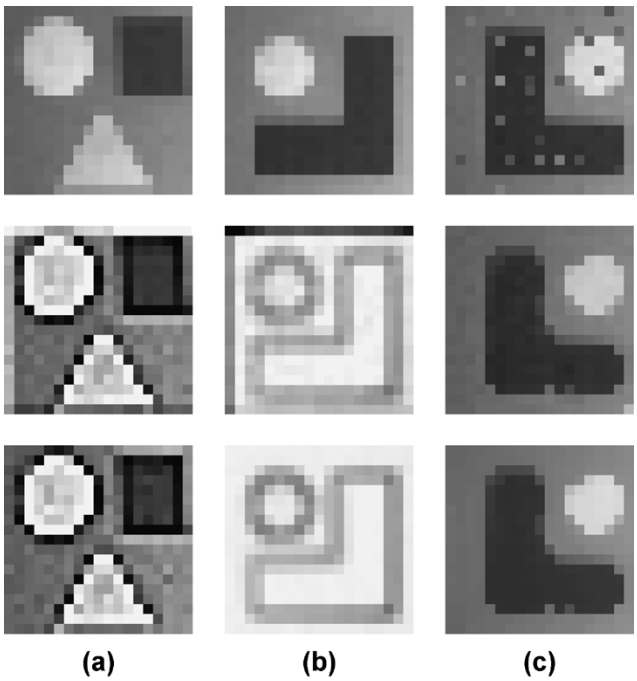


Fig. 10. Image processing examples: (a) sharpening, (b) Sobel edge detection, (c) median filter. Top: acquired image, middle: results of focal-plane processing on SCAMP chip, bottom: results of "ideal" (numerical) image processing.

TABLE I
TIME OF EXECUTION OF SEVERAL ALGORITHMS ON THE SCAMP CHIP (NOT INCLUDING READ-OUT TIME)

algorithm	execution time
Smooth using 3×3 convolution template	5.6 μs
Sharpen using 3×3 convolution template	6.0 μs
Edge detection with Sobel templates	11.6 μs
Median Filter in 3×3 neighbourhood	61.6 μs
Binary Morphology (erosion, dilation)	8.0 μs
Conway's Game of Life (per generation)	13.2 μs
Adaptive threshold (threshold level set based on average value in 6×6 neighbourhood)	31.6 μs
Histogram with 64 bins	205.6 μs
Motion estimation (21×21 global block search matching in horizontal direction, with max. displacement ±3 pixels)	46.4 μs
A/D converter (5-bit conversion, ramp)	130.8 μs
D/A converter (5-bit conversion)	11.2 μs

It also has to be noted that some additional errors may be caused by the decay of the analog values stored in the registers as a result of leakage currents, particularly since the chip is exposed to light. At 125 lux illumination the value stored in the register decreases by approximately 15 nA (i.e., 0.19% of maximum data value) per millisecond. This is usually not very significant since most algorithms only store intermediate results for a very short time (a few clock cycles), however, sometimes it might be necessary to provide longer term storage. For this purpose, APE registers can be used as dynamic digital memories. We have developed software routines for pixel-parallel analog–digital (A/D) conversion, digital–analog (D/A) conversion, and memory refreshing [14]. A measured characteristic of the 5-bit A/D conversion executed in the APE is shown in Fig. 11. The A/D conversion is based on a ramp algorithm comparing the analog values with increasing input levels. An image

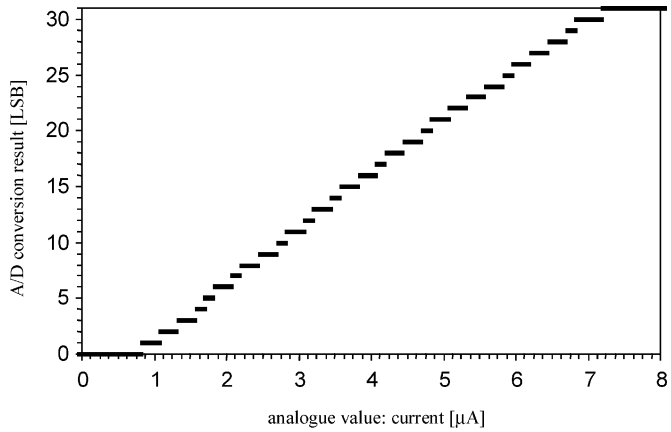


Fig. 11. Characteristic of the 5-bit analog to digital conversion executed on the APE.

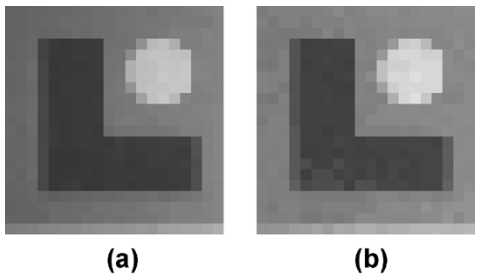


Fig. 12. Parallel in-pixel data conversion: (a) input image (b) output image, after A/D and D/A conversion chain inside the SCAMP array.

reconstructed using the D/A conversion, from digital data stored inside the SCAMP array (obtained as a result of A/D conversion) is shown in Fig. 12. The D/A conversion algorithm is based on accumulating binary weighted input currents, according to the stored binary number. All operations are performed in a pixel-parallel fashion.

V. PERFORMANCE AND COMPARISONS

Each APE executes up to 2.5 MIPS (million instructions per second), which yields a peak performance of over 1100 MIPS per 21 × 21 chip. Peak power dissipation is below 40 mW per chip (with 3.3-V analog and 5-V digital power supply voltages). However, as there is no dc current in an idle APE, power dissipation is much reduced when the time of processing is short compared with the frame rate. So, for example, while performing real-time edge detection at a frame rate of 25 frames/s we obtain power dissipation of 13 nW per APE. This means that the power dissipation figure for our chip can be lower than it is for many application-specific analog vision chips working in continuous time. At the same time, the APE area of 98.6 μm × 98.6 μm is not much larger than the pixel area of many special purpose vision chips, which implement algorithms in hardware [3], [4]. As compared with other programmable SIMD vision chips, the SCAMP approach outperforms the digital SIMD vision chip described in [7], which performs edge detection and smoothing in 5.6 and 7.7 μs, respectively, (using 4-bit numbers and simplified 4-neighbor templates only) while the power dissipation is 27 times larger than that of our chip. Although the bit-serial digital processing elements effectively contain less memory (25-bits)

TABLE II
MAIN PARAMETERS OF THE SCAMP CHIP

Technology	0.6 µm CMOS
Array Size	21×21
Clock frequency	2.5 MHz
Peak Performance	1.1 GIPS
Pixel pitch	98.6µm
Photodetector fill factor	8.4 %
PE density	102 cells/mm^2
Power per PE (maximum)	85 µW
No. of transistors per PE	128
Memory per PE	8 (analogue)
Error (single transfer)	$\approx 0.6\%$
Accuracy (image filtering)	$\approx 2.5\%$

than the APE, the equivalent cell area (150 μm $\times$ 150 μm in 0.35-μm CMOS) is over six times larger than that of the APE. Further comparisons can be made, and it is worth noting that single-bit digital SIMD vision chips with limited memory [5], [6] can achieve smaller cell area. However, they are not as versatile as the SCAMP chip and can mainly be used for simple transformations of binary images. The latest CNN-UM vision chip [11], on the other hand, can process grayscale images and is particularly efficient at executing CNN-type algorithms. Yet its equivalent cell area of 76 μm$\times$73 μm in 0.35-μm CMOS and power dissipation of 150 μW per cell are still somewhat higher than these of the APE.

VI. CONCLUSION

A general purpose programmable vision chip that allows real-time focal plane processing of grayscale images has been presented. The SCAMP chip is an SIMD processor array with an analog data-path. It attempts to combine, in the most efficient way, the flexibility of a software-programmable digital computer and high processing speed, low power dissipation and small cell area that can be achieved using analog circuits. A prototype 21×21 chip has been fabricated, and is fully functional. The main parameters of the chip are summarized in Table II.

The proposed architecture is scalable and even quite large arrays may be integrated onto a single silicon die using contemporary CMOS technologies. Based on the present design it is estimated that a 256×256 array fabricated in a 0.18-μm technology would measure 76 mm^2 and perform 500 GIPS while dissipating 2 W of power.

REFERENCES

[1] A. Moini, *Vision Chips*. Norwell, MA: Kluwer, 2000.

[2] V. Gruev and R. Etienne-Cummings, "Implementation of steerable spatiotemporal image filters on the focal plane," *IEEE Trans. Circuits Syst.s II: Analog Digit. Signal Process.*, vol. 49, no. 4, pp. 233–244, Apr. 2002.

[3] S. Y. Lin, M. H. Chen, and T. D. Chiueh, "Neuromorphic vision processing system," *Electron. Lett.*, vol. 33, no. 12, pp. 1039–1040, Jun. 1997.

[4] C. M. Higgins, R. A. Deutschmann, and C. Koch, "Pulse-based 2-D motion sensors," *IEEE Trans. Circuits Syst.s II: Analog Digit. Signal Process.*, vol. 46, no. 6, pp. 677–687, Jun. 1999.

[5] J. E. Eklund, C. Svensson, and A. Åström, "VLSI implementation of a focal plane image processor—a realization of the near-sensor image processing concept," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 4, no. 3, pp. 322–335, Sep. 1996.

[6] F. Paillet, D. Mercier, and T. M. Bernard, "Making the most of $15k\lambda^2$ silicon area for a digital retina," in *Proc. SPIE*, vol. 3410, 1998, pp. 158–167.

[7] M. Ishikawa, K. Ogawa, T. Komuro, and I. Ishii, "A CMOS vision chip with SIMD processing element array for 1 ms image processing," in *Proc. Int. Solid State Circuits Conf.*, 1999, Paper No. TP 12.2.

[8] P. Dudek and P. J. Hicks, "A CMOS general purpose sampled-data analog processing element," *IEEE Trans. Circuits Syst.s II: Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 467–473, May 2000.

[9] J. Schemmel, K. Meier, and M. Loose, "A scalable switched capacitor realization of the resistive fuse network," *Analog Integrated Circuit Signal Process.*, vol. 32, pp. 135–148, 2002.

[10] A. Dupret, J. O. Klein, and A. Nshare, "A DSP-like analog processing unit for smart image sensors," *Int. J. Circuit Theory Applicat.*, vol. 30, pp. 595–609, 2002.

[11] G. Liñán, S. Espejo, R. Domínguez-Castro, and A. Rodríguez-Vázquez, "Architectural and basic circuit considerations for a flexible 128 $\times$ 128 mixed-signal SIMD vision chip," *Analog Integrated Circuit Signal Process.*, vol. 33, pp. 179–190, 2002.

[12] P. Dudek and P. J. Hicks, "An analog SIMD focal plane processor array," in *Proc. IEEE Int. Symp. Circuits and Syst.*, vol. IV, May 2001, pp. 490–493.

[13] ——, "A general purpose vision chip with a processor-per-pixel SIMD array," in *Proc. Eur. Solid State Circuits Conf.*, Villach, Austria, Sep. 2001, pp. 228–231.

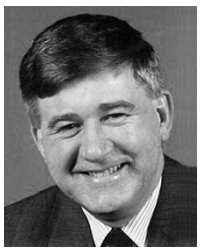
[14] P. Dudek, "A Programmable focal-plane analog processor array," Ph.D. dissertation, Dept. Elect. Eng. Electron., Univ. Manchester Inst. Sci. Technol., Manchester, U.K., May 2000.

[15] J. B. Hughes and K. W. Moulding, "S^{2}I: Aswitched-current technique for high performance," *Electron. Lett.*, vol. 29, no. 16, pp. 1400–1401, Aug. 1993.

Piotr Dudek (S'98–M'01) received the mgr. inż. degree in electronic engineering from the Technical University of Gdańsk, Gdańsk, Poland, in 1997 and the M.Sc. and Ph.D. degrees from the University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., in 1996 and 2000, respectively.

He was a Research Associate at UMIST until 2002 when he became a Lecturer in Integrated Circuit Engineering. His research interests are in analog and mixed-mode VLSI circuits, smart sensors, machine vision, massively parallel processors, cellular arrays, bio-inspired engineering, and spiking neural networks.

Dr. Dudek is a member of the IEEE Circuits and Systems Society Technical Committee on Sensory Systems.



Peter J. Hicks (M'79) received the B.Sc. and Ph.D. degrees from the University of Manchester, Manchester, U.K., in 1969 and 1973, respectively.

He was appointed to the post of Lecturer in the Department of Electrical Engineering and Electronics, University of Manchester Institute of Science and Technology, in 1978 and subsequently promoted to Senior Lecturer in 1985 and Professor of Microelectronic Circuit Design in 1988. In 1996, he was appointed as Vice-Principal of UMIST with responsibility for Information Systems Strategy and in 1999 was appointed as Dean of UMIST. His research interests are in the area of microelectronic circuit design and "systems on silicon" and are mainly focused on integrated sensors and transducers. He has published over 120 papers and articles and has been actively involved in the development of computer-based learning material for use in electronic engineering education.