

AN ANALOGUE SIMD FOCAL-PLANE PROCESSOR ARRAY

Piotr Dudek and Peter J. Hicks

Department of Electrical Engineering and Electronics
University of Manchester Institute of Science and Technology (UMIST)
PO Box 88, Manchester M60 1QD, United Kingdom
p.dudek@umist.ac.uk, p.j.hicks@umist.ac.uk

Abstract

A new smart-sensor VLSI circuit intended for focal-plane processing of grey-scale images is presented. The architecture is based on a fine-grain software-programmable SIMD array. Processing elements, integrated within each pixel of the imager, are implemented utilising a switched-current analogue microprocessor concept. In a $0.6\mu\text{m}$ CMOS process the cell size is equal to $98.6\mu\text{m}\times 98.6\mu\text{m}$. A prototype 21×21 array chip executes over 1.1 GIPS (Giga Instructions Per Second) while dissipating below 40mW of power and demonstrates a real-time performance on a variety of early vision tasks.

1. Introduction

In a way akin to biological vision systems, where preliminary image processing is performed directly on the retina, computer vision systems have recently been build, where some image processing is performed on the focal-plane [1]. Low-level image processing tasks are computationally intensive, but inherently pixel-parallel in nature (identical, localised operations are performed on every pixel). Integrating a processing element within each pixel of the image sensor array enables real-time processing speeds, required in many applications, to be achieved. At the same time, this integration eliminates the I/O bottleneck between the sensor and the processor and reduces the power dissipation, size and cost of the system.

Vision chips are often built as special-purpose devices, performing specific tasks (e.g. smoothing or edge detection) using analogue circuitry [2,3]. However, it can be beneficial to employ instead a versatile device, whose functionality can be easily modified. A suitable architectural model for the software-programmable vision chip is provided by a fine-grain SIMD (Single Instruction Multiple Data) processor array. Each processing element (PE) in such an array contains an execution unit, neighbour communication paths and a number of local memory cells used to store intermediate processing results. The PEs perform in parallel identical instructions issued by a single controller. As the silicon area available for the processing circuitry is very limited in a pixel-per-processor array, the PEs are necessarily very simple. Single-bit digital processors were used in [4] and [5], whereas the “analogic” approaches reviewed in [6] are based upon the CNN (Cellular Neural Network) concept. The approach presented in this paper utilises analogue sampled-data techniques to implement a digital-like processor architecture. The processing elements thus obtained provide a particularly good compromise between cell area and functionality, performance and power dissipation [7].

2. Architecture

The architecture of the proposed system (named SCAMP, SIMD Current-mode Analogue Matrix Processor) is presented in Fig.1. The main feature is a mesh-connected array of analogue processing elements (APEs). Each APE, associated with a single pixel, is an “analogue microprocessor” [7] which comprises six general-purpose registers (each register can store an analogue sample of data), ALU (arithmetic logic unit), nearest neighbour communication register, I/O port, activity flag register and photodetector.

The operations are performed on analogue samples of data, yet the APEs work in a software-programmable SIMD fashion. They execute a sequence of instructions issued by an external digital controller. The APEs support a fairly conventional instruction set, comprising register-transfer operations, arithmetic operations,

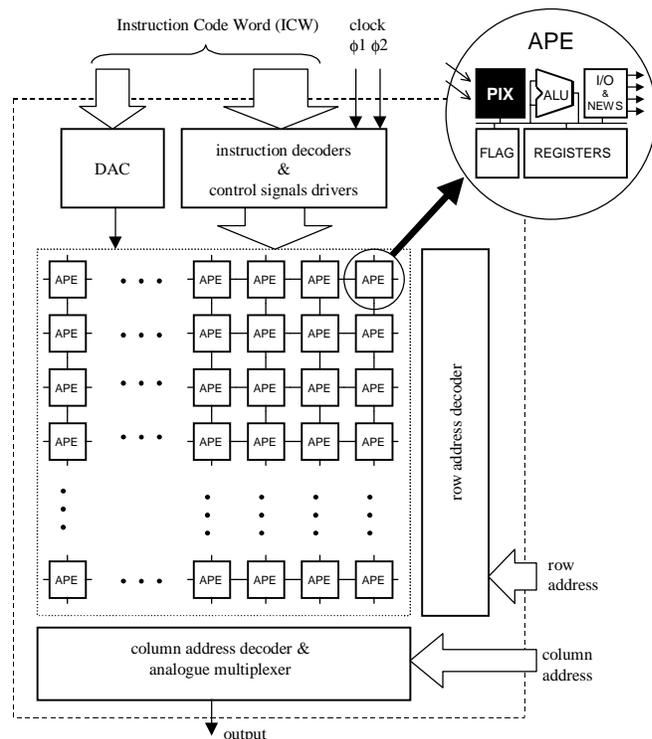


Fig.1. SCAMP chip architecture

neighbour-communication operations and I/O operations (including image acquisition). Conditional branches are supported via an activity flag register, which can be set or reset locally in each APE as a result of a comparison operation.

3. APE Implementation

The APE is implemented as a switched-current (SI) [8] circuit and a simplified schematic diagram of the APE is presented in Fig.2. For clarity, conventional SI cells are shown in Fig.2, although in practice S^2I cells [9] are used in order to reduce the errors of processing originating from charge injection and output conductance effects. Furthermore, additional switches are introduced to make sure that DC biasing currents are turned-off when not needed, thus reducing power dissipation.

a) Basic APE operation

The APE is a discrete-time system in which data is represented as current samples. General-purpose registers are implemented as switched-current memory cells (i.e. they are capable of storing grey-level image data or some other analogue variable). Registers and other functional blocks are connected to the single-wire analogue bus by means of analogue switches. Switches are also used to control other functions of the circuit.

The voltages controlling the switches are derived from the instruction code word, which corresponds to a machine-level program instruction. 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 instruction denoted as $A \leftarrow -B + K$ we close switches W_A , S_A , S_B and S_K (Fig.2.), and following the operation of SI cells we obtain $i_A = -(i_B + i_K)$, i.e. the value stored in the register **A** is an inverted sum of values from the registers **B** and **K**. It is worth noting, that the arithmetic operations of addition and subtraction are performed with no need for explicit ALU circuitry (inversion is inherent in the SI memory cell and addition is performed directly on the analogue 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 analogue processing element can be found in [7].

b) Neighbour 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 analogue buses of four nearest neighbours, thus the current samples can be transferred from one processor 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 neighbour the following instructions are performed:

$NEWS \leftarrow A$; close switches W_O , S_O and S_A
 $B \leftarrow SOUTH$; close switches W_B , S_B and S_N

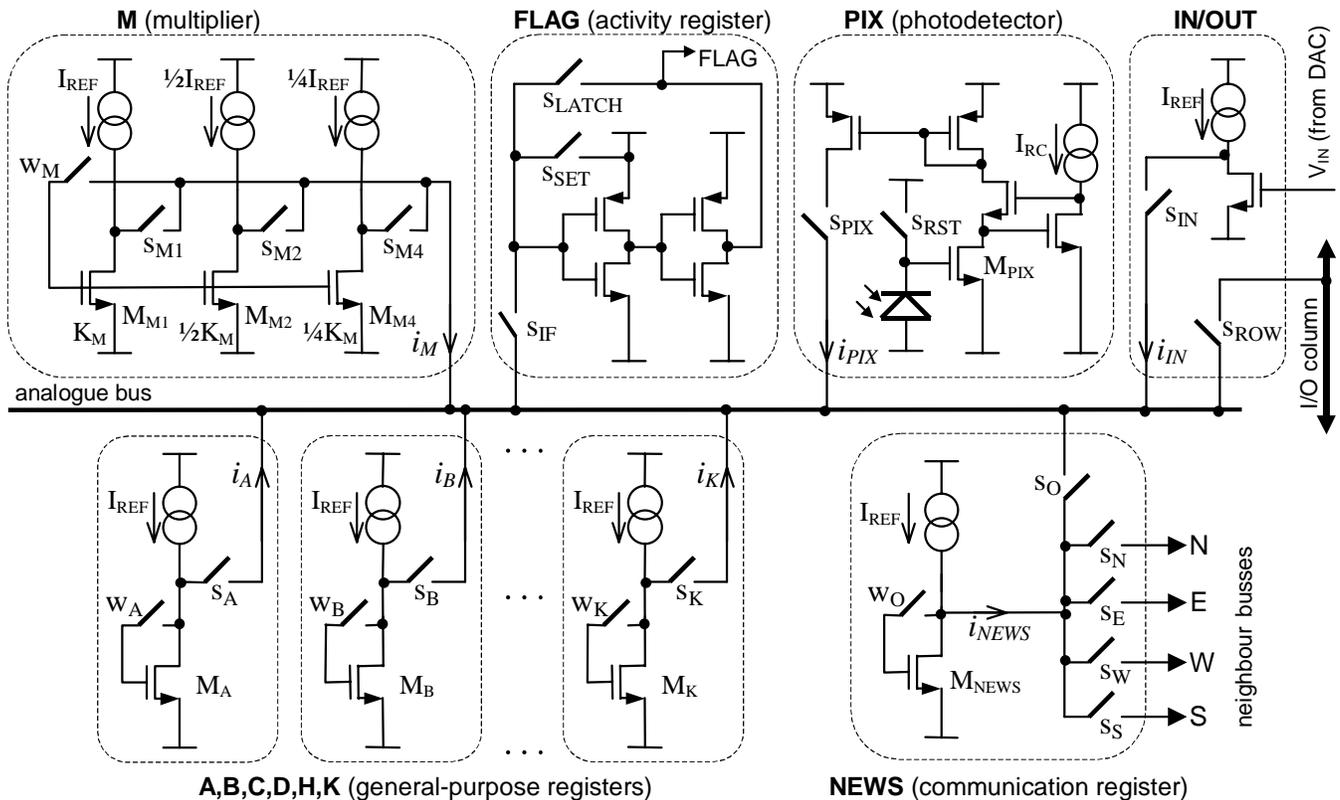


Fig.2. Simplified schematic diagram of the analogue processing element (APE)

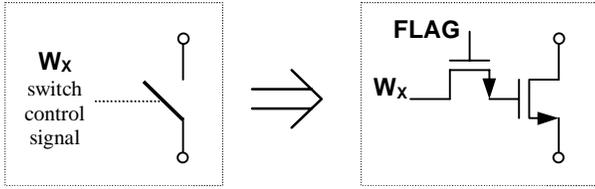


Fig.3. Implementation of conditionally disabled W -switches

c) Activity flag

The APE performs broadcast instructions only if its activity flag is set. This is realised by conditionally disabling the storage operation. To achieve this, all W -switches are implemented using the circuit arrangement shown in Fig.3. (NB. To ensure good analogue switch performance the switch-control voltages are derived from the 5V digital power supply whereas the nominal analogue power supply voltage is 3.3V). It can be seen that a W -switch will be closed only if both the corresponding control signal and the FLAG signal are active. If FLAG is not active, then the content of the registers in the APE cannot change and the APE remains disabled.

The flag register (see Fig.2) is implemented as a D-latch. It can be set globally by instruction **ENDIF** (which closes switch S_{SET}) or conditionally by a comparison instruction **IF X**, where $X \in \{A, B, C, \text{etc.}\}$. When the current from a selected register X is routed to the high-impedance flag register input node (S_{IF} and S_X closed), this node is charged high or low, depending on the sign of the current from that 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 FLAG signal, which is latched by closing S_{LATCH} .

d) Pixel

The photodetector (**PIX** circuit in Fig.2) works in an integration mode. The voltage on the gate capacitance of M_{PIX} is reset by closing switch S_{RST} (instruction **RPIX**). Then 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 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 read-out to the analogue bus (by closing 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 simple subroutine at the beginning of each video frame:

```

A←PIX      ; sample integration result into A
RPIX       ; reset photodetector
B←A+PIX    ; calculate difference & store output image in B

```

e) Random access and global I/O

The array supports random access analogue and digital I/O. The analogue bus of an APE is connected to the array column bus via an access switch S_{ROW} controlled by a row-select signal. One column

of the array is selected using an analogue column-select multiplexer. (Additionally, column-parallel analogue 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 that can be used to extract global image information or image features.

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

Digital output, random-access digital input and analogue input are also possible via a combination of the random-access feature, immediate argument generation and conditional instructions [10].

4. Experimental Results

A prototype SCAMP chip was fabricated in a 0.6 μm technology. The chip comprises a 21 \times 21 array of APEs, random-access I/O logic, a DAC and control logic. By combining the instruction code word with appropriate phases of the clock signals the switch-control signals are generated and distributed to the APEs using separate drivers for each row and column of the array. The chip can be easily scaled-up to a larger array size.

Each APE contains 128 transistors. The design of the APE involves trade-offs between size, power dissipation and accuracy of processing. In the present implementation the APE size is equal to 98.6 μm \times 98.6 μm . The photodiode area is equal to 820 μm^2 , which yields a fill factor of 8.4% (the sensitivity is further reduced by metal wires that pass over the photodiode area). With 1000 lux illumination level full-contrast images are obtained at 25 frames/second. The measured fixed pattern noise of the imager, with correlated double sampling, is equal to 1% rms.

a) Accuracy

Switched-current cells are characterised by a limited accuracy. The cells were carefully laid-out to minimise parasitic capacitances. The magnitude of the signal-dependent error of the register transfer operation in the APE was measured to be equal to approximately 40nA, that is 0.5% of the maximum signal level of 8 μA . Each transfer also contributes a noise of 8.5nA rms (i.e. 0.11%).

Leakage currents cause a decay of analogue values stored in the registers at a rate of 15nA per ms, at 125 lux, however this is not very significant since most algorithms only store intermediate results for a very short time. Alternatively, analogue registers can be used as dynamic digital memories, which together with analogue-digital conversion, digital-analogue conversion and memory refreshing routines that can be executed on the APE [10] provide a way to realise long-term storage.

b) Processing examples & performance

A software-programmable architecture allows the implementation of a variety of low-level image processing tasks, e.g. convolution, filtering, edge detection, segmentation, morphological operations, histogramming and histogram modifications, motion detection and

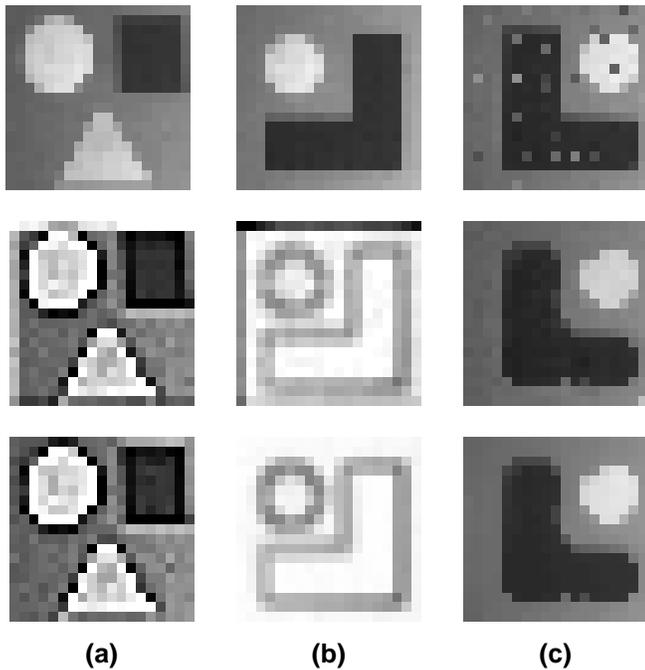


Fig.4. Image processing examples: (a) sharpening, (b) Sobel edge detection, (c) median filter. Top: acquired image, Middle: results of focal-plane processing on SCAMP chip, Bottom: results of "ideal" (numerical) image processing.

estimation, etc. In Fig.4 the results of sharpening, edge detection, and median filtering algorithms executed on the SCAMP chip are presented. As analogue operations are performed with an error it is interesting to compare the experimental results with "ideal" results (obtained using numerical computations). For the images in Fig.4 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 analogue computations are performed with a limited accuracy, the end result should be satisfactory for many computer vision applications. On the other hand, massively parallel focal-plane computing results in very high processing speeds. The APEs work with clock frequencies up to 2.5MHz, which yields a peak performance of over 1100 MIPS (Million Instructions Per Second) per 21x21 chip. Peak power dissipation is below 40mW per chip, however it can be much reduced depending on the frame

TABLE I. Execution time and power dissipation for SCAMP chip performing exemplary early-vision algorithms (figures do not include read-out time).

algorithm	execution time @ 2.5MHz clock	power per pixel @ 25 frames/sec
Smooth using 3x3 convolution template	5.6 μ s	15 nW
Edge detection Sobel templates	11.6 μ s	25 nW
Median Filter in 3x3 neighbourhood	61.6 μ s	150 nW

rate and algorithm being performed. In Table I the processing time and power dissipation for the above-mentioned low level image processing tasks are listed.

5. Conclusions

A smart sensor chip that allows real-time focal-plane processing of grey-scale images has been presented. With the pixel area not much larger than that of many special-purpose vision chips, a completely software-programmable solution has been obtained. The APEs, which are built utilising switched-current analogue processing techniques, are more compact than the processing elements of a digital SIMD array with comparable capability [5] and the CNN-based visual microprocessors [6]. Another advantage of the proposed solution is low power dissipation. A prototype 21x21 chip has been fabricated, however the proposed architecture is scalable and even quite large arrays could be integrated onto a single silicon die using present-day CMOS technologies. Based on the present design it is estimated, that a 256x256 array fabricated in a 0.18 μ m technology would measure 76mm² and perform 500 GIPS while dissipating 2W of power.

References

- [1] A.Moini, "Vision chips ", Kluwer Academic Publishers, Boston, 2000
- [2] P.Yu, S.J.Decker, H.S. Lee, C.G.Sodini, J.L.Wyatt Jr., "CMOS Resistive Fuses for Image Smoothing and Segmentation", in *IEEE Journal of Solid-State Circuits*, vol.27, no.4, pp.545-553, April 1992.
- [3] S.Y.Lin, M.H. Chen and T.D. Chiueh, "Neuromorphic vision processing system", in *Electronics Letters*, vol.33, no.12, pp.1039-1040, June 1997
- [4] F. Paillet, D. Mercier, and T.M.Bernard, "Making the most of 15k λ^2 silicon area for a digital retina", Proc. SPIE, Vol. 3410, Advanced Focal Plane Arrays and Electronic Cameras, AFPAEC'98, 1998
- [5] M.Ishikawa, K.Ogawa, T.Komuro, I.Ishii, "A CMOS Vision Chip with SIMD Processing Element Array for lms Image Processing", Proc. International Solid State Circuits Conference, ISSCC'99, TP 12.2, 1999.
- [6] T.Roska and A.Rodríguez-Vazquez, "Review of CMOS Implementations of the CNN Universal Machine-Type Visual Microprocessors", Proc. International Symposium on Circuits and Systems, ISCAS 2000, pp.II-120-II-123, Geneva, Switzerland, May 2000.
- [7] P.Dudek and P.J.Hicks, "A CMOS general-purpose sampled-data analog processing element", *IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing*, vol. 47, no. 5, May 2000, pp. 467-473
- [8] C. Toumazou, J. B. Hughes and N. C. Battersby (Eds.), "Switched-Currents: An Analogue Technique for Digital Technology", Peter Peregrinus Ltd., London, 1993.
- [9] J. B. Hughes and K. W. Moulding, "S²I: A Switched-Current Technique for High Performance", in *Electronics Letters*, vol.29, no.16, pp.1400-1401, August 1993.
- [10] P.Dudek, "A programmable focal-plane analogue processor array", Ph.D. Thesis, UMIST, Manchester, May 2000.