

Q1 Energy aware simplicial processor for embedded morphological visual processing in intelligent internet of things

M. Villemur, P. Julian[✉] and A. G. Andreou

Q3 This Letter presents the architecture implementation and testing of an SIMD processor for energy aware embedded morphological visual processing using the simplicial piece-wise linear approximation. The architecture comprises a linear array of 48×48 processing elements, each connected to an eight-neighbour clique operating on binary input and state data. The architecture is synthesised from a custom designed ultra low-voltage CMOS library and fabricated in a 55 nm CMOS technology. The chip is capable of dynamic voltage/frequency scaling with power supplies between 0.5 and 1.2 V. The fabricated chip achieves an overall performance of 293 TOPS/W with dynamic energy dissipation efficiency of 3.4 fJ per output operation at 0.6 V.

Introduction: Energy aware embedded visual processing is necessary for systems where endowing cameras with local processing capability is crucial for operational autonomy, such as in autonomous or semi-autonomous vehicle navigation [1], processing data from wide area motion imagery [2], and feature-driven learning/intelligent processing in intelligent Internet of things [3] (I^2oT). IoT devices are often thought as the ‘edge’ of a large sophisticated cloud processing infrastructure. Processing image data at the edge reduces system latency by removing the delays in the aggregation tiers of the IoT infrastructure. In addition to minimising latency, edge processing increases system security and mitigates privacy concerns when processing data in the cloud. Autonomous operation and decision making coupled with real-time ability to do local-processing before transmitting the data/information necessitate feature-driven ‘intelligent’ sensing nodes that have extreme energy efficiency. This study presents an energy efficient SIMD digital accelerator for morphological binary data processing. The primary application target is low-level large-scale parallel data processing/acceleration in embedded applications.

Theoretical foundations: Morphological image processing [4] is a large discipline in computer vision that has its foundation the rich theory of mathematical morphology [5]. Morphological operations process input images based on shape of object features in the image (morphi-μορφη is the Greek word for shape). Dilation and erosion are the most common morphological operations. Other binary set-theoretic operations such as complement (NOT), intersection (AND) and union (OR) can be included to form more complex morphological processing functions. While most applications of mathematical morphology are in the field of digital image processing, the theory can be applied to process and transform spatial structures, for example graphs with applications in large-scale graph analytics.

The morphological processor core described in this Letter consists of a 2D-array of 48×48 processing elements (PEs). Every PE operates on nine 1-bit inputs: the input corresponding to the PE itself and the eight inputs corresponding to the neighbours. In particular, the PE implements a simplicial piecewise linear (PWL) function approximation of a symmetric non-linear function of nine inputs [6], i.e.

$$y = f(x_1, \dots, x_9) \quad (1)$$

A symmetric function does not change under a permutation of its arguments [7]

$$y = f(x_{i_1}, \dots, x_{i_9}) = f(x_{j_1}, \dots, x_{j_9}) \quad (2)$$

for any two sets of indices $\{i_1, \dots, i_9\}$, $\{j_1, \dots, j_9\}$, such that $i_k, j_k \in \{1, \dots, 9\}$, $i_k \neq i_l, j_k \neq j_l$ if $k \neq l$. For example, the functions $y = |x_1 - x_2|$, $y = \max(x_1, x_2) = 0.5(x_1 + x_2 + |x_1 - x_2|)$, $y = \min(x_1, x_2) = 0.5(x_1 + x_2 - |x_1 - x_2|)$, $y = 1 - \max(x_1, x_2) = 1 - 0.5(x_1 + x_2 + |x_1 - x_2|)$ depicted in Figs. 1a–d are symmetric. Notice that if the inputs are restricted to digital values, i.e. $\{0,1\}$, then the functions previously introduced are actually: XOR(x_1, x_2), OR(x_1, x_2), AND(x_1, x_2), and NAND(x_1, x_2). Conversely, the function $y = x_1 + 2x_2$ is an example of a non-symmetric function.

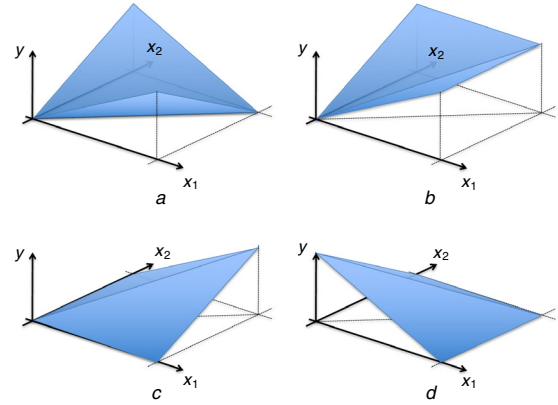


Fig. 1 Symmetrical functions on simplicial domain

$$\begin{aligned} a \quad y &= |x_1 - x_2| \\ b \quad y &= \max(x_1, x_2) = 0.5(x_1 + x_2 + |x_1 - x_2|) \\ c \quad y &= \min(x_1, x_2) = 0.5(x_1 + x_2 - |x_1 - x_2|) \\ d \quad y &= 1 - \max(x_1, x_2) = 1 - 0.5(x_1 + x_2 + |x_1 - x_2|) \end{aligned}$$

By definition, a symmetric function is independent of the order of the inputs, thus it can be implemented using the simplicial algorithm [6] with only one simplex; for example, the one defined as $S = \{x \in R^9: 0 \leq x_1 \leq x_2 \leq \dots \leq x_9 \leq 1\}$. A symmetric function (2) can only represent a subset of all possible nine-input functions; however, any generic function can be implemented by composing different symmetric functions and masking inputs adequately. On the other hand, a symmetric PWL function of N inputs has an efficient representation, since only needs $N + 1$ parameters. In fact, the function in (2) implemented by each PE is specified uniquely by the 10 (1-bit) parameters $\{f_0, f_1, \dots, f_9\}$, defined as the values of f at points $v_0 = (0, 0, \dots, 0)$, $v_1 = (0, 0, \dots, 1)$, \dots , $v_8 = (0, 1, \dots, 1)$, $v_9 = (1, 1, \dots, 1)$. These values, which can be thought of as the operation codes for a particular programme instruction, are stored in a register file outside the array. During normal operation, the bits in the register file are broadcasted serially to every PE; the internal circuits detect the value the PE needs and stores it in an internal register for computation. This implementation strategy avoids the storage of the 10 bits in every PE, therefore trading PE size for computation time: one programme instruction requires 10 clock cycles.

Simplicial morphological processor architecture: Every PE has three 1-bit registers that can be used as inputs, namely X, U and T in Fig. 2. The output of the selected register is collected with the corresponding signals from the eight neighbour PEs. These nine signals are bitwise operated using an AND with nine global mask signals $b_i \in \{0, 1\}$, $i = 1, \dots, 9$ and the resulting bits are added producing a 4-bit value that is the PE function argument:

$$\text{arg}(f) = b_1x_1 + b_2x_2 + \dots + b_9x_9 \quad (3)$$

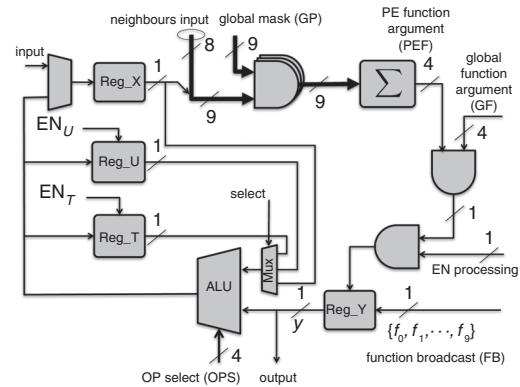


Fig. 2 Architecture of single simplicial morphological processor cell

Notice that $0 \leq \text{arg}(f) \leq 9$. The PE function argument signal is compared with the broadcasted global function argument signal. When they coincide, the function value f_i of the function broadcast signal is stored in register Y. This output together with the three input registers enter the

ALU and are operated with a selectable operation. The output of the ALU can be fed back to the input registers. The ALU can implement 16 functions, namely OR, NOR, NAND, XOR of each of the three input registers with output y , and also can route directly the signals X , U , T and y . Fig. 3a shows the layout of the morphological processor; Fig. 3b shows the micro-photograph of the chip. The core area is $800\ \mu\text{m} \times 700\ \mu\text{m}$.

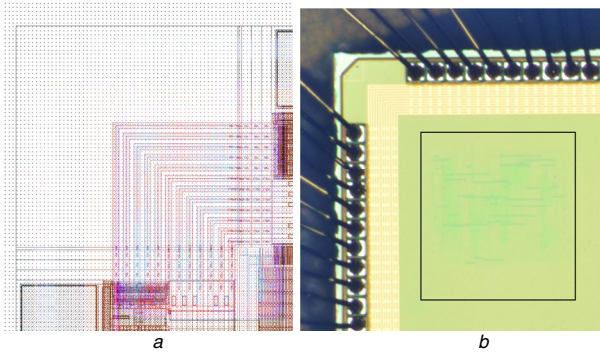


Fig. 3 Integrated circuit

a Layout
b Micro-photograph

Experimental results: The processor core was fabricated in a 55 nm technology and tested using a custom designed board with a Spartan 3 FPGA. The specifications of the fabricated test chip are listed in Table 1.

Table 1: Simplicial morphological processor specifications

Technology	Size (mm ²)	Transistors	PE
55 nm	0.85 × 0.65	1.43 M	48 × 48

Table 2: Simplicial morphological processor performance

Supply	Clock	Power	GOPS	E/OP	TOPS/W	FPS
0.6 V	75 MHz	3 mW	881	3.40 fJ	293	7.5 M

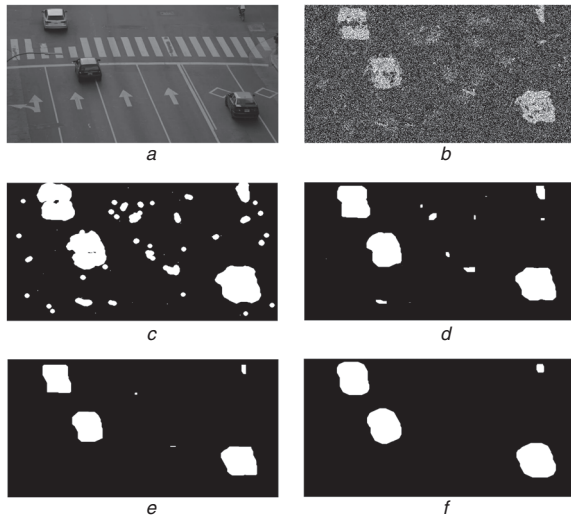


Fig. 4

a Original image frame
b Image with added noise
c-f Output frames corresponding to programmes 26, 50, 68 and 107, respectively

The processor operates at 75 MHz at a 0.6 V power supply and consumes 3 mW. Considering that it takes 10 clock cycles to execute one instruction over the entire array-frame- (regardless the array size), the chip can process 7.5 M frames/s. In terms of performance, every PE computation is equivalent to 50 (2-input) logic gates and a 1-bit accumulate every 10 cycles. If we define an elementary 1-bit output operation (OP) as that of a 2-input logic gate, the array performs $51 \times 48 \times 48 \times 75 \times 10^6 \div 10 = 881$ GOPS, reaching 293 TOPS/W. Table 2 summarises the performance. Figs. 4c-f show an example of a sequence of 107 erosion and dilation instructions to extract cars

from an intentionally corrupted image (Fig. 4b) of traffic (Fig. 4a); Table 3 lists the instructions used and the value of operation codes. In the latter example, it takes 1070 cycles to complete the processing for an entire frame, hence the processing to extract blobs (objects) from a complex video sequence can be done at a rate of 70,000 frames/s with an energy efficiency of $18.52\ \text{pJ} \div 5457\ \text{OPS} = 3.4\ \text{fJ/OP}$.

Table 3: Sequence of programmes for example in Fig. 4

Programme instruction	Op. code $\{f_0, \dots, f_9\}$	Cycles	OP	E (pJ)	Operation
1-3	0 0 0 1 1 1 1 1 1 1	30	153	0.52	erosion
4-18	0 0 1 1 1 1 1 1 1 1	150	765	2.60	dilation
19-23	0 0 0 0 1 1 1 1 1 1	50	255	0.86	erosion
24-33	0 0 1 1 1 1 1 1 1 1	100	510	1.73	dilation
34-45	0 0 0 0 1 1 1 1 1 1	120	612	2.08	erosion
46-53	0 0 1 1 1 1 1 1 1 1	80	408	1.38	dilation
54-68	0 0 0 0 1 1 1 1 1 1	150	765	2.60	erosion
69-82	0 0 1 1 1 1 1 1 1 1	140	714	2.42	dilation
83-89	0 1 1 1 1 1 1 1 1 1	70	357	1.21	dilation
90-107	0 0 0 0 0 0 0 0 0 1	180	918	3.12	erosion
Total	—	1070	5457	18.52	—

Discussion and conclusions: We have designed, fabricated and tested a 1-bit morphological processor core in 55 nm technology. While the processor was designed with the general class of morphological processing tasks in mind, the architecture can also be employed for other algorithms operating with binary data as well as to processing using local binary patterns [8]. This Letter extends and complements our previous work [9, 10] where we also discuss how such a processor core can be programmed at the high level using basic morphological primitives.

Acknowledgment: This work was partially supported by PICT 2657-2010 and 2009-2016 of ANPCyT/MINCYT Argentina, by the NSF grant SCH-INT 1344772, by ONR MURI N000141010278 and DARPA UPSIDE HR0011-13-C-0051 project.

© The Institution of Engineering and Technology 2018
Submitted: 24 December 2017
doi: 10.1049/el.2017.4738

One or more of the Figures in this Letter are available in colour online.

A. G. Andreou (Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218, USA)

M. Villemor and P. Julian (Departamento de Ingeniería Eléctrica y Computadoras, Universidad Nacional del Sur and IIIE, CONICET, Bahía Blanca, Argentina)

✉ E-mail: pjulian@uns.edu.ar

References

- Bojarski, M., *et al.*: 'End to end learning for self-driving cars', arXiv.org, April 2016
- Andreou, A.G., Figliolia, T., Sanni, K., *et al.*: 'Bio-inspired system architecture for energy efficient, BIGDATA computing with application to wide area motion imagery'. Proc. 2016 IEEE LASCAS, 2016, pp. 1-6
- Atzori, L., Iera, A., and Morabito, G.: 'The internet of things: a survey', *Comput. Netw.*, 2010, **54**, (15), pp. 2787-2805
- Maragos, P.: 'Tutorial on advances in morphological image processing and analysis', *Opt. Eng.*, 1987
- Serra, J.: 'Online course on mathematical morphology'. 2017, Available at <http://cmm.enscm.fr/~serra/cours/pdf/en/ch1en.pdf>
- Julian, P., Dogaru, R., and Chua, L.O.: 'A piecewise-linear simplicial coupling cell for CNN gray-level image processing', *Trans. Circuits Syst. I Fundam. Theory Appl.*, 2002, **49**, (7), pp. 904-913
- David, F.N., Kendall, M.G., and Barton, D.E.: 'Symmetric function and allied tables' (Cambridge University Press, 1968)
- Ojala, T., Pietikainen, M., and Harwood, D.: 'A comparative study of texture measures with classification based on featured distributions', *Pattern Recognit.*, 1996
- Federico, M.D., Mandolesi, P.S., Julian, P., *et al.*: 'Experimental results of simplicial CNN digital pixel processor', *Electron. Lett.*, 2008, **44**, (1), pp. 27-29
- Mandolesi, P.S., Julian, P., and Andreou, A.G.: 'A scalable and programmable simplicial CNN digital pixel processor architecture', *Trans. Circuits Syst. I Regul. Pap.*, 2004, **51**, (5), pp. 988-996

EL20174738

Author Queries

M. Villemur, P. Julian, A. G. Andreou

- Q1** Please confirm the changes made in the article title. Please note that it is the IET's house style to remove words such as 'Novel', 'New' and 'Study of' as well as 'A', 'An' and 'The'.
- Q2** Please check authors and their affiliation.
- Q3** Please expand the abbreviation SIMD.
- Q4** As per journal style references are renumbered in the text and reference list. Please confirm.
- Q5** As per journal style, names of up to three authors are provided. If there are more than three, only the first three should be given followed by et al. Please provide next author names in Ref. [1] as required by journal style.
- Q6** Please provide volume number and page range in Refs. [4 and 8].
- Q7** Please provide place of publisher in Ref. [7].