Fast and Efficient Implementation of Trigger-Wave Propagation on VLSI Cellular Processor Arrays

Piotr Dudek

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

ABSTRACT: *In this paper circuit implementations of cellular processor arrays intended for image processing applications are discussed. It is demonstrated that a departure form the standard CNN model can lead to a significant improvement when processing binary (black/white) images. An asynchronous cellular logic array circuit is presented, which is capable of simulating trigger-waves in an excitable medium. The circuit is implemented using CMOS dynamic logic circuits, and exhibits high speed and low power consumption. Simulation results are presented.*

1. Introduction

The cellular neural networks (CNN) have provided a convenient paradigm for the modelling of spatio-temporal phenomena. The CNN-Universal Machine (CNN-UM) has enabled the use of these phenomena in image processing applications and has established a foundation for the development of general-purpose visual microprocessors [1-2] that integrate image sensors and processing circuitry in a pixel-per-processor array. At the same time, there is a continuing research interest in the development of general-purpose analogue [3] and digital [4,5] sensor/processor arrays based on more conventional cellular computer architectures, which origins might be traced back to the von Neumann's work on cellular automata [6] and early SIMD (single instruction multiple data) computers [7,8]. In reality, there is no hard-line distinction between the two approaches, particularly since the original Chua and Yang's CNN formulation has been extended to a much wider definition of a cellular *nonlinear* network [9]. Processor arrays based on discrete-time CNNs [10] and Boolean logic cells [11] were proposed, while asynchronous (continuous-time) operations were used in digital processor arrays [12,13]. Indeed, the constraints set by the limited silicon area and required low power consumption in massively parallel pixel-processors, imply that in a search for the most optimal solution the designers combine various approaches [14] and simplify the models [2] rather than providing a faithful implementation of a particular model. Nevertheless, the CNN approach, with the functionality determined by the cloning template (A, B, bias), is still often used to provide a framework for the VLSI implementations of image processing systems.

1.1. Binary Image Processing on CNN chips

A salient feature of a standard CNN is its ability to solve a complex spatio-temporal mathematical problem using simple circuitry. Where the processing task requires solving certain partial differential equations, then the speed-up offered by the CNN chip, as opposed to a numerical calculation on a digital computer, is very significant. A great number of practical computer vision problems, however, can be efficiently solved using alternative methods. These issues were discussed in some more detail in [14], they are also discussed in [15], where the problem of using CNN-based computers for grey-level filtering is addressed.

The majority of practical spatio-temporal operations, that are being solved by the CNNs in the image processing context, are those that operate on binary (i.e. black/white) images. This is consistent with the overall make-up of a majority of useful CNN templates, which produce a stable solution when all the cell outputs are saturated. In particular, binary trigger-wave propagation has been demonstrated to provide a powerful image processing tool, offering a “single-instruction” solution to various object detection and recognition problems [16]. From the integrated circuit implementation point of view, concentrating on this class of problems also offers certain advantages. One of the most difficult tasks, when designing high-density pixel-per-processor cellular arrays, is to ensure good accuracy of operation despite the fact that the accuracy of analogue circuits reduces as the circuit area is reduced. It has been shown, that robust operation can be achieved with much relaxed accuracy requirements if only binary operations are required [2].

It should be noted, that problems involving binary propagation can be always solved in a discrete-time iterative way by cellular automata (or a general-purpose SIMD machine). Indeed, even the CNN processor chips are sometimes used in this iterative fashion, if more complicated operations are required during the wave-propagation [16,17]. But the continuous-time propagation offers significant advantages in terms of the speed of operation. It has been also pointed out, that the discrete-time solution is not efficient [13]. The standard CNN approach, which uses essentially analogue calculations (“solving differential equations”) to perform Boolean operations, is inherently not efficient also. Not only it consumes power performing redundant operations, the cell circuitry also draws dc currents when all inputs and output remain in a saturated state, i.e. dissipates static power. CNNs have been proposed, that can be used to implement any Boolean function in a single cell. However, the discrete-time approaches [10] can’t perform asynchronous propagations at all, while the continuous-time approaches [11] still rely on analogue circuits, besides they prove to be difficult to implement due to mismatch problems [18]. The programming of analogue CNNs, involving distribution of analogue signals over large chip area, is also a difficult task.

In a significant effort towards providing cellular processor architecture well suited to VLSI implementation, circuits implementing a simplified CNN model have been developed [2]. Monotonic, binary-valued CNNs are of particular interest [19]. As has been observed, these simplified CNNs are not as versatile as their full-range counterparts, but they do enable the execution of the vast majority of useful propagating templates reported so far. It should be noted, that the functionality of some other templates (e.g. ‘smallkiller’) is easily implemented through a sequential execution of a small number of local logic operations.

The efficiency of implementation is of critical importance when it comes to practical implementations of VLSI systems. If we consider a single-chip massively-parallel pixel-per processor array, then both the circuit area and the power consumption have to be optimised. To that end, concentrating on useful image processing operations, rather than providing general but hardly-ever used capabilities, is appropriate when engineering a visual microprocessor system. Thus motivated, in the following section of this paper we describe an approach to the design of an asynchronous cellular logic array (ACLA), capable of executing an important generic operation - trigger-wave propagation. Building the array from a “cellular logic” rather than “cellular neural” perspective, and consequently using digital rather than analogue circuit techniques, we demonstrate that power and speed of operation can be optimised.

2. Asynchronous Cellular Logic Network

Consider a cellular array, on a rectangular grid with a 4-neighbour topology. The ‘next state’ of each cell, y, is described by the following Boolean equation

$$y = (y_N + y_E + y_W + y_S) \cdot u \qquad (1)$$

Where u is the input of each cell, and where y_N, y_E, y_W and y_S are the values of the 'present state' (in the North, East, South and West direction, respectively). This operation describes the propagation of a trigger-wave in the *state array* **y**, with the initial state equal to the *marker image* **m**, and where the propagation can be locally inhibited, depending on the state of the *input image* **u**. In image processing context this propagation operation is very useful; it can be used for geodesic reconstruction of an object from a marker, or as a part of hole detection (or closed-curve detection) routine.

The evolution of a state cell **y** in equation (1) is always monotonic. The 'next state' calculation could be either synchronous (like in cellular automata) or asynchronous (like in binary-valued CNNs). In case of asynchronous propagation, the combinatorial logic circuit in Figure 1 could be used. We will refer to the array comprising circuits shown in Figure 1 as the asynchronous cellular logic array (ACLA).

2.1 Cell Circuit

The functionality of the ACLA can be easily obtained in CNN processor chips [1,2], by using an appropriate cloning template. The same functionality is also available in the global logic unit presented in [12]; the circuit described in [13] also achieves equivalent operation. Here, however, we will illustrate how a compact, high-speed and low-power implementation of the propagation operation can be achieved. The ACLA cell is implemented using dynamic logic circuit shown in Figure 2, and a domino-effect is exploited to optimise the power efficiency of the entire network. The operation consist of a precharge phase, and a propagation phase. During precharge, the inputs u and m are initially set to logic '0' and the node y is discharged to ground by opening the switch M_d. Then, the node x is charged-up to V_{DD} by opening the switch M_p. The switches M_d and M_p are then closed and the input image is applied to the input u. The propagation is initiated by applying marker image to inputs m of the cells. This will result in triggering the propagation, since x becomes '0' at the marker locations (the node

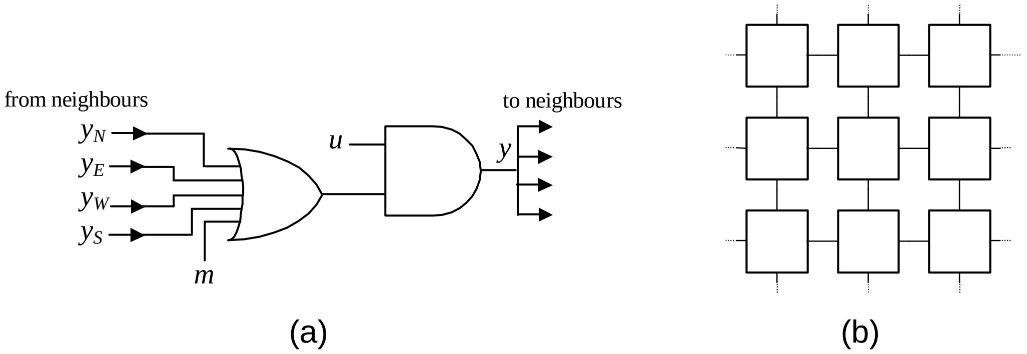

Figure 1. (a) Logic diagram of a single cell in the Asynchronous Cellular Logic Array

(b) Network topology

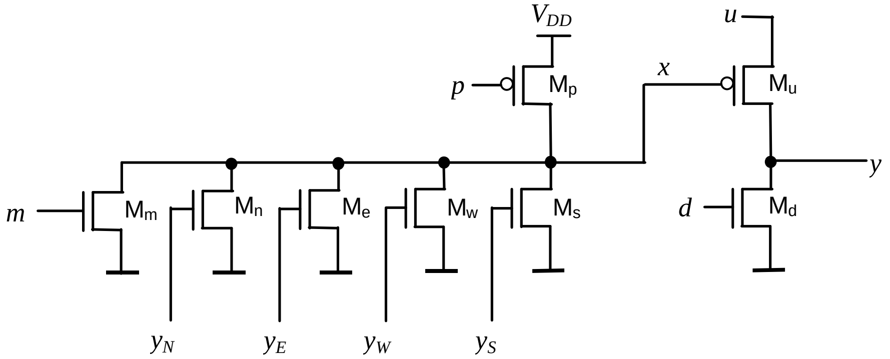

Figure 2. Schematic diagram of the cell circuit

x is discharged via M_m), which in turn leads to the output *y* switching to ’1’ (node *y* charged-up via M_u) if the input *u* at this location is equal to ’1’. The output *y* going high will then trigger neighbour cells, their *x* nodes being discharged to ‘0’ via transistors M_n, M_e, M_w and M_s. This will in turn lead to their *y* nodes being charged up to ‘1’, which will trigger further neighbour cells and so forth. Note, that this propagation is conditioned by the state of the input *u* at each cell, since the node *y* can be only charged up to ‘1’ if the input *u* is equal to ‘1’. Otherwise (when *u* = ’0’) the cell output *y* remains equal to ‘0’. The propagation continues in the domino-effect fashion (note that the state evolution is monotonic, and always from ‘0’ to ‘1’) until the trigger-wave reaches all cells belonging to objects defined by **u**, marked by **m**. The result of the processing can then be read from the output **y**.

2.2 Efficiency Issues

The essential feature of the circuit in Figure 2 is, that it minimises the overall power consumption associated with the trigger-wave propagation. There are only two nodes per each cell (*x* and *y*) where the logic functions are evaluated, and each of the circuit nodes is charged-up to V_{DD} and discharged to ground only once (at most) during the entire propagation. There are no static currents (apart from leakage currents) in the circuit, and at no point (even during switching) there is a direct path from V_{DD} to ground. The total energy per cell, required to carry out the operation can be therefore estimated as energy required to charge up and down (between 0 and V_{DD}) the capacitances of the six circuit nodes (*x*, *y*, *d*, *p*, *u* and *m*). In most cases the total energy will be smaller, as it depends on the number of 1’s and 0’s in the input, output, and marker images in each particular case.

Some modifications to the basic ACLA circuit are also possible [20], which reduce power consumption even further. Simple functional modifications are also possible. Introduction of additional switches, that should be placed in series with transistors M_n, M_e, M_w, M_s and M_u, creates dynamic latches, which enables synchronous operation - useful if a more sophisticated control of the propagation operation is required. At the same time, additional control signals can be used to selectively disable/enable the propagation in each direction. In fact, such circuit offers a compact implementation of a global logic unit proposed in [12]. The control signals could be also viewed as coefficients of the A-template in a simplified CNN. The power consumption, however, remains at the absolute minimum.

3. Simulation Results and Discussion

The operation of the ACLA circuit has been confirmed via computer simulations. The cell was modelled using minimum-size transistors from a standard 0.35μm CMOS technology. A 20×20 array of cells has been simulated with SPICE, with a supply voltage V_{DD} set at 2.5V (lower voltage operation is easily achievable). The effect of the wave propagation in the network is illustrated in Figure 4. This figure was obtained by plotting the results of the transient SPICE simulation, so that the voltages at nodes *y* of the cells form images, in which the pixel brightness corresponds to the voltage level, white is 2.5V (logic ‘1’), black is 0V (logic ‘0’). In this simulation the inputs *u* were set to ‘1’ and the propagation was triggered by setting a single-pixel marker in the centre of the array.

The geodesic reconstruction operation performed by the proposed circuit is illustrated in Figure 5. The input image containing several objects was applied to the network, while the propagation was triggered by placing a marker inside one of the objects. Another useful image processing operation that can be performed via asynchronous propagation - closed curve detection – is shown in Figure 6. In this case the marker was placed in the bottom-right corner of the image (although in practice the triggering would be rather performed at all pixels along the border of the array, to reduce the overall propagation time). The result of the

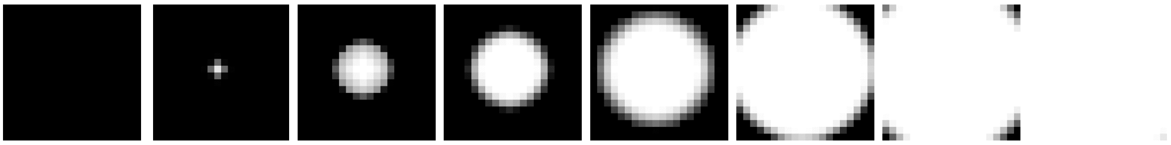

Fig. 4. Trigger-wave propagation in the proposed circuit. State captured at 0.4 ns intervals.

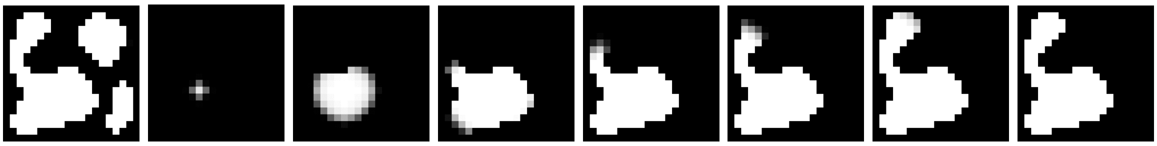

Figure 5. Geodesic Reconstrucion. Input image (left) and state captured at 0.5 ns intervals

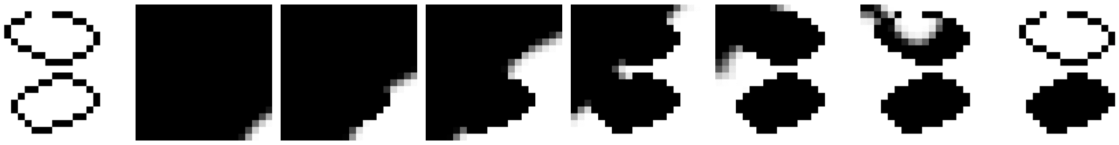

Figure 6. Closed curve detection: input image and state captured at 1 ns intervals.

asynchronous propagation operation can be easily combined with the original image via local logic operations, for example to delete the non-closed curves, or to detect holes.

It is interesting to note, that the boundary of the trigger-wave forms a circular shape, rather than a "diamond" shape, which is obtained in a synchronous system implementing iteratively the equation (1). This can be simply explained by the fact, that the speed of the OR gate in the cell circuit is increased if a greater number of its inputs is in logic state '1', since more than one transistor is then discharging the node *x*. A more detailed insight is of course obtained by considering the magnitudes of the transistor currents and node voltages during the transition period, i.e. analogue circuit analysis, as shown by the SPICE simulation.

The similarity of the network's behaviour to an analogue CNN generating a trigger-wave [16] is apparent. But it has to be emphasised, that unlike many other analogue and digital CNN implementations, the ACLA exhibits a critically important feature - that of a minimal power consumption. At the same time, the propagation speed is very high. From simulation results we obtain that the propagation occurs with the speed of approximately 0.18 ns per pixel, while the total energy expended during propagation is equal to 0.37 pJ per cell. To put this figures in some perspective, consider an array of 128×128 cells used in a real-time computer vision application, to perform 200 trigger-wave propagations per each image frame, at the processing speed of 30 frames/second. The total power consumption contributed by the ACLA to the overall power consumption of the system is only 2.2 nW per pixel, or 36.4 μW per 128×128 array. These figures do not take into account capacitances that would be created by physical layout, but on the other hand are based on a 0.35μm digital CMOS technology, which can be easily replaced today by a lower-scale technology.

4. Conclusions

An asynchronous cellular logic array circuit suitable for binary propagation operations in image processing applications has been described. The proposed method of trigger-wave generation is, in terms of its power efficiency and speed of operation, very close to a limit of what can be achieved in the CMOS technology. This can be contrasted with a CNN-style implementation. Even in the most simplified case [21] the logic functions are still implemented by a CNN using analogue summation, where the circuits corresponding to positive and negative coefficients in the cloning template "compete" with each other, dissipating static power and slowing-down the state evolution. At the same time, analogue circuits require large transistors to ensure robust implementation. In the ACLA circuit minimum-size transistors can be used, resulting in a very compact implementation.

Of course, it can be argued that circuits build according to a more general CNN paradigm are more versatile. Indeed, a standard CNN can be used to solve binary propagation, as well as grey-scale problems, while some other CNNs attempt to implement any Boolean function in a single cell. It is our view, however, that when it comes to engineering practical visual microprocessor systems the practical considerations dictate in favour of solutions that provide a set of useful image processing operators, implemented in the most efficient way. In this paper such a solution, suitable for today's CMOS technology, has been presented. It is expected that the proposed circuit will be used as a coprocessor on a massively parallel SIMD processor array.

References

[1] G. Liñán *et. al*. "Architectural and Basic Circuit Considerations for a Flexible 128x128 Mixed-Signal SIMD Vision Chip", *Analog Integrated Circuits and Sig. Proc.* , vol.33, pp.179–190, 2002

[2] A.Paasio, A.Kananen, K.Halonen and V.Porra, "A QCIF Resolution Binary I/O CNN-UM Chip", in *Journal of VLSI Signal Processing*, vol 23, pp.281-290, 1999.

[3] P.Dudek and P.J.Hicks, "An Analogue SIMD Focal Plane Processor Array", IEEE International Symposium on Circuits and Systems, ISCAS 2001, vol.IV, pp.490-493, May 2001.

[4] M.Ishikawa, K.Ogawa, T.Komuro, I.Ishii, "A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing", Proc. ISSCC'99, TP 12.2, 1999.

[5] F. Paillet, D. Mercier, and T.M.Bernard, "Making the most of 15k lambda2 silicon area for a digital retina", Proc. Conf., Proc. SPIE, Vol. 3410, AFPAEC'98, 1998

[6] J. von Neumann, "A system of 29 states with a general transition rule" in A.W.Burks (Ed.), "Theory of Self-reproducing Automata", Univ. of Illinois, 1966.

[7] S.H.Unger, "A computer oriented to spatial problems", in *Proc. IRE*, vol. 46, pp.1744-1750, 1958

[8] M.J.B.Duff, "Review of the CLIP image processing system", in Proc. National Computer Conference, pp.1055-1060, 1978

[9] L.O.Chua, "CNN-a paradigm for complexity", Int. J. Bifurcation and Chaos, vol.7, August 1997

[10] P.Julián, R.Dogaru and L.O.Chua, "A Piecewise-Linear Simplicial Coupling Cell for CNN Gray-Level Image Processing", in *IEEE Tranactions on on Circuits and Systems – I: Fundamental Theory and Applications*, vol.49, no.7, pp.322-335, July 2002

[11] R.Dogaru, L.O.Chua, "Universal CNN Cells", *International Journal of Bifurcation and Chaos*,Vol.9, pp.1–48, 1999.

[12] J.E.Eklund *et. al*. "VLSI Implementation of a Focal Plane Image Processor – A Realisation of the NSIP Concept", in *IEEE Trans.on VLSI Systems*, vol.4, no.3, pp.322-335, September 1996

[13] T.M.Bernard, "Multi-purpose semi-static shift registers for digital programmable retinas", SPIE Proc. 3965, Photonics West, San Jose, CA, January 24-25, 2000

[14] A.Paasio et. al, "A 32×32 Cellular Test Chip Targeting New Functionalities", IEEE International Symposium on Circuits and Systems, ISCAS 2003, vol III, pp.506-509, 2003.

[15] P.Dudek, "Accuracy and Efficiency of Gray-level Image Processing on VLSI Cellular Processor Arrays", CNNA'04

[16] Cs. Rekeczky and L.O. Chua, "Computing with Front Propagation: Active Contour and Skeleton Models in Continuous-Time CNN", *Journal of VLSI Signal Processing*, 23, pp.373-402, 1999.

[17] I.Szatmari and Cs.Rekeczky, "A Nonlinear Wave Metric and its CNN Implementation for Object Classification", *Journal of VLSI Signal Processing*, vol. 23, pp.437-447, 1999.

[18] J. Poikonen, A.Paasio, "Robustness Analysis of a Physical Multi-Nested CNN Implementation", ECCTD'03, vol III, pp.365-368, 2003

[19] I.Fajfar and F.Bratkovic, "Design of Monotonic Binary-Valued Cellular Neural Networks", Proc. Conf. CNNA'96, pp.321-326

[20] P.Dudek "An Asynchronous Boolean Network for Trigger-wave Image Processing on Fine-grain Massively Parallel Arrays", to be published.

[21] A.Paasio, A.Kananen and K.Halonen, "A compact computational core for image processing", Proc. European Conference on Circuit Theory and Design, ECCTD 2001