

Trigger-Wave Collision Detecting Asynchronous Cellular Logic Array for Fast Image Skeletonization

Przemyslaw Mroszczyk and Piotr Dudek
School of Electrical & Electronic Engineering
The University of Manchester
Manchester, M13 9PL, United Kingdom
przemyslaw.mroszczyk@postgrad.manchester.ac.uk
p.dudek@manchester.ac.uk

Abstract—This paper presents the design of an asynchronous cellular logic array for binary image processing algorithms based on wave propagation/collision in an excitable medium. The array consists of identical logic cells enabling the propagation and detection of wave-front collisions necessary for the object skeletonization. Low power, low area and high processing speed requirements were met by employing the asynchronous dynamic logic approach resulting in a processing time less than 0.45ns/pixel and energy consumption of less than 0.15pJ/pixel. The cell consists of 19 transistors and occupies an area of $7.5 \times 6.3 \mu\text{m}^2$ in 90nm CMOS technology. The proposed array could be used as a coprocessor in pixel-parallel SIMD architectures aiding the fast execution of medium-level image processing algorithms.

I. INTRODUCTION

Massively parallel processor arrays based on the single-instruction multiple data (SIMD) principle are used for efficient execution of low-level image processing algorithms. In particular, solutions based on "processor-per-pixel" architectures enable the design of high-performance, compact, low-power integrated vision systems. In the literature there are many such processor array designs based on analogue, digital and mixed processing elements [1]-[5]. One of the important applications of vision systems is the fast recognition of objects' features, very often aided by a preliminary binary object skeletonization. There are many applications for the skeletonization operation overviewed in [6] such as character recognition, biological cell analysis or fingerprint classification.

The skeleton of a binary image consists of lines and curves creating continuous structure defining the shape and the size of the object. There are a number of different methods for image skeletonization exhibiting different levels of complexity and returning slightly different results. The most common ones are based on thinning and distance transformation [6], [7]. Thinning algorithms operate on binary images and require a large number of templates, sometimes of two pixels radius, which in a hardware realization may require more physical interconnections between cells. In the case of distance

transformation, every processing element has to store numerical information about its position relative to the object's edge which involves integer computation and makes the solution image-size variant. All of these implementations are based on iterative processes similar in execution to a wave propagating across the array where only the wave-front pixels are involved in the processing, whereas the remaining ones can be turned into the idle state [5]. The hardware implementation of this mechanism requires the build of an array providing an appropriate excitable medium for wave propagation [8]. In particular, the wave propagation method can be applied to a variety of morphological operations such as object reconstruction, hole filling, closed curve detection, Voronoi tessellation, skeletonization and distance transformation [9], [10].

This paper presents the VLSI design of the asynchronous logic array constituting the excitable medium for the binary image processing algorithms based on wave propagation. To obtain a skeleton of an object, waves are triggered from the background of the image and they propagate to the inside of the object. It has been observed that the shape of the resulting collision line (where two opposite wave-fronts meet) is close to the skeleton of the object [9]. Rather than attempting to inhibit the propagation at the point where two wave-fronts meet (as is typically done in the iterative thinning methods) we propose to employ a separate collision detecting mechanism. This simplifies the design of the cell and enables a purely asynchronous operation. The idea of the propagation gate presented in [10] has been adopted and equipped with the wave-front collision-detecting stage. Section II of this paper presents the circuit design, Section III presents simulation results, Section IV discusses implementation issues and Section V concludes the paper.

II. CIRCUIT DESIGN

The Asynchronous Cellular Logic Array (ACLA) is a homogenous structure which consists of identical blocks (cells) performing the same logic functions [10]. The array is organised in a way that every cell receives output signals from its neighbours. Considering the limitations of the VLSI

hardware implementation, a 4-neighbourhood with one pixel radius and a rectangular grid were chosen for the practical realisation of the circuit.

A. Propagation and wave-front collision-detecting circuits

The logic function associated with the propagation gate can be verbally expressed in the following way: "If any of my neighbours is active, I become active". Graphically, the group of the active cells grows with time and, in the end, encompasses the whole propagation space. The idea of the expansion of the propagation wave is presented in Fig. 1a. and the logic circuit of a single cell is shown in Fig. 1b.

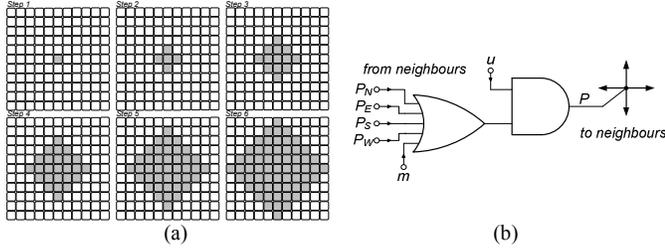


Figure 1. Wave propagation circuit: a) the expansion of the propagation wave, b) the conceptual schematic diagram of the propagation OR-AND gate proposed in [10].

To initiate the propagation, an additional signal m (marker) is used to drive the selected group of cells. The signal u can enable or disable the propagation gate individually, thus it is used for defining the propagation space. In the case of skeletonization, signal u can be permanently connected to the high state which converts the AND gate into a buffer. Both signals are generated individually for every cell.

The proposed wave-front collision-detecting mechanism can be verbally described in the following way: "If I have just become active and all my neighbours are active, then I belong to the collision line". It is assumed that the resulting collision line should be continuous and of (at most) two pixels width, depending on whether there is an even or odd number of pixels in between the wave-fronts. The exemplar case showing the collision of two waves propagating from the opposite directions is presented in Fig. 2a (simulation snapshots showing colliding waves are shown in Fig. 7c). The neighbourhood condition can be detected by the AND gate shown in Fig. 2b.

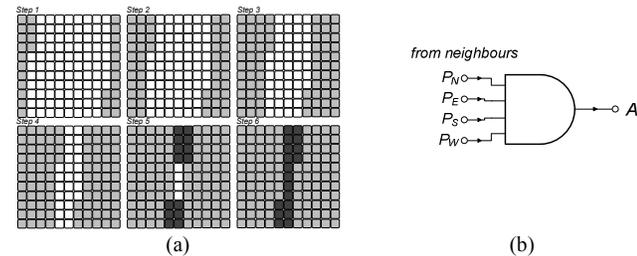


Figure 2. Collision detection circuit: a) skeleton line resulting from collision of two wave-fronts, b) AND gate indicating the collision condition.

The aforementioned verbal description of the collision-detecting mechanism implicitly assumes the existence of a certain time interval (the "I have just become active" time slot) when the collision-detecting module ought to be enabled. The

conceptual schematic diagram of the complete ACLA cell circuit with an additional delay gate and a D latch saving the collision detection result is shown in Fig. 3. The corresponding diagram illustrating timing relations between signals is presented in Fig. 4.

Once the propagation gate P_1 receives the signal from any of its neighbours or via marker m , it responds after the propagation time T_{PD} by setting its output (P_1) to the high state. This moment denotes the beginning of the time slot for this cell. The AND gate A_1 "checks" the states of all the neighbours and returns the computed result A_1 after time T_{AD} . Delay buffer B_1 generates appropriate delay T_{DD} after which the signal A_1 is latched and the time slot is terminated. For correct operation it must be ensured that: $T_{AD} < T_{DD} < T_{PD}$ (this can be achieved through a proper design and using controllable delay gates). To simplify the analysis we assume an ideal D latch with zero setup and hold time.

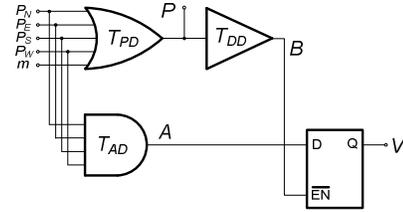


Figure 3. Schematic diagram of the proposed ACLA cell.

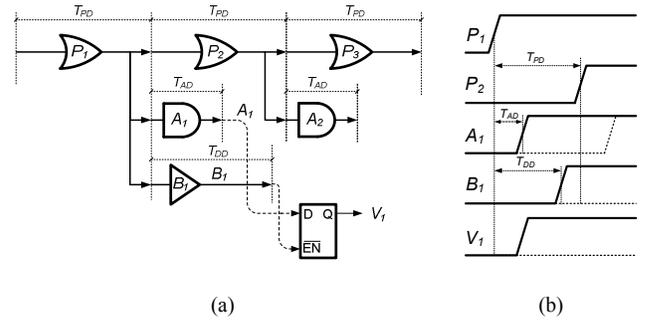


Figure 4. The timing analysis of the proposed ACLA cell: a) schematic of a 1D array illustrating timing relations, b) timing diagram (dotted lines of A_1 and V_1 show diagrams for no-collision case).

B. Transistor level implementation

Following the idea used in [10] the dynamic logic approach is employed to realise the functionality of the structure from Fig. 3. The transistor level schematic diagrams of the constituent gates of the proposed collision-detecting cell are shown in Fig. 5. The circuit implements the logic of Fig. 3 under the assumption that the transitions of signals P_N , P_E , P_S , P_W are always from '0' to '1' (which is the case here). The circuit requires the initialization cycle before every evaluation cycle. To initialize the gate, the global signal *discharge* is set to V_{DD} discharging the parasitic capacitances of nodes P and V through M_8 and M_{19} respectively. Simultaneously, the capacitances of *NOR* and *NAND* nodes are charged to V_{DD} through M_6 and M_{12} . In order to prevent these charges from leaking, transistors M_6 and M_{12} work as weak keepers assuring the high logic state of these nodes as long as all the inputs remain inactive. If any of the input signals turns to the high

state, node *NOR* discharges turning on M_7 and setting the output P to V_{DD} . If all the inputs turn to the high state the node *NAND* can be discharged depending on the state of the signal *EN* (*enable*). This signal is generated by the inverting stage with delay time controlled by the analogue input *delay*. Once the ACLA gate is discharged it remains in that state until the next initialization.

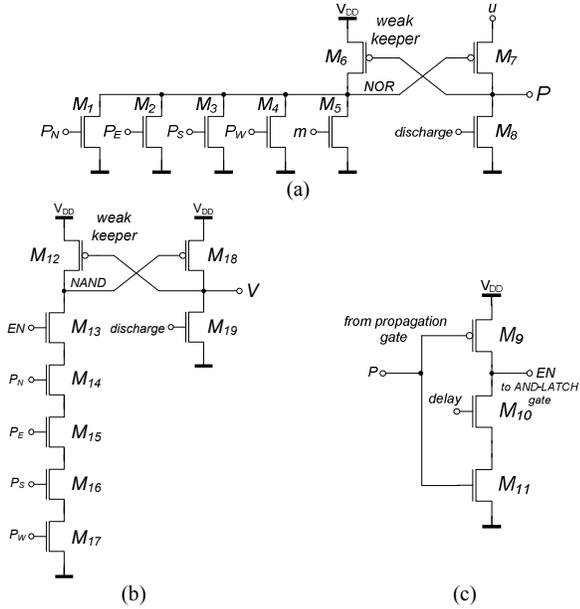


Figure 5. Schematic diagrams of a) propagation gate, b) AND-LATCH gate and c) adjustable inverting delay gate.

III. SIMULATION RESULTS

Simulation results demonstrating the operation of the proposed array consisting of 128×128 ACLA cells are shown in Fig. 6 and Fig. 7. For simulations HSPICE program and level 54 MOS transistor models from a standard 90nm CMOS technology were used. The skeletons of the input images were also computed by means of a Matlab built-in procedure (*bwmorph*) and a software implementation of the iterative synchronous collision-detecting algorithm (Fig. 6). The difference between the two software results (Fig. 6b and c) can be put down to the simplicity of the second approach whereas the difference between software and hardware implementations (Fig. 6c and d) is the result of the asynchronous propagation in the circuit affecting the wave-front shape (this will be discussed in section IV.B).

IV. IMPLEMENTATION ISSUES

A. Power consumption and leakage currents

The main contributors to the overall power consumption are the discharge cycle, when all the nodes either discharge to GND through M_8 and M_{19} or precharge to V_{DD} through the weak keeping transistors M_6 and M_{12} , and the propagation (evaluation) cycle, when the voltages of nodes *NOR* and *NAND* drop down to GND whereas the corresponding weak keepers are still turned on. The post layout simulations of the arrays supplied from a 1V source showed that the average propagation time per pixel is equal to about 0.45ns (SS corner)

and the total energy consumed by a pixel per one discharge and propagation cycle is 0.13pJ for non-collision pixel and 0.15pJ when a collision was detected (FF corner). Arrays of different sizes are characterized in Table 1.

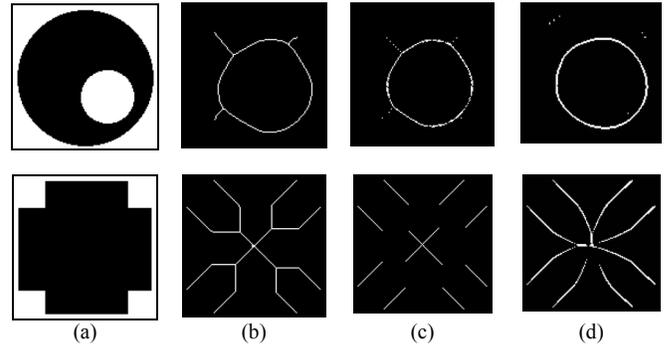


Figure 6. The comparison of the results: a) the input image, b) Matlab *bwmorph*, c) collision-detecting software algorithm, d) proposed circuit.

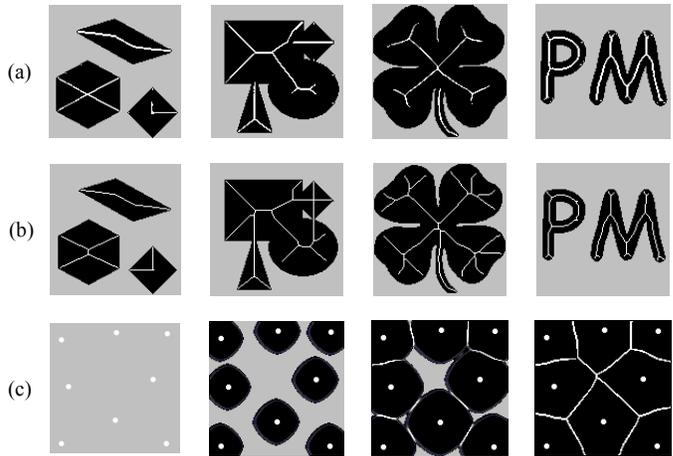


Figure 7. Skeletonization results: a) proposed circuit, b) Matlab *bwmorph*, c) Vronoi tessellation, input image and propagation snapshots after 7ns, 10ns and 15ns (final result).

TABLE I. TIME AND POWER PERFORMANCE OF ACLA ARRAYS.

Array Size	Processing Time ¹⁾	Average supply current ²⁾	Energy/Frame ²⁾
32×32	1 ns	5.72mA	132pJ
64×64	18.06ns	22.62mA	520pJ
128×128	33.2ns	88.35mA	2.1nJ

1) the processing time includes 3ns precharge cycle (SS corner)

2) FF corner

After the discharge cycle, the leakage currents of M_7 and M_{18} cause the gradual increase of the voltage on outputs P and V until the leakage currents of $M_7 - M_8$ and $M_{18} - M_{19}$ equalize. This effect may erase the initial logic values written to the nodes and also may trigger a spurious propagation in the array. To protect the circuit from this effect, transistors M_7 and M_{18} were implemented as low-leakage devices. Additionally, to increase the leakage currents of transistors M_8 and M_{19} after the discharge cycle, it is proposed not to turn the *discharge* line to GND but to apply a low voltage (150mV) to the gates of these transistors to slightly increase their drain currents. With this solution, the current consumed by a single cell after

propagation is $1.2\mu\text{A}$ for a pixel which did not detect the collision and $2.2\mu\text{A}$ when the collision was detected (FF corner). The current consumed by a single cell can be reduced to about 100nA (FF corner) by setting the propagation space to the full image size and triggering the propagation only from one arbitrarily selected pixel to "discharge" the array without collisions and keeping the *discharge* line at 0V . This state of the array can be considered as the low-power idle mode, applied when the array is not used.

B. Propagation issues

In Section II it is assumed that the time slot T_{PD} is constant whereas in a practical realization this condition may not be met due to several reasons. Firstly, the structure of the AND-LATCH gate is not symmetric because input transistors $M_{14} - M_{17}$ exhibit different gate capacitances due to different bulk-source voltages (this is a systematic error in addition to any mismatch effects). The second reason is shortening of the propagation time T_{PD} when the particular cell is triggered from more than one neighbour simultaneously; in such case the capacitance of node *NOR* discharges faster. These effects affect the shape of the propagation wave-front and the resulting skeleton. A further development of the propagation gate with tunable T_{PD} time is presented in Fig. 8. The additional transistor M_{14} can be used to limit the total current discharging node *NOR* when more than one input is active. The resulting wave-front is then in the shape of a diamond. When M_{14} is fully turned on, the propagation time is not uniform, and the shape becomes more circular and finally a square depending on voltage V_{MODE2} controlling the gates of $M_1 - M_5$. In other words the shape of the wave-front denotes the metric in which the skeleton (or any other algorithm) is developed.

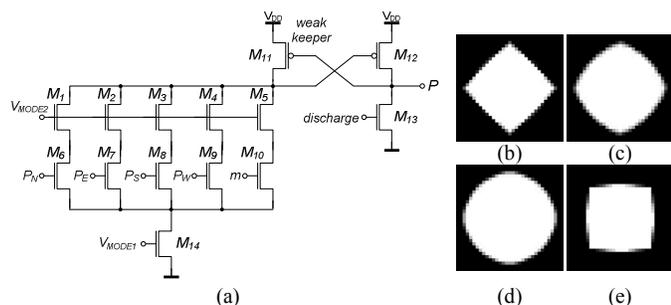


Figure 8. a) Schematic diagram of the gate with propagation metric control and obtained propagation wave shapes: b) diamond ($V_{MODE1}=0.4\text{V}$, $V_{MODE2}=1\text{V}$), c) diamond-circular ($V_{MODE1}=V_{MODE2}=1\text{V}$), d) circular ($V_{MODE1}=1\text{V}$, $V_{MODE2}=0.55\text{V}$), e) square ($V_{MODE1}=1\text{V}$, $V_{MODE2}=0.4\text{V}$).

C. Mismatch Issues

A simple way of reducing the influence of mismatch on the propagation time variability is to enlarge critical transistors contributing to time delays T_{PD} and T_{DD} . Fig. 9 shows the results of Monte Carlo mismatch analysis of two arrays developing the skeleton of a rectangle. Proper scaling of the transistors in the propagation and delay gates is required to ensure correct operation. In order to further reduce the mismatch errors, the structure of the delay gate from Fig. 5c should be chosen rather than a current-starved inverter [11]. The variability of a time delay of the proposed structure is

almost 30% less while both circuits occupy the same area. In practice, for higher image resolutions the propagation wave-fronts will be distorted by the global parameter variations limiting the maximum size of the array.

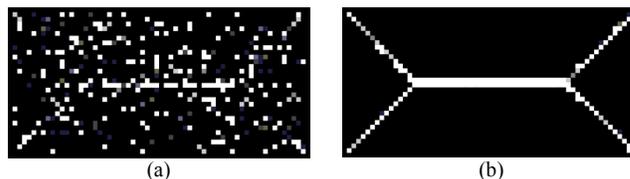


Figure 9. Monte Carlo mismatch simulation: a) using minimum-size transistors the result is highly erroneous, b) enlarging the critical transistors produces a correct result.

V. CONCLUSIONS

In this paper, the collision-detecting ACLA cell has been presented. The cell array implements asynchronous binary wave propagation while detecting the collisions of different wave-fronts. In particular, the skeletonization can be executed within 0.45ns/pixel consuming less than 0.15pJ/pixel energy. The cell was designed in UMC 90nm CMOS technology, it consists of only 19 transistors and occupies an area of $7.5 \times 6.3\mu\text{m}^2$. The proposed circuit will be of use in the design of pixel-parallel cellular processor array devices.

REFERENCES

- [1] Eklund J.E., Svensson Ch., Astrom A., "VLSI Implementation of Focal Plane Image Processor - A Realization of the Near-Sensor Image Processor", IEEE Transactions on Very Large Scale of Integration (VLSI) Systems, vol. 4, no. 3, Sep. 1996
- [2] Poikonen J., Laiho M., Paasio A., "MIPA4k: A 64×64 Cell Mixed-mode Image Processor Array", IEEE International Symposium on Circuits and Systems ISCAS 2009, May 2009
- [3] Rodriguez-Vazquez A. et al. "ACE16k: The Third Generation of Mixed-Signal SIMD-CNN chips towards VSoCs", IEEE Transactions on Circuits and Systems I, vol. 51, issue 5, May 2004
- [4] Ishikawa M., Ogawa K. Komuro T., Ishii I., "A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing", IEEE International Solid-State Circuits Conference ISSCC 1999
- [5] Lopich A., Dudek P., "Hardware Implementation of Skeletonization Algorithm for Parallel Asynchronous Image Processing", Journal of Signal Processing Systems, Springer, Volume 56, Number 1, pp. 91-103, Jul. 2009
- [6] Lam L., Lee S.W, Suen Ch.Y., "Thinning Methodologies - A Comprehensive Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 9, Sep. 1992
- [7] Davies E. R., "Machine Vision: Theory, Algorithms, Practicalities", Cambridge University Press, 1990
- [8] Krinsky V., Biktashev V., Efimov N., "Autowaves Principles for Parallel Image Processing", Physica D, vol. 49, pp. 247-253, 1991
- [9] Rekeczky C., Chua L. O., "Computing with Front Propagation: Active Contour and Skeleton Models in Continuous Time CNN", Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, Volume 23, Numbers 2-3, pp. 373-402, 1999
- [10] Dudek P., "An Asynchronous Cellular Logic Network for Trigger-Wave Image Processing on Fine-Grain Massively Parallel Arrays", IEEE Transactions on Circuits and Systems - II, vol. 53, no.5, pp. 354-358, May 2006
- [11] Dudek P., Szczepanski S., Hatfield J. V., "A High-Resolution CMOS Time-to-Digital Converter Utilizing a Vernier Delay Line", IEEE Transactions on Solid-State Circuits, vol. 35, no. 2, Feb. 2000