Fast and Hardware-Efficient Systolic Architectures for Binary Morphological Processing

E. N. Malamas[†], A. G. Malamos[†], and T. A. Varvarigou[††]
[†]Electronics Laboratory, Dept. of Electronics and Computer Eng.,
Technical University of Crete,
Gr-731 00 Chania,
GREECE

[††]Telecomunications Laboratory, Dept. of Electrical and Computer Eng.,
National Technical University of Athens,
Eroon Polutechniou 9, GR-157 73 Zografou, Athens,
GREECE.

Abstract- In this paper we present novel systolic architectures for the fast execution of common morphological operations, that is dilation, erosion, closing, and opening. Their novelty stems from the fact that the same unit, the combined Erosion-Dilation Architecture (EDA), is used to perform either dilation, or erosion, or both of them in parallel (depending on control signals). The proposed architectures show a major advantage on using reduced resources for storing the structuring element (SE), lead to full resource utilization, and provide high processing rates. We emphasize on 1-dim structuring elements and present an improved architecture, that performs dilation and erosion in half the time compared to other architectures, using a workload partitioning technique. Furthermore, the amenability of the EDA to VLSI implementation is exemplified by a processor that performs binary morphological operations with 1x3 structuring sets. Finally, we show that the modularity of the proposed architectures allows the direct extension to 2-dim morphology.

Key-Words: - mathematical morphology, structuring element, systolic architectures, VLSI implementation.

1. Introduction

Mathematical morphology (MM) [1] is a relatively new image processing technique based on shape. It is strongly related to the theory of sets and the Minkwoski algebra [2] and provides an effective tool for geometrical analysis of images, since conventional signal processing techniques are unable to model shape as efficiently as MM does. Morphological filters [3] are extensively used in several experimental and industrial applications, in early stages of processing images, as low level neighborhood operators [4]. The generality of the theory underlying MM, allows the use of multidimentional filters in gray-scale, time-varying, as well as colored images. The two-dimentional case refers to binary images, where the intensity of each pixel is expressed by one bit.

The importance of MM in the image processing area is immense and there has been a continuous effort of the scientific community to implement fast and reliable morphological systems. The rich set of the Minkwoski algebra properties, have made feasible the creation of very effective systolic architectures such as the NPP [5], the systolic structure in [6], the PAPRICA system in [7], and pipelined systems such as [8], [9]. Moreover, in [4] and [10] a variety of systems intended to the execution of morphological operations are presented.

In this paper we propose a novel architecture that effectively uses some important properties of MM, in order to implement binary morphological operations fast and efficiently in terms of hardware use. We prove that the resources of the proposed architecture can be fully utilized, either during the intensive execution of simple morphological operations (dilation, erosion), or during complex morphological operations (opening, closing). The efficient use of hardware is supported by the locality of calculations, the regularity of the processing elements used, and the simplicity of the communication among them. The major advantage is that the demands on memory for the storage of the components of the structuring element (SE), are reduced to half for closing and opening, compared to other similar architectures. A technique for partitioning the image workload, that leads to the reduction of processing time for dilation and erosion to half, leads to an improved architecture.

In the second section of the paper we present the basics of the MM theory and some properties that are used in subsequent sections. The third section is devoted to the analysis of the proposed systolic structures. In the fourth section we present an analysis on the hardware complexity of the architectures and an improved version with respect to the time performance, as well as a VLSI implementation. The fifth section is devoted to the extention of the proposed arrays to the domain of 2–dim SEs. Finally, some concluding remarks are presented in the sixth section.

2. Binary Mathematical Morphology Basics

The idea of MM is based on the extraction of structural information from image signals. This extraction is done through two basic steps: 1) the image signal, denoted by A from this point forward, is transformed into another signal by an image operator, called SE (denoted by B). The type of this transformation $\Lambda(A)$, is strongly depended on the characteristics of the SE. 2) A measurement $\mu[\Lambda(A)]$, with physical meaning such as size, shape, connectivity of the image A, is extracted from this transformation.

There are two basic transformations in MM, namely dilation and erosion, that will be presented in the domain of binary image signals. Assuming that A and B are two subsets of the n–dim Euclidean space E^n, where A is the image and B is the structuring element, dilation and erosion are defined as follows

$$A \oplus B = \{c \in E^n \mid c = a + b,\ \forall a \in A,\ b \in B\} \text{ (Dilation)}, \quad (1)$$

$$A \ominus B = \{x \in E^n \mid x + b \in A,\ \forall\ b \in B\} \text{ (Erosion)}, \quad (2)$$

where a, b, c, and x in the definitions are n–dim tuples of element coordinates. Formulation of definitions (1) and (2) into matrix format, leads to relations that employ n–dim binary matrices that are more appropriate for implementation on hardware. Thus, dilation and erosion become

$$D_{i_1,\ldots,i_n} = \sum_{k_1,\ldots,k_n} \left(b_{k_1,\ldots,k_n} \cdot a_{i_1-k_1,\ldots,i_n-k_n} \right)$$

$$E_{i_1,\ldots,i_n} = \prod_{k_1,\ldots,k_n} \left(b^c_{k_1,\ldots,k_n} + a_{i_1+k_1,\ldots,i_n+k_n} \right) \text{ respectively,}$$

where $D_{i_1,\ldots,i_n}$, $E_{i_1,\ldots,i_n}$ are the elements of the binary matrices representing dilation and erosion and $a_{i_1-k_1,\ldots,i_n-k_n}$, $a_{i_1+k_1,\ldots,i_n+k_n}$ are the shifted versions of the image pixel $a_{i_1,\ldots,i_n}$. The amount of this shift is defined by the element $b_{k_1,\ldots,k_n}$ of the SE, whose logical complement is $b^c_{k_1,\ldots,k_n}$. The symbols Σ and + are used for logical ORing between image components, while the symbols Π and $\cdot$ are used for logical ANDing. These matrix format relations constitute generalized n–dim convolutions [11] and their reduction to 1–dim signals leads to the following:

$$D_i = \sum_{k=0}^{N-1} \left(b_k \cdot a_{i-k} \right), \quad (3)$$

$$E_i = \prod_{k=0}^{N-1} \left(b^c_k + a_{i+k} \right), \quad (4)$$

where N is the length of the SE. Some important properties of the Minkowski algebra, employing dilation and erosion are as follows:

i) Parallel decomposition for dilation and erosion

$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C). \quad (5)$$

$$A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C). \quad (6)$$

ii) Duality property between dilation and erosion

$$(A \ominus B)^c = A^c \oplus \breve{B}, \quad (7)$$

where c denotes the complement and $\breve{B}$ is defined to be $\breve{B} = \{x \mid \forall b \in B,\ x = -b\}$ and called the geometrical negative of B or reflection [12].

iii) Shift invariance of dilation and erosion

$$A_t \oplus B = A \oplus B_t = (A \oplus B)_t \quad (8)$$

$$A_t \ominus B = A \ominus B_{-t} = (A \ominus B)_{-t} \quad (9)$$

where t is the translation vector. Complex morphological operations are most often used in image processing and computer vision applications [12, 13]. The most usual of them are the opening and closing, whose definitions are quoted below

$$A, B \subseteq E^n,\ A \circ B = \cup \{B + x : B + x \in A\} = (A \ominus B) \oplus B \text{ (Opening)}, \quad (10)$$

$$A, B \subseteq E^n,\ A \bullet B = \cap \{\breve{B} + x : \breve{B} + x \cap A \neq \varnothing\} = (A \oplus B) \ominus B \text{ (Closing)}, \quad (11)$$

respectively. Opening and closing are decomposed into successive dilations and erosions which are alternating, depending on the operation. Examples of the application of the four basic operations can be found in Fig. 1. Generally speaking, dilation creates a possibly translated (if the origin does not belong to the SE) superset of the original image, while erosion creates a subset. Furthermore, opening produces a subset of the original image, where isthmuses are removed and curves are smoothed. In contrast, closing produces a superset of the image, where holes are filled in and the connectivity of the image is increased [14].

3. Systolic Architectures For Morphological Operations

3.1 Review Of Morphological Systems

The common practice in the design of morphological systems is either to use dedicated units for each operation [5, 6] or use one unit as the core for the implementation of more than one operation [15]. We have drawn in Fig. 2 architectures, that follow these two most popular methodologies. The system in Fig. 2(a) uses two separate units for dilation and erosion [6]. It computes such operations as closing and opening by cascading the two units in a pipelined fashion. When it computes only dilation, then the erosion unit remains idle and vice versa. Furthermore, the SE must be stored in both units, which causes even more hardware overheads.

On the other hand, the architecture in Fig. 2(b) stems its function from the duality property that holds between dilation and erosion, therefore using the same unit for both operations [15]. This however, creates the need for a memory unit to store the intermediate results, in case cascaded operations are performed, e.g. closing. Time is also wasted, since the execution of the second operation cannot start before the complete execution of the first. The two structures in Fig. 2 compose a common trade–off between hardware complexity and execution time in the context of implementation of morphological operations.

3.2 The Combined Erosion–Dilation Unit

Our approach stands in between the two methodologies presented in 3.1 and includes the combined Erosion–Dilation Architecture (EDA), which is shown in Fig. 3. The basic novelty of the EDA is the capability of executing all basic morphological operations on one unit (as opposed to the system in Fig. 2(a)), without wasting hardware or introducing delays. Note also that the SE is stored only once and is shared by all system modules. As shown in Fig. 3, the proposed architecture consists of two basic units: a control unit, denoted as *control* and a core EDA, denoted as $A \oplus B / A \ominus B$. The remaining hardware consists of three multiplexers, which support the operation of the overall system.

3.2.1 The Control Unit And Relative Logic

The control unit is used to configure the architecture in order to execute properly the requested operation. The two control inputs cover the four possible operations and an arbitrary assignment of binary values to the control signals, depending on the morphological operation selected, is as follows: Dilation→ {control1, control2}= {0, 0}, Erosion→ {control1, control2}= {0, 1}, Opening→ {control1, control2}= {1, 0}, and Closing→ {control1, control2}= {1, 1}. Using ordinary digital circuit design techniques, we may construct the logic circuit of the control unit. The association of the values of the selection signals to the four morphological operations that the system is capable of performing is as follows: Dilation→ {sel_1, sel_2, sel_out}= {0, x, 1}, Erosion→ {sel_1, sel_2, sel_out}= {x, 0, 0}, Opening→ {sel_1, sel_2, sel_out}= {1, 0, 1}, and Closing→ {sel_1, sel_2, sel_out}= {0, 1, 0}, where x stands for the "don''t care condition". The multiplexers shown can be implemented with transmission gates [16].

3.2.2 The Internal Structure Of The Core EDA

The core EDA is the component that implements all four of the morphological filters. We focus on the case of 1–dim SEs, in order to present the functionality of this unit. The extension of our study to 2–dim SEs is straightforward and is presented in Section 5.

For the construction of 1–dim EDAs, we use the duality property between erosion and dilation (relation (7)), as it is applied in the case of 1–dim SEs. As can be readily seen, when a 1–dim SE $B=\{b_0, b_1, ..., b_{N-1}\}$ is used for dilation, its components are used in the order $b_0, b_1, ..., b_{N-1}$, as dictated by (3). On the other hand, when erosion is performed, the components are used in the reversed order, that is $b_{N-1}, b_{N-2}, ..., b_0$, since for 1–dim SEs $\breve{B}$ results from rotating the SE B by 180^o around its origin. These properties allow us to store the components of the SE in the EDA of Fig. 3 only once. This is in contrast to systems such as [5, 6], where complex morphological operations require the duplication (and pipelining) of the architectures for dilation and erosion and correspondingly of the memory for the SE.

In Fig. 4(b) we show the internal structure of a cell of the EDA, called Erosion–Dilation Unit (EDU). The Dilation Segment denoted as DS={rd1, rd2, rd3, rd4, ANDd, Ord, rse} is devoted to the execution of dilation as dictated by (3) and the Erosion Segment denoted as ES={re1, re2, re3, re4, NOTe, ANDe, Ore, rse} executes erosion, according to (4). The two parts retrieve the appropriate component of the SE from the common memory storage unit, the latch *rse*, as shown in Fig. 4(b). When erosion (dilation) is executed, only the ES (DS) is needed, but when closing (opening) is

executed, image data are inserted into the DS (ES) and fed back from the output to the input of the ES (DS). A schematic representation of the core EDA performing the above operations, is depicted in Fig. 4(a) for the generic 1xN SE B={b_0, b_1, ..., b_{N-1}}. An exemplary system performing opening for 1x2 SEs is described in detail in [17]

3.2.3 Timing Analysis

The latency of the EDA for an erosion and a dilation operation is proved to be $T_l^e = 2(N+1)T_c$ and $T_l^d = (N+3)T_c$ respectively (T_c is the cycle time such that $T_c = 2T_{pd}^g + T_{pd}^l$, where T_{pd}^g is the propagation delay of a single gate and T_{pd}^l is the propagation delay of a latch). The rationale underlying the difference in the two latencies, is that the erosion is a shrinking operation and provides the first valid data later than the dilation operation does [5].

The latency for opening and closing, is proved to be $T_l^o = T_l^c = 3(N+1)T_c$, which is not $T_l^e + T_l^d$. This deviation stems from the fact that the output of the ES is connected to the input of the DS, causing an overlap of two cycles between the latencies of the erosion and the dilation. The delay breakdown presented above is depicted in Fig. 5, as a function of the size of the 1–dim SE.

4. Implementation Analysis

4.1 Hardware Complexity

The multiplexing circuitry accompaning the core EDA, is negligible compared to the hardware used for the implementation of the individual EDU cells. Indeed, one such multiplexer consists of two transmission gates, that is four transistors [16].

Using the analysis in [18, 5], we build a cost function for the binary case and compare the two architectures in terms of hardware savings. Note that the latches used to store the components of the SE were exempted from the analysis in [5], although they are necessary in a real system, in order to avoid further delays stemming from distant memory accesses on the components of the SE. A complete cost function for both units (dilation and erosion) in [5] should be 105N logic gates, where N is the size of the SE. On the other hand, the EDA cost function is 95N, the 9.5% saving in logic gates coming from the single storage facility provided. We have plotted the two functions in Fig. 6 for several sizes of the SE.

4.2 Performance Improvement

Although the proposed architecture has a pipeling period α=1 and image data can be processed one at a time, the resources of each cell are not utilised 100%, in case dilation or erosion is executed. For example, if dilation is performed the ES is idle. To overcome this drawback, we use an idea that is based on splitting the processing load (processed image) into two parts and assigning them to both the ES and the DS of each cell. This can be done by exploiting the properties (5), (7), (8), and (9) as follows. Dilation: Assuming that the dimentions of A are [A]=M_1xM_2, then A=$(A_1)\cup(A_2)_t$, where t=[(M_1/2)–1, 0]. Thus

$$A\oplus B=(A_1\oplus B)\cup((A_2)_t\oplus B)=(A_1^c\ominus \breve{B})^c\cup(A_2\oplus B)_t$$
$$=D_1\cup D_2. \quad (12)$$

Operation D_1 is assigned to the ES and operation D_2 to the DS. Note that additional steps are required, namely complementation of A_1 and $A_1^c\ominus \breve{B}$. Although this fact increases latency by two cycles, the total processing time is reduced to half, with respect to the processing time required if only the DS of each cell is used. Erosion: Similarly with dilation, erosion becomes (for $A^c=A_{1c}\cup(A_{2c})_t$, where A^c is the complementary set of A)

$$A^c\oplus \breve{B}=(A_{1c}\oplus \breve{B})\cup((A_{2c})_t\oplus \breve{B})\Rightarrow$$
$$A\ominus B=(A^c\oplus \breve{B})^c=[(A_{1c}\oplus \breve{B})\cup((A_{2c})_t\oplus \breve{B})]^c$$
$$=[(A_{1C}^c\ominus B)^c\cup(A_{2c}\oplus \breve{B})_t]^c=[E_1\cup E_2]^c. \quad (13)$$

Similar remarks hold for the erosion as for dilation, except the fact that the SE components should be stored in reversed order, that is {b_{N-1}, …, b_1, b_0}. Note that both operations make use of the input frame buffer, which must support two concurrent accesses in order to store the image data and process them in a timely fashion.

A schematic conception of the improved version of the core EDA is shown in Fig. 7. Note the changes in the multiplexing circuitry and concomitant changes in the control unit, caused by the additional steps required for dilation and erosion. The modified selection signals are associated to the morphological operations as follows: Dilation→{sel_1, sel_2, sel_out}={0, 00, 0}, Erosion→{sel_1, sel_2, sel_out}={0, 00, 1}, Opening→{sel_1, sel_2, sel_out}={1, 01, 1}, Closing→{sel_1, sel_2, sel_out}={1, 10, 1}. The additional hardware complexity of the improved EDA is proved to be 95N+50. The additional hardware is overcome for N=5 length of the SE, in relation to [5]. However, the superfluous logic gates in the budget of the improved EDA are considered a

negligible trade–off, for the reduced delay for dilation/erosion execution.

Note that there are two independent data streams fed in and out of the core EDA, in case dilation or erosion is performed. Using two FIFO structures for both the input and the output buffers, is a simple way to support these streams. Each FIFO may store the appropriate portion of the image, i.e. the A_1 and A_2 half–portions of the input image A and the D_1 (E_1) and D_2 (E_2) half–portions of the dilated (eroded) image respectively. Provision for the synchronization between the two streams may also be required, due to the different latencies between the dilation and the erosion operation. A comparative analysis between the performances of the EDA presented in 3.2 and the improved EDA is shown in Table 1. The comparison is done in terms of the length N of the SE and the image dimentions $[A]=M_1xM_2$. It is expected that this improved architecture, is suitable for fast morphological processing of image signals that must be stored before being processed, such as static images or compressed (mpeg) image streams.

TABLE 1: COMPARISON BETWEEN THE TWO ARCHITECTURES

SYSTEM / OPERATION	EDA		IMPROVED EDA	
	L	P.T.	L	P.T
D	N+3	[A]	N+5	[A]/2
E	2N+2	[A]	2N+4	[A]/2
O&C	3N+3	2[A]	3N+8	2[A]

L: Latency in number of cycles
P.T.: Processing Time in number of cycles

4.3 An Example Implementation

We exemplify the implementation of the EDA architecture presented in 3.2, by a VLSI system that is intended for N=3 SEs. The processor is capable of executing all the four morphological operations, depending on the values of the control signals *control1* and *control2*. We used a design methodology based on VHDL and synthesis and the processor was handcrafted using the standard cell library of the Alliance v3 CAD tool. The technology used was a typical CMOS 1μm supported by the tool. The final design was simulated and tested for its proper operation. The implementation details are summarized in Table 2.

TABLE 2: IMPLEMENTATION SUMMARY

Tool	Alliance v3
Technology	1μm CMOS Standard Cell VHDL
Structuring Element	1x3
Total Area (mm^2)	0.28
Cycle Time (ns)	3

5. Extension To 2–dim SEs

The locality and modularity of the proposed EDU provides the ability to extend the ideas of the previous sections to 2–dim SEs. We can use the properties (5) and (6) to decompose the 2–dim morphological operations into 1–dim, in order to process them in parallel and thus reduce the overall processing time. This set–theoretic decomposition is very efficient in terms of computational speed, as opposed to other decompositions of the SEs [7, 19, 20]. Consider a typical binary 3x3 SE $B=B_1\cup(B_2)_{t1}\cup(B_3)_{t2}$, each one of the composites B_i being 1–dim, that is 1x3 and $t_1=[1, 0]$ and $t_2=[2, 0]$ are the translating vectors. Then the following may apply:

$$A\oplus B=A\oplus(B_1\cup(B_2)_{t1}\cup(B_3)_{t2})=$$
$$(A\oplus B_1)\cup(A_{t1}\oplus B_2)\cup(A_{t2}\oplus B_3) \text{ (Dilation).} \quad (14)$$

$$A\ominus B=A\ominus(B_1\cup(B_2)_{t1}\cup(B_3)_{t2})=$$
$$(A\ominus B_1)\cap(A_{-t1}\ominus B_2)\cap(A_{-t2}\ominus B_3) \text{ (Erosion).} \quad (15)$$

$$A\circ B=(A\ominus B)\oplus B=(A\ominus B)\oplus(B_1\cup(B_2)_{t1}\cup(B_3)_{t2})=$$
$$((A\ominus B)\oplus B_1))\cup((A\ominus B)_{t1}\oplus B_2))\cup((A\ominus B)_{t2}\oplus B_3))$$
$$\text{(Opening).} \quad (16)$$

$$A\bullet B=(A\oplus B)\ominus B=(A\oplus B)\ominus(B_1\cup(B_2)_{t1}\cup(B_3)_{t2})=$$
$$((A\oplus B)\ominus B_1))\cap((A\oplus B)_{-t1}\ominus A_2))\cap$$
$$((A\oplus B)_{-t2}\ominus B_3)) \quad \text{(Closing).} \quad (17)$$

In Fig. 8 we show the proposed architectures for dilation and opening. The other two are similarly implemented. Note that the two blocks denoted as $\cap$ and $\cup$ in Fig. 8, are networks of appropriately connected AND and OR logic gates respectively. In case the 2–dim SE is decomposed into a small number of 1–dim SEs, then the $\cap$ ($\cup$) block is reduced to a single AND (OR) gate with small fan in, that connects all core EDAs together. However, if the number of the parallel EDAs is large, then the logic gates consisting the blocks $\cap$ and $\cup$ should be pipelined, in order to reduce the effective cycle time. Moreover, the partial control units of the EDAs can be reduced to one control unit, that supervises the overall architecture, since the partial core EDAs perform identical operations.

6. Conclusion

We introduced some new systolic architectures devoted to the execution of the four morphological operations for binary image signals. They were proved to be simple, modular, with reduced hardware requirements and they provided high processing rates. A method for reducing the processing time by 50% by fully utilising the hardware resources, was presented in the case of dilation and erosion execution. The analysis focused on 1–dim SEs and an implementation of a fast morphological processor for 1x3 SEs was presented. The efficiency of the proposed architectures made feasible their extension to 2–dim SEs, as a decomposition of 1–dim structures.

References:

[1] J. Serra, *Image Analysis and Mathematical Morphology*, New York: Academic Press, 1982.

[2] G. Matheron, *Random Sets and Integral Geometry*, Wiley, N.Y., 1975.

[3] P. Maragos and R. W. Schafer, "Morphological Systems for Multidimentional Signal Processing," *IEEE Proceedings*, vol. 78, no. 4, April 1990, pp 690–710.

[4] P. E. Danielsson and S. Levialdi, "Computer Architectures for Pictorial Information Systems," *IEEE Computer Magazine*, Nov. 1981, pp 53–67.

[5] A. C. P. Loui, A. N. Venetsanopoulos, and K. C. Smith, "Flexible Architectures for Morphological Image Processing and Analysis," *IEEE Trans. on Circuits for Video Technology*, vol. 2, no. 1, March 1992, pp 72–83.

[6] K. I. Diamantaras and S. Y. Kung, "A Linear Systolic Array for Real–Time Morphological Image Processing," *Journal of VLSI Singal Processing Systems*, vol. 17, 1997, pp 43–57.

[7] A. Broggi, G. Conte, F. Gregoretti, C. Sansoe, and L. M. Reyneri, "The PAPRICA Massively Parallel Processor," *Proc. MPCS–IEEE Int. Conf. on Massively Parallel Computing Systems*, IEEE CS–EuroMicro, Ischia, Italy May 2–4, 1994, pp 16–30.

[8] L. Abbott, R. M. Haralick, and X. Zuang, "Pipeline Architectures for Morphological Image Analysis," *Machine Vision and Applications*, vol. 1, 1988, pp 23–40.

[9] F. Y. Shih, C. T. King, and C. C. Pu, "Pipeline Architectures for Recursive Morphological Operations," *IEEE Trans. on Image Processing*, vol. 4, no. 1, January 1995, pp 11–17.

[10] Proceedings of the IEEE Computer Society Workshop for Pattern Analysis and Image Database Management, Miami Beach, Florida, November 1985.

[11] M. Maresca and H. Li, "Morphological Operations on Mesh Connected Architecture: A Generalized Convolution Algorithm," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, June 1986, pp 299–304.

[12] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI–9, no. 4, July 1987, pp 532–550.

[13] R. Loogheed and L. Tomko, "Robot Guidance Using a Morphological Vision Algorithm," *Proc. SPIE Conf. on Intelligent Robots and Computer Vision*, Sept. 1985.

[14] E. R. Dougherty, *An Introduction to Morphological Image Processing*, SPIE Optical Engineering Press, 1992.

[15] L. Lucke and C. Chakrabarti, "A Digit Serial Architecture for Gray–Scale Morphological Filtering," *IEEE Trans. on Image Processing*, vol. 4, no. 3, March 1995, pp 387–391.

[16] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed., Addison–Wesley, 1993

[17] E. N. Malamas, A. G. Malamos, and T. A. Varvarigou, "Fast Implementation of Binary Morphological Operations on Hardware–Efficient Systolic Architectures," *submitted to the Journal of VLSI Signal Processing Systems*, Kluwer Academic Publishers, USA.

[18] J. S. Ward, P. Barton, J. B. G. Roberts, and B. J. Stainer, "Figures of merit for VLSI implementations of digital signal processing algorithms," *IEEE Proc.*, vol. 131, part F, no. 1, Feb. 1984, pp. 64–70.

[19] I. Pitas and A. N. Venetsanopoulos, "Morphological Shape Decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, 1990, pp 38–45.

[20] X. Zuang and R. M. Haralick, "Morphological Structuring Element Decomposition," *Computer Vision, Graphics, and Image Processing*, vol. 35, September 1986, pp 370–382.

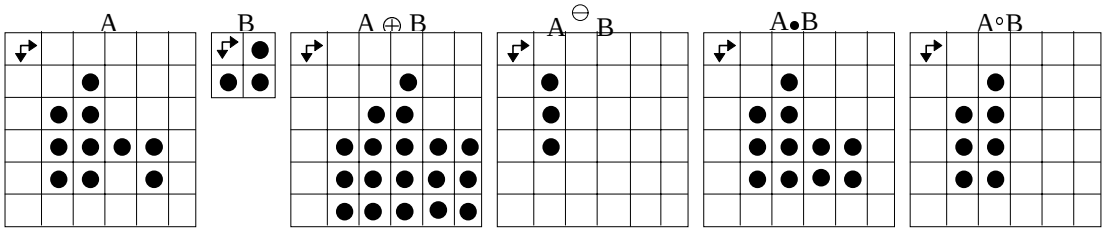
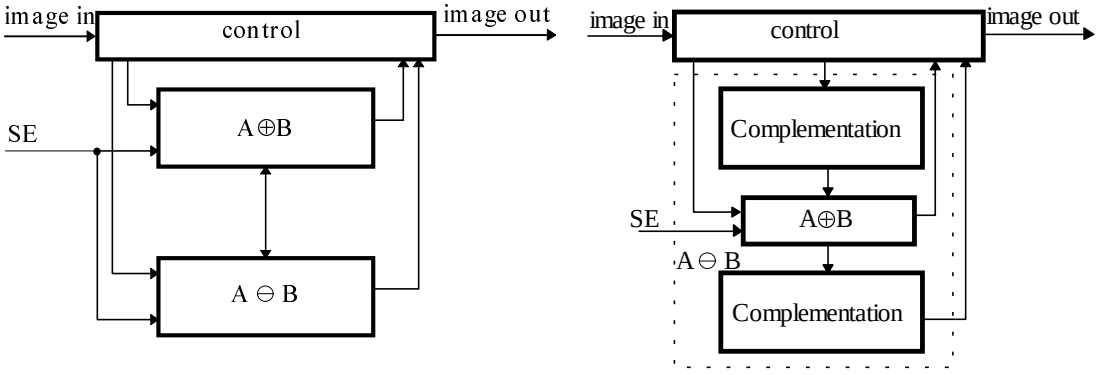


Fig. 1: Examples of the four morphological operations



(a) (b)

Fig. 2: Two basic architectures for morphological operations: (a) Separate dilation and erosion units, (b) A core dilation unit also used for erosion

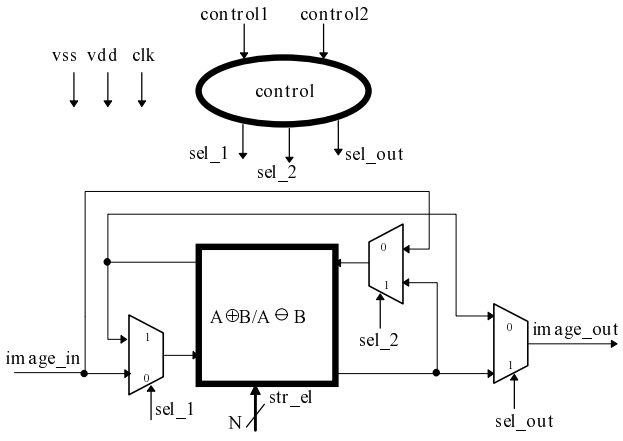


Fig. 3: The architecture of the EDU

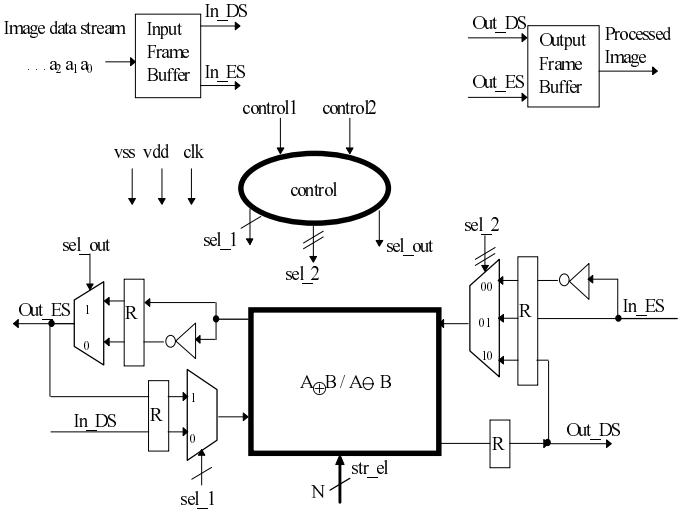
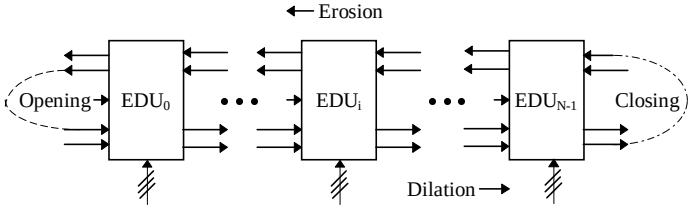
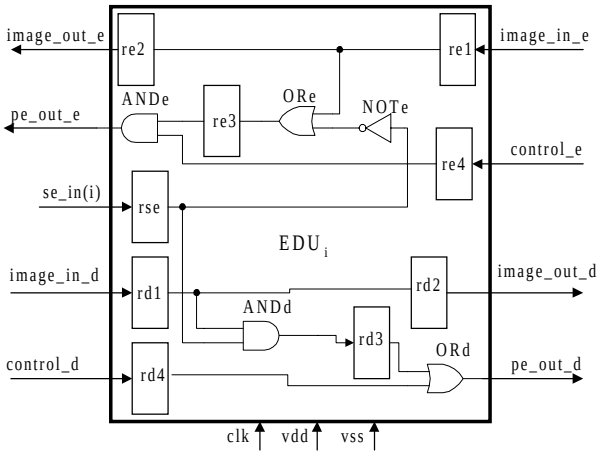


Fig. 7: An improved EDA for fast dilation/erosion



(a)



(b)

Fig. 4: (a) Implementation of the four operations with 1xN SE. (b) The I/O and internal structure of the EDU.

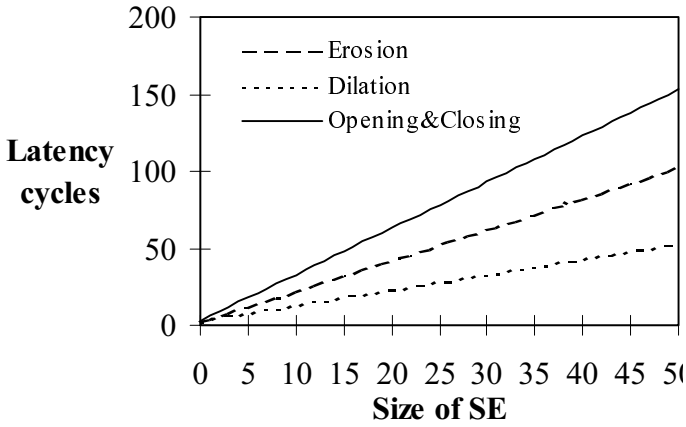


Fig. 5: Latency of operations performed by EDA

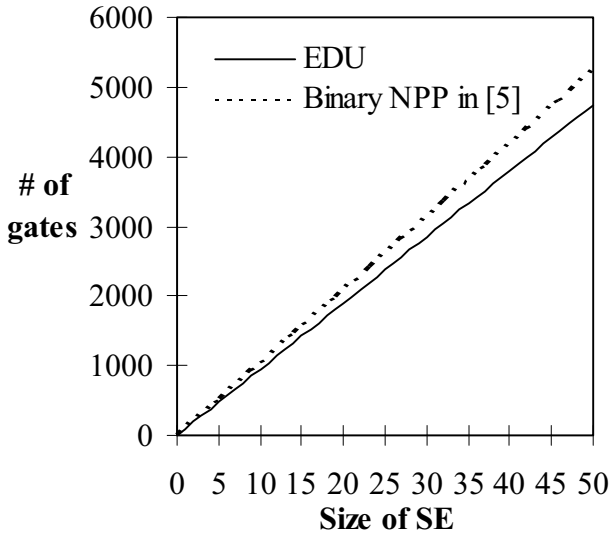


Fig. 6: The plot of the two cost functions

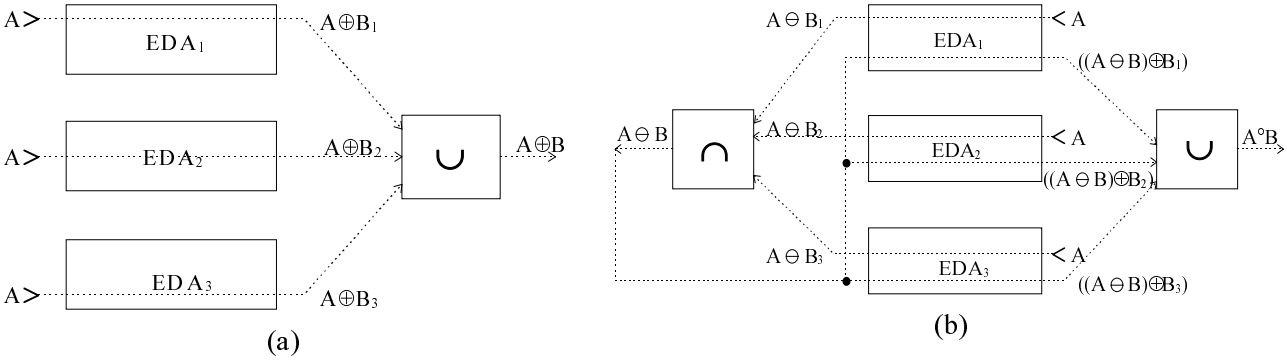


Fig 8: (a) A 2–dim dilation (b) A 2–dim opening