



Al-Khwarizmi Engineering Journal, Vol. 5, No. 1, PP 83-93 (2009)

Al-Khwarizmi  
Engineering  
Journal

## Multidimensional Systolic Arrays of LMS Algorithm Adaptive (FIR) Digital Filters

Riyadh A.H. AL-Helali \* and Bakir A. R. Al-Hashemy\*\*

\* Department of Electrical Engineering / AL-Mustansiriyah University

\*\* Department of Electrical Engineering / Baghdad University

(Received 12 June 2006; accepted 6 April 2009)

### Abstract

A multidimensional systolic arrays realization of LMS algorithm by a method of mapping regular algorithm onto processor array, are designed. They are based on appropriately selected 1-D systolic array filter that depends on the inner product sum systolic implementation. Various arrays may be derived that exhibit a regular arrangement of the cells (processors) and local interconnection pattern, which are important for VLSI implementation. It reduces latency time and increases the throughput rate in comparison to classical 1-D systolic arrays. The 3-D multilayered array consists of 2-D layers, which are connected with each other only by edges. Such arrays for LMS-based adaptive (FIR) filter may be opposed the fundamental requirements of fast convergence rate in most adaptive filter applications.

**Keywords:** Adaptive signal processing; LMS algorithm; VLSI signal processing.

### 1. Introduction

High-speed multidimensional digital filtering is very useful for real-time video signal processing such as video image coding, bandwidth compression, sampling rate conversion and the enhancement of televisions signals [1].

As a first step toward the implementation of an algorithm onto a processor array, it is common practice to go through a number of refinements (regularization, single assignment form...) that make the algorithm more suitable to a simple and modular VLSI design. The usual assumption is that the algorithm can be expressed by a system of recurrence equation [2].

Many emerging applications of specific systems characterized by many features which are the problem size begins to grow so as to guarantee better quantity of solutions and the problems must be implemented in real-time. These features demand high-performance computers. The algorithms, used by these applications, are typically highly parallel consisting of a huge number of simple and regular computations. These applications emphasis different demands on the processor design than the classical numerical

and necessitate finding the new architectural solutions for processors [3].

Systolic arrays have been one of the more interesting paradigms for reconfigurable computing in recent times because the design process creates an excellent paradigm for implementing algorithms via space and time transformations [4].

Real-time systolic implementations of multidimensional recursive digital filters have been discussed in prior work [5]-[8]. However, prior work has concentrated either on minimizing the critical path (that is, the unlatched signal path that requires the longest processing time) [5], [6] or on achieving the modularity or regularity of filter structures [7]-[8]. Thus the previously proposed filter structures do not satisfy the practical requirement of simultaneously having a critical path containing no more than one multiplier and one adder and the desired modularity and regularity. Moreover, they do not provide the property of local interconnectivity, which is essential in deep submicron VLSI implementations. In [1], proposed 1-D ladder filter structure based on which introduced a multilevel approach to derive multidimensional

ladder filter structures that are regular, modular and locally interconnected on the one hand, and yet have the shortest critical path of one multiplier plus one adder and the canonic number of storage registers in the higher dimensions (e.g., image line and frame registers).

Dedicated VLSI processors/chips are not always suitable for this purpose due to the need to provide flexible devices that can be adjusted to the new problems and problem sizes [9]. Thus, the processors must be reprogrammable or reconfigurable. Iso-plane method may be introduced [4]. In this case the problem is represented by iso-planes instead of a dependence graph in the classical case. Iso-planes describe shared variables and reduction operators. To extract more parallelism from a problem specification, the idea of increasing the dimensionality of problem representations is introduced. This is based on tiling and expanding of the index space [2].

Increasing the degree of parallelism demands an increase in the number of processor elements (PEs) to be employed. This is possible only either by increasing the dimensionality or the dimension of the array. Increasing the dimensionality of an array can result in a more efficient parallel algorithm implementation (parallel computing). This paper demonstrates how to extend existing designing methods for systolic arrays [11, 12, 13] and benefits from the approach in [4] to synthesize multilayered structures. This approach is demonstrated on the basis of 1-D convolution (inner product sum) of the output, which gives flexibility in the design procedure for extracting more than one array. Systolic arrays for 1-D LMS-based adaptive (FIR) filter that decreases the latency has the area complexity  $O(n)$  while the latency time complexity is  $O(n^{1/2})$ . Arrays with the increased throughput rate have the area complexity  $O(sn)$ ,  $1 \leq s < N$ , and the time complexity is  $O(N/s)$ , where  $n$  and  $N$  are sizes of two sequences of the inner product sum output.

This decreasing latency time and increasing throughput rate limits the inherent limitations of LMS algorithm, which necessitates a compromise between the opposing fundamental requirements of fast convergence rate and small maladjustment for high area complexity.

In section 2, a mapping of regular algorithms onto multilayered 3-D reconfigurable processor array is described to design the systolic arrays, that it achieves the optimum local interconnection pattern and the highest speed. The complete 2-D and 3-D design of systolic LMS-based adaptive (FIR) digital filter is given in section 3. In section

4, we discuss the throughput and processing rate of the system. A comparison is made between the LMS systolic array and the serial design employing fast convolution and FFT as regards throughput. Finally, we present conclusions in section 5.

## 2. Mapping Procedure

Regular array design means that the order of computations of reduction operators and the order of spreading shared variables must be chosen [4]. Imposing a partial order onto iso-planes will do this. Different partial orders can develop to improve the array design that affects on the structure and time parameters, such as latency and throughput rate.

Imposing a partial order onto an iso-set produces dependence vectors  $\delta$ :  $y = x + \delta$ ,  $x, y \in \text{ISO}^v(x)$ . Therefore, the affine equations are transformed into uniform recurrences. To offer a regular array design, the number of dependence vectors must be as small as possible.

The corresponding regular array can be derived by an affine space-time mapping [2].

$$IS' = T(IS) + Y \quad \dots(1)$$

Where  $IS \subset Z^r$  and  $IS' \subset Z^r$  are the source and the target polytopes, respectively,  $T \in Z^{r \times r}$  is a transformation matrix and  $Y = Z^r$  is an offset vector.

Space-time mapping transforms the source polytope  $IS$  into a target polytope,  $IS'$ , that contains the same points but in a new coordinate system in space and time: every  $x \in IS$  is mapped to the possible time step:

$$t(x) = \lambda x + \alpha \quad \dots(2)$$

And to the possible PE:

$$a(x) = \sigma x + \beta \quad \dots(3)$$

Where  $\lambda \in Z^r$  is a time schedule and  $\sigma \in Z^{(r-1) \times r}$  is allocation,  $\alpha \in Z$ ,  $\beta \in Z^{r-1}$ .  $\lambda$  and  $\sigma$  are components of the transformation [4]

$$T = \begin{bmatrix} \lambda \\ \sigma \end{bmatrix} \quad \dots(4)$$

To be a valid mapping, transformation  $T$  must satisfy certain constraints:

1. The time schedule  $\lambda$  must be consistent with the dependence vector  $\delta$ , i.e.  $\lambda\delta > 0$ .
2. The second constraint excludes the conflict that two distinct computations would be scheduled for the same processor at the same time, i.e. transformation  $T$  must be bijective. And thus  $|T| \neq 0$ . If the transformation is unimodular, i.e.  $|T| = 1$ , then the resulting array uses each time step efficiently, and, thus, this design is usually preferred. In this case, when  $|T| = k$ ,  $k > 1$ , each processor works only in every  $k$ th time steps, thus less efficiency.

The channels between processors can be computed from dependence vectors and transformation. Each dependence vector  $\delta$  produces a channel  $l$ :

$$l = T\delta \quad \dots(5)$$

With the direction:

$$l_{dir}(\delta) = \sigma\delta \quad \dots(6)$$

And the delay:

$$l_{del}(\delta) = \lambda\delta - 1 \quad \dots(7)$$

### 3. Systolic Design of LMS Algorithm Using ISO-plane Method

The systematic design of this method gives various (1-D) and (2-D) arrays with modified property comparison to classical method as follows:

#### 3.1 (1-D) Classical Systolic Array of LMS Algorithm

1-D convolution sum of the inner product of the tap-weight vector and the tap-input vector, as represented as [10];

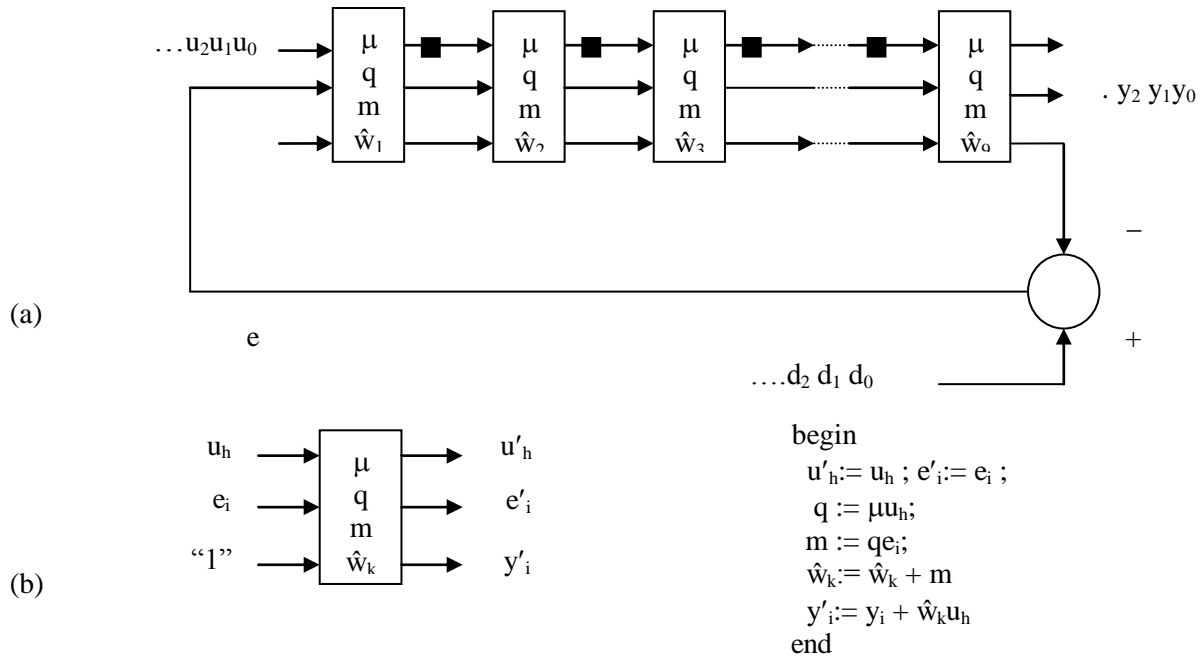


Fig. 1. (a) 1-D systolic array realization for LMS algorithm. (b) Functional specification of a PE.

$$y(n) = \sum_{k=1}^M w_k^* u(n-k+1) \quad \dots(8)$$

$$= \hat{w}^H(n)u(n)$$

When the superscript  $H$  signifies Hermitian transposition, and the inner product of the tap-weight vector  $\hat{w}(n)$  and the tap-input vector  $u(n)$  represents the output of the filter, and is specified as:

$$(\forall i: 0 \leq i < N : \quad \dots(9)$$

$$y[i] = (\sum j: 1 \leq j < M : \hat{w}[j]u[i-j+1]),$$

Where  $\hat{w}[j]$  and  $u[i]$  are weight and input sequences respectively, and  $y[i]$  is an output sequence. In practical applications  $M \ll N$ . Two shared variables,  $\hat{w}[k]$  and  $u[i]$ , are used by several operators (multiplications) for computing different instances of the variable  $y[i]$  by the reduction operator  $\sum$ . The index space is a 2-D polytope:

$$IS = \{[i,j]^T \mid 0 \leq i \leq N \wedge 1 \leq j < M\} \quad \dots(10)$$

Each instance,  $i$ , of a variable, say  $v[i]$ , is represented by an iso-plane  $ISO^v(x)$ ,  $x \in IS$  [4]. The respective source polytope with iso-planes is depicted for Kernel size  $M = 9$  and  $N = 15$ .

The 2-D index space for 1-D convolution sum of the inner product  $\hat{w}^H(n)u(n)$  gives the channels applied to each point of allocation space, the respective allocation direction and the direction of channels for variables. Therefore; the required 1-D LMS systolic array can be derived from the inner product algorithm as derived in [14] and represented as shown in figure (1).

### 3.1.1 Decreasing the Latency

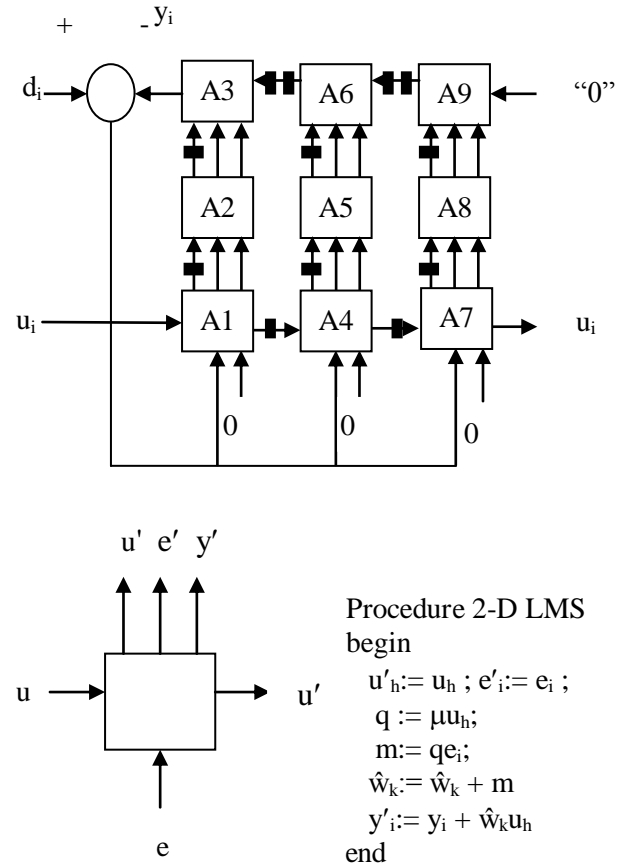
It is possible to speed-up the computations using recursive doubling that guarantees the logarithmic time complexity in the case of associative and commutative reduction operators. The same approach can be used and decompose the reduction operator into several parts that will be evaluated in parallel. This gives an increase in the degree of parallelism and decrease in the latency [4]. Partitioning and expanding the index space can be used; such step increases the array dimensionality.

Partitioning of the range of index,  $i$ , is defined by a vector:

$$P_i = [p_j], 0 \leq j < r \quad \dots\dots(11)$$

Where  $r$  is the dimensionality of the index space. If  $j = i$ , then  $P_j > 0$ , otherwise,  $p_j = 0$ . The vector  $P_i$  defines the factorization of the upper bound,  $N$ , of index,  $i$ , by setting  $N = p_i s_i$  for appropriate natural numbers  $p_i$  and  $s_i$ .

Depending on the inner product  $\hat{w}^H(n)u(n)$  of LMS algorithm, the 2-D systolic array realization for LMS algorithm can be derived. The large boxes in Figure (2-a) represent the processor elements and its functional specification is in (b) of the same Figure.



**Fig. 2. (a) (2-D) systolic array realization of LMS algorithm. (b) Functional specification of a PE.**

### 3.1.2 Increasing the Throughput Rate

Classical systolic arrays implement LMS algorithm sequentially, i.e., they input and output data sequentially. This put the limit onto throughput rate and time complexity – the linear

complexity is the best that can be achieved. To improve the time parameters of systolic arrays, several data items must be entered into the array simultaneously. This is achieved using partitioning of the range of input stream, forming thus a 2-D input stream instead of a 1-D stream in the classical case.

The data items must be entered into the array in parallel, for each  $p_i$  data inputs and thus needs a new block which converts the serial (M-tap) data sequence into the required  $p_i$  parallel data that named serial to parallel converter [15]. The systolic array representation of the convolution sum, which was derived in [4] and increased the throughput rate, is used to derive the corresponding LMS array in Figure (3).

### 3.1.3 Improving the Latency and Throughput

To increase the throughput rate and decrease the latency, one can partition the ranges of indexes: the ranges of index  $i$  that describes the reduction operator and the range of index  $j$ , which described the input stream. The corresponding 4-D polytope shape has index point:  $x = [i', j', l, k]^T$  and the allocation is taken along the axis  $k$  that provides regular structures as depicted in [14].

Depending on the inner product  $\hat{w}^H(n)u(n)$  of the required adaptive transversal filter with the assumed coefficients order, the whole array structure of the LMS algorithm can be suggested by entering the error signal through the array and compute the updated values of the tap-weight vector, the filter output is shown in Figure (4).

In this improvement, also the data items must enter, simultaneously for each  $p_i$  inputs, into the general array structure of the layers that prepare the incoming  $p_i$  inputs data according to certain sequence to enter to its own (2-D) array layer and then get the outputs from the summation circuits assigned to each layer structure. Thus, the whole systolic array represents a 3-D systolic array of the size  $3 \times 3 \times 3$  for a 1-D computational operations as shown in Figure (4).

## 4. Throughput and Processing Rate

Now by using the systolic arrays and applying the simulation of data flow, we find that:

### 4.1 Classical systolic arrays

The designed systolic arrays of the inner product  $\hat{w}^H(n)u(n)$  and the LMS algorithm by iso-plane method have the same performance of quantities of that designed by dependence graph method, but gives flexibility of increasing in the degree of parallelism and improving the algorithmic properties by partitioning and expanding the index space as in the previous section.

### 4.2 Systolic array to decrease the latency

By taking the 2-D semi-systolic array realization for the LMS algorithm that designed in chapter three in Figure (2). Consider the data flow for several consecutive clock periods, applying the procedure to be executed by each cell, added the intermediate results and getting the output signal sample, and then the algorithm gives:

- Number of cells:  $= M$ .
- Running time: clock periods  $= M$ .
- Commutative product-type measures:  $(M) M$ .
- Efficiency and utilization: 100%.
- Broadcasting of the error signal, this is propagated to all cells.
- The throughput rate of the 2-D LMS array is the same as the 1-D or conventional LMS array (only for this case). The latency time is reduced because the data that passes through the pipe for processing, takes short path (less number of cells ( $M^{1/2}$ ) than the first pipe that has  $M$  cells), therefore, the amount of time the data stays in the pipe (of  $M^{1/2}$  cells) is less than the time that the data stays in the pipe (of  $M$  cells). The overall computation time is:

$$T_o = (2 T_x + 2 T_+)M \quad \dots(12)$$

Where  $T_x$  &  $T_+$  are the necessary times to compute one multiply and one add operations.

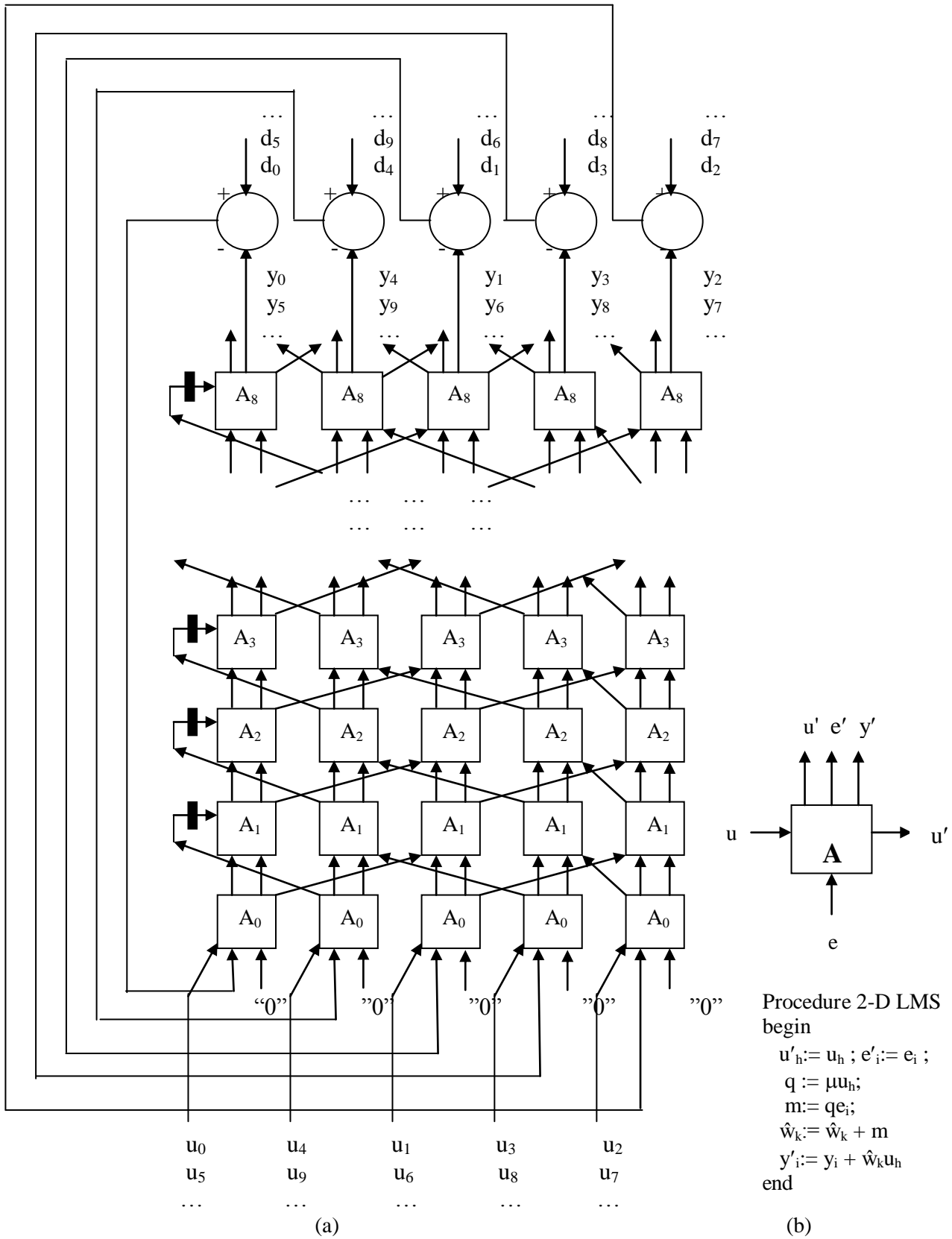


Fig. 3. (a) (2-D) systolic array realization of LMS algorithm. (b) Functional specification of a PE.

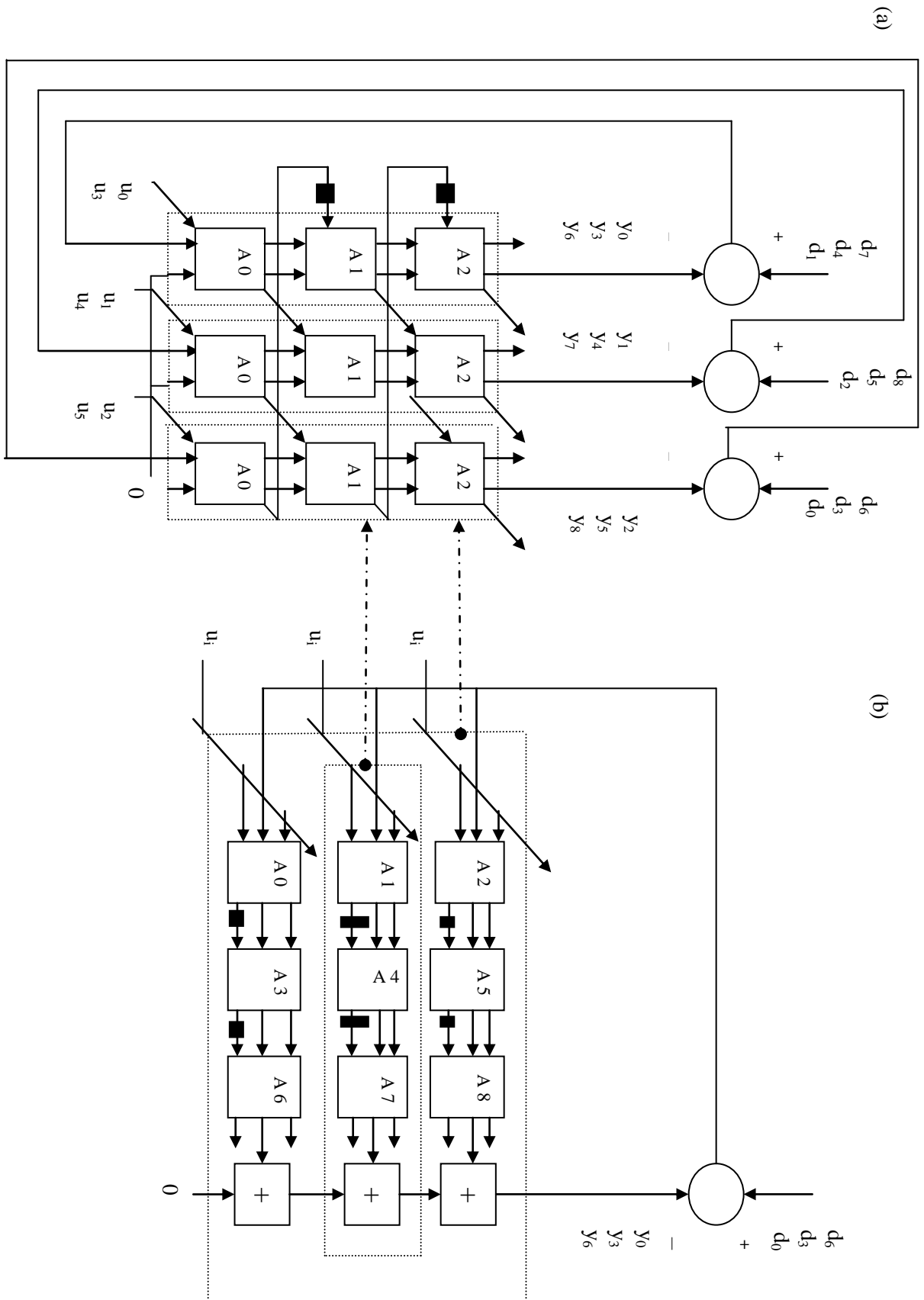


Fig. 4. 3-D systolic array of the size 3x3x3 for the LMS algorithm (for M=9), (a) the general structure and the interconnection pattern of layers, (b) The layer structure of the 2-D piecewise regular array of LMS algorithm.

### 4.3 Systolic array to increase the throughput rate

By simulating the data flow that enters the 2-D semi-systolic array realization of LMS algorithm designed of Figure (3), to improve the time parameters (throughput rate), five data items were entered into the array simultaneously, broadcasting to the input of the five lower cells. Then by computational activities of the simulation, the array gives that:

- Number of cells:  $= (p_i)M$ , or the area complexity is  $O(p_iM)$ , where  $p_i$ , represents number of the parallel inputs to the array (here,  $p_i = 5$ ).
- Running time: clock periods  $= M$ .
- Commutative product-type measures:  $(p_iM)M$ .
- Efficiency and utilization: 100%.
- Broadcasting of the  $p_i$  parallel inputs signal, error signal, which are propagated to all cells.
- Throughput rate has been improved, from the operation of the array, for each number of data outputs ( $N$ ), there are sets of data inputs, which entered simultaneously into the array.

Therefore, the time complexity is  $O(N/p_i)$ . The input signal enters the first cell of the array and exits from the last cell of the pipe of the array. After applying the computational activities in the ( $M$ ) cells, the latency is the same as in the classical case:  $O(M)$ .

### 4.4 Systolic array for improving the latency and throughput

By combining the last two 2-D LMS systolic arrays, a 3-D multilayered LMS systolic array was obtained, which improves the two above features. The simulations of data flow and computational activities for the array of Figure (4) give:

- Number of cells:  $= (p_i)M$ , or the area complexity is  $O(s_j p_i^2) = O(p_iM)$ , where  $p_i^2$ , represents number of the parallel inputs to the array (here,  $p_i = 3$ ), (here,  $s_j = 3$ ) and ( $s_j p_i = M$ ).
- Running time: clock periods  $= M$ .

- Commutative product-type measures:  $(p_iM)M$ .
- Efficiency and utilization: 100%.
- Broadcasting of the  $p_i$  parallel inputs signal, error signal, which are propagated to all cells of the 3-D systolic array with size  $3 \times 3 \times 3$  for 1-D LMS systolic array.
- Throughput and latency are improved to achieve the requirements of the 3-D multilayered array design and equal to  $O(N/p_i)$ ,  $O(M^{1/2})$  respectively.

From the obtained simulation results, an efficient 3-D systolic LMS array algorithm design can be selected to represent the preferable systolic design.

The throughput rate of the 3-D designed systolic array is greater than that of the architectures [16], [17], but smaller than that of the pipelined architecture [18]. In [18], drawbacks of broadcasting the input signal, the overall complexity are high and its output latency is equal to the FIR filter length (i.e.,  $L$ ).

Thus our algorithm has single assignment form,  $L$ -cells, high throughput rate, 100% utilization,  $L^{1/2}$  latency,  $L$ -clock periods for one sample block, and finally the flexibility in the design, which takes the inner product (convolution sum)  $w^H(n)u(n)$  in the design consideration, which enables us to extract more than one algorithm for the same problem.

### 4.1.3 Conclusions

From the Systolic arrays were designed and simulating the data flows entering into the arrays, we deduced the following conclusions:

- ❖ Various 2-D and 3-D systolic arrays of 1-D LMS-based adaptive (FIR) transversal digital filters were designed which can effectively be implemented using only local communications on a parallel system comprised of combinatorial circuits, clocked delay elements and distributed memory.
- ❖ The technique described exploits the recurrence inherent in the application and is based on the convolution sum (inner product  $\hat{w}^H(n). u(n)$ ) in the LMS algorithm, which gives flexibility of deriving more than one systolic algorithm from the basic systolic structure.
- ❖ Proposed systolic arrays for 1-D LMS algorithm allow an increase in the number of processors to increase throughput rate and to reduce the latency that is not achievable using



the conventional systolic array synthesis method.

- ❖ The final systolic array of LMS-based adaptive transversal filter is simple and exhibit increased throughput rate and decreased the latency, which is 3-D regular systolic algorithm making it amenable to VLSI implementation and can be used for a large variety recurrence computations in signal and image processing. However, the output of our efficient adaptive transversal filter is delayed by  $M^{1/2}$  samples, where  $M$  is the filter length.
- ❖ Unlike the other structures, the throughput is independent of the filter length, implying that LMS adaptive FIR filter systolic array with several hundreds of filter coefficients can be represented by a word-level systolic arrays (multibit numbers). A word-level systolic array considers the inner product step as a typical operation for various signal-processing operations.

The future work on a 2-D LMS adaptive nonrecursive (or FIR) digital filters with excitation  $u(n_1, n_2)$ , response  $y(n_1, n_2)$  and order of the pair  $(M_1, M_2)$  to extract a systolic array for the algorithm. Similarly by taking the 2-D inner product step of the weight vector and the input vector as a basic systolic representation of the algorithm, it can be designed it by the dependence graph method. The 2-D LMS algorithm can formulate as:

$$e(n_1, n_2) = d(n_1, n_2) - y(n_1, n_2) \quad \dots(13)$$

$$\hat{w}(n_1+1, n_2+2) = \hat{w}(n_1, n_2) + \mu e(n_1, n_2) u(n_1, n_2) \quad \dots(14)$$

Where  $d(n_1, n_2)$  is a 2-D desired response, and  $e(n_1, n_2)$  is a 2-D error signal.

By using iso-plane method considered, partition one of the four indices of the 2-D convolution sum to reduce the latency and, by the same method try to obtain an (3-D and 4-D) LMS algorithm which has improved throughput and latency features.

Systolic representation of the 2-D adaptive digital filters serves in many important applications such as reduction of noise in images, enhancement of edges in images, processing of geophysical signals and other geological applications [19], since continuous signals are encountered that are inherently (2-D).

## Abbreviations

Del	Delay
dir	Direction
FIR	Finite-impulse-response
FFT	Fast Fourier Transform
IPS	Inner Product Step
IS	Index Space
ISO	Iso-plane
LMS	Least-mean square
PEs	Processor Elements
VLSI	Very-large scale integration

## Symbols

$A^*$	set of interconnection pattern
$d(n)$	desired response
$E$	expected value expected value
$e$	projection vector
$e(n)$	error signal vector
$H$	Hermitian transposition
$M$	number of taps
$N$	set of nonnegative integer numbers
$O( )$	the class of all functions
$p_i$	parallel inputs
$T$	transposition
$t$	time of execution
$T_+$	time execution of the addition operation
$T_\times$	time execution of the multiplication operation
$u(i)$	discrete-time series
$u(n)$	observation vector
$V^*$	set of cells/processors
$w(n)$	weight-error vector
$\hat{w}(n)$	estimate value of the weight-error vector
$w^*$	conjugate value of $w$
$w_0$	optimum tap-weight vector
$y(n)$	output signal vector
$Z$	set of integer numbers
$Z^n$	dependence graph space
$Z^{n-1}$	processor space

$z^*$	processor location
$d, \delta$	dependence vectors
$\rho$	Offset vector
$\mu$	Step-size parameter
$\sigma$	allocation
$\sigma^2$	variance
$\lambda$	time schedule
$\eta$	basis vector basis vector

## Acknowledgements

The authors would like to thank the anonymous reviewers for their useful comments.

## 6. References

- [1] L. Xiaojian and T. B. Leonard, "High-Speed Systolic Ladder Structures for Multidimensional Recursive Digital Filters," *IEEE Trans. Signal Processing*, vol. 44, no. 4, pp. 1048-1055, Nov. 1996.
- [2] F. Lorenzelli and K. Yao, "A linear systolic array for recursive least squares," *IEEE Trans. Signal Processing*, vol. 43, no. 12, pp. 3014-484, Apr. 1992.
- [3] N. Petkov, *Systolic Parallel Processing*. Elsevier Science Publishers, North Holland, 1993.
- [4] T. P. Plaks, "Mapping Regular Algorithms onto Multilayered 3-D Reconfigurable Processor Array," *IEEE Proceedings of the 32<sup>nd</sup> Hawaii International Conference On System Sciences*, pp. 1-10, 1999.
- [5] K. M. Ty and A. N. Venestanopoulos, "A fast filter for real-time image processing," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 948-957, Oct. 1986.
- [6] R. Gnanasekaran, "2-D filter implementation for real-time signal processing," *IEEE Trans. Circuits Syst.*, vol. 35, no. 5, pp. 587-590, May. 1988.
- [7] W. Luk and G. Jones, "Systolic Recursive Filters," *IEEE Trans. Circuits Syst.*, vol. 35, no. 8 pp. 1067-1068, Aug. 1988.
- [8] S. Sunder and V. Ramachandran, "Systolic implementation of multidimensional nonrecursive digital filters," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 6, pp. 399-407, Dec. 1993.
- [9] J.M. Jover and T. Kailath, "A parallel architecture for Kalman filter measurement update and parameter estimation," *Automatica* vol. 22, no. 1, pp. 43-57, 1986.
- [10] S. Haykin, *Adaptive Filter Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1986.
- [11] P. Quinton and Y. Robert, *Systolic Algorithms and Architectures*. Prentice Hall, Masson, UK, 1991.
- [12] S. Rao and T. Kailath, "Regular iterative algorithms and their implementations on processor arrays," *Proc. IEEE*, VOL. 76, NO.3, PP. 259-282, Mar. 1988.
- [13] J. Teich and L. Thiele, "Partitioning of processor arrays: A piecewise regular approach," *INTEGRATION*, VOL. 14, NO.3, PP. 297-332, 1993.
- [14] R. A. H. Al-Helali, "Systolic Algorithms of LMS, RLS Adaptive (FIR) Digital Filters for Adaptive Channel Equalization," M.Sc. Thesis, Univ. of Baghdad, Baghdad, Oct. 2005.
- [15] A. V. Oppenheim, and R. W. Schaffer, *Digital Signal Processing*. London: Prentice Hall, 1975.
- [16] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoustic., Speech, Signal Processing*, vol. 37, pp. 1397-1405, Sept. 1989; vol. 40, pp. 230-232, Jan. 1992.
- [17] D. L. Jones, "Learning characteristics of transpose-form LMS adaptive filters," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 745-749, Oct. 1992.
- [18] S. C. Douglas, Q. Zhu, and K. F. Smith, "A pipelined LMS Adaptive FIR Filter. Architecture Without Adaptation Delay," *IEEE Trans. Signal Processing*, vol. 46, pp. 775-778, Mar. 1998.
- [19] A. Antonio, *Digital Filters Analysis, Design, and Applications* McGraw-Hill, 1993.

## المصفوفات المتعددة الأبعاد من المعالجات المتوازية لخوارزمية أقل مربع المعدل للمرشح الرقمي المكيف ذات الاستجابة المحدودة الإخراج

رياض علي عبد الحسين الهلالي\* باقر عبدالرسول الهاشمي\*\*

\* قسم الهندسة الكهربائية / كلية الهندسة / الجامعة المستنصرية  
\*\* قسم الهندسة الكهربائية / كلية الهندسة / جامعة بغداد

### الخلاصة

تتضمن المقالة تصميم منظومات بشكل مصفوفات من المعالجات (الخلايا الانقباضية في جسم الإنسان) لخوارزمية مربع أدنى معدل للمرشح الرقمي وذلك باستخدام طريقة يتمثل (تحويل) الخوارزميات الرياضية المنظمة (المرتبة) إلى منظومة من المعالجات المتوازية. بل اعتماد على مصفوفة المعالجات الأحادية الأبعاد للمرشح الرقمي والذي بدوره يعتمد في عمله على مجموع حاصل ضرب مصفوفة الإخراج، تم اشتقاق مجموعة أنواع من المصفوفات والتي تشكل ترتيب منظم من المعالجات (الخلايا) وطريقة الارتباط الداخلي بين هذه الخلايا والتي تجعلها مهمة الاستخدام في دوائر التكامل القياسية الكبيرة جدا (VLSI). إن الزمن المستغرق للحصول على قيمة واحدة قد أنخفض وكذلك معدل الإخراج قد ازداد مقارنة بمصفوفات ذات الأبعاد الأحادية. إن المصفوفة الثلاثية الأبعاد والمتعددة الطبقات تتألف من طبقات ثنائية الأبعاد والتي ترتبط مع بعضها البعض عن طريق الحافات فقط، إن مثل هذه المصفوفات لخوارزمية مربع أدنى معدل للمرشح الرقمي يمكنها تلبية الاحتياجات في تسريع معدل الاقتراب للنتائج النهائي في معظم تطبيقات المرشحات الرقمية الملائمة.