

Application of Systolic Architectures and Switched Capacitor Techniques to Implement Recursive Filters

DJAMEL CHIKOUCHE, RAIS EL HADI BEKKA
Electronics Department, Faculty of Engineering
University of Setif
19000 Setif
ALGERIA

Abstract: The conditions required by the majority of signal processing algorithms can be really satisfied in the design of parallel processors in VLSI technology. In this paper, the discrete state space recursive filters are implemented in the form of array processors. The state space description permits the straightforward application of systolic architectures of the Kung-type to realize recursive filters of both 1D and 2D types. We show that the recursivity inherent to the filtering algorithm introduces a latency proportional to the filter order which has a direct effect on the computation throughput of these structures. Moreover, we show that the use of CTP decomposition technique together with the cylindrical-type structures reduces significantly this latency and improves the computation throughput of these arrays. The processing cells of the systolic array are designed via Switched-Capacitor techniques.

Key Words: Recursive filters, Systolic, Cylindric, CTP, Switched-capacitor, processing elements.

1 Introduction

The concept of systolic architecture has been introduced by H. T. Kung in his paper "Why systolic architectures?" [1] as a general methodology for mapping high-level computations into hardware structures. According to Kung, in a systolic system, data flows from the computing machine memory in a rhythmic fashion, passing through many processing elements before it returns to memory, much as blood circulates to and from the heart. Moreover, to implement a variety of computations, data flow in a systolic system may be done at multiple speeds in multiple directions (both inputs and partial results flow, contrarily to classical pipelined systems, where only results flow in the system).

Consequently, a systolic system can be easily implemented, because of its regularity and can be easily reconfigured (in order to meet the various outside constraints) because of its modularity. The concept of systolic architecture was developed for the first time during the years 1979 and 1980 at the Carnegie-Mellon-University [1], and many versions of systolic processors have been designed and constructed by several industrials and researchers [1-12].

In a previous work [7, 8], we have presented a methodology for the implementation of state space recursive filters on systolic architectures of the Kung-type [1] and the cylindrical-type [3]. In this paper, we present a review of the application of systolic system concept (of both the Kung-type and the cylindrical-one) to the realization of discrete recursive filters described in the state space by a simple matrix equation. We will show that the recursivity inherent to the filtering algorithm introduces a latency proportional to the filter order which has a direct effect on the computation throughput of these architectures. Furthermore, the use of CTP decomposition technique [7, 8] together with the cylindrical structures can considerably reduce the latency of the array, thus improving its computation throughput rate.

We will start our study by introducing the principle of the Kung-type systolic implementation of 1D discrete recursive filters. Systolic structures of the cylindrical-type together with the CTP technique are considered in section 3 for the implementation of discrete recursive filters. In the last section, we propose the design of processing elements, of the

different systolic architectures presented in this paper, by using switched-capacitor architectures.

2 Systolic structure for discrete recursive filters

A discrete recursive filter can be described in the state space domain by the following two equations :

$$\begin{aligned} x(n+1) &= Ax(n) + Be(n) \\ y(n) &= Cx(n) + De(n) \end{aligned} \quad (1)$$

or, in a matrix form as:

$$\begin{bmatrix} x(n+1) \\ y(n) \end{bmatrix} = \begin{bmatrix} AB \\ CD \end{bmatrix} \begin{bmatrix} x(n) \\ e(n) \end{bmatrix} \quad (2)$$

where: A, B, C, and D are the state matrices of the filter, $x(n) \in \mathbb{R}^N$ the state signal vector of dimension $(N \times 1)$, $e(n) \in \mathbb{R}$ the input signal and $y(n) \in \mathbb{R}$ the output signal.

The internal state space description of the filter permits to represent the filtering algorithm as a simple product of a square matrix with a column vector.

The systolic array implementation of the discrete filter, represented in figure 1 uses the the global state matrix elements to load the PE's memories of the systolic array.

The computation throughput of the systolic architecture of figure 1 is estimated to

$$\frac{1}{(2N+1)(t_m + t_a)}$$

where t_m and t_a are respectively the times required to perform a multiplication and an addition.

In the next section, we will show that the use of CTP technique together with systolic architectures of the cylindrical-type [7, 8] permits to improve the computation throughput of these structures.

3 Fast systolic architectures with dynamic reconfiguration for discrete recursive filters

Consider an $(N-1)^{\text{th}}$ order 1D discrete recursive filter ($N = pq$) described by equation (2). Let:

$$\mathbf{H} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} x(n+1) \\ y(n) \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}$$

Equation (2) is then equivalent to the following linear relation:

$$\mathbf{v} = \mathbf{H}\mathbf{u} \quad (3)$$

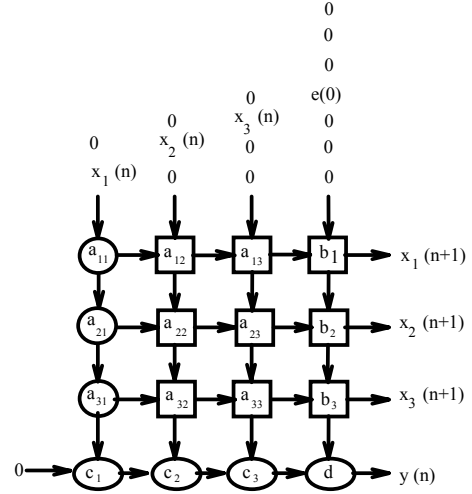


Fig. 1 Systolic implementation of a third order discrete recursive filter.

In this section, we will apply the CTP decomposition technique [7] to our recursive filtering algorithm (3) in order to obtain a faster form.

Consider the example of a third order recursive filter described by the state space equation (3) with $N = 4 = 2 \times 2$, $p = q = 2$, and:

$$\mathbf{H} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \\ c_1 & c_2 & c_3 & d \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \\ x_3(n+1) \\ y(n) \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ e(n) \end{bmatrix}$$

A single term CTP decomposition of \mathbf{H} can be found by using methods of [8]. This decomposition is defined by the following (2×2) matrices \mathbf{L} and \mathbf{R} :

$$\mathbf{L} = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}$$

such as \mathbf{H} is the tensor product of \mathbf{L} and \mathbf{R} . Mapping the vector \mathbf{u} on a $(p \times q)$ matrix \mathbf{U} by using segments of \mathbf{u} as columns of \mathbf{U} , we get:

$$\mathbf{U} = \begin{bmatrix} x_1(n) & x_3(n) \\ x_2(n) & e(n) \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} x_1(n+1) & x_3(n+1) \\ x_2(n+1) & y(n) \end{bmatrix}$$

The matrix \mathbf{V} is obtained by the same procedure from the vector \mathbf{v} .

The CTP expansion associated with equation (3) takes then the following fast form:

$$\mathbf{V} = \mathbf{LUR} \quad (4)$$

The cylindrical arrays of [3] are compatible with the CTP decomposition. Fig. 2 represents a cylindrical array performing the (2×2) matrix-matrix product \mathbf{LU} . The triangular figures denote local memory wherein elements of the matrix \mathbf{L} are stored as indicated in Fig. 2a. We transmit the columns of \mathbf{U} down the longitudinal paths. At each node, the longitudinal input is multiplied by the scalar stored in its internal register. The resulted product is added to the input arriving along the transversal path. This sum is retransmitted transversally. The longitudinal sequence is retransmitted without alteration. Fig. 3a depicts the calculation at the start of the second step. Fig. 2b shows the computation at the second step.

We assume our array operates synchronously. The sequences available on the transversal paths at the bottom of the array are the rows of \mathbf{LU} . We can verify that the top row nodes complete their computations at the same time with the completion of computation of the first row of \mathbf{LU} by the bottom row nodes. At the p^{th} step (here $p=q=2$), the array is switched as indicated in Fig. 2b. The row sequences of (\mathbf{LU}) are fed back on the transversal paths of the input nodes.

The \mathbf{R} row sequences follow the \mathbf{U} row sequences on the longitudinal paths. When the new computation starts down the array, the node operation changes to another form.

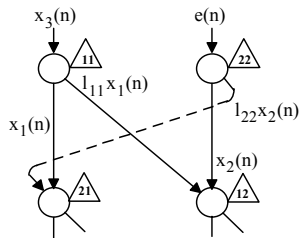
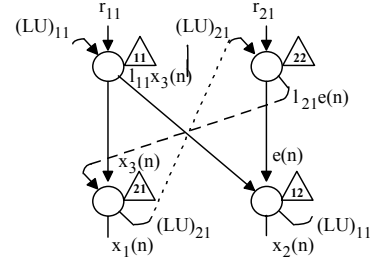
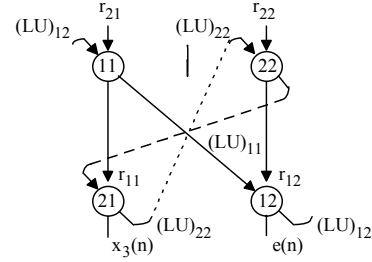


Fig. 2a Step 1.



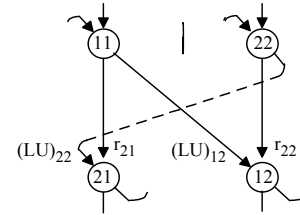
$$\begin{aligned} (LU)_{21} &= l_{21}x_1(n) + l_{22}x_2(n) \\ (LU)_{11} &= l_{12}x_2(n) + l_{11}x_1(n) \end{aligned}$$

Fig. 2b Step 2.



$$\begin{aligned} (LU)_{22} &= l_{21}x_3(n) + l_{22}e(n) \\ (LU)_{12} &= l_{12}e(n) + l_{11}x_3(n) \end{aligned}$$

Fig. 2c Step 3.



$$\begin{aligned} V_{11} &= (LU)_{11}r_{11} + (LU)_{12}r_{21} \\ V_{22} &= (LU)_{21}r_{12} + (LU)_{22}r_{22} \\ V_{21} &= (LU)_{21}r_{11} + (LU)_{22}r_{21} \\ V_{12} &= (LU)_{11}r_{12} + (LU)_{12}r_{22} \end{aligned}$$

Fig. 2d Step 4.

Fig. 2 Operating principle of the fast cylindrical array with dynamic reconfiguration of a third order filter.

This time, the node retransmits all input sequences unchanged while iteratively calculating the dot product of these sequences. This product is stored at the node memory as indicated in Fig. 2. The switch in function of the nodes will propagate down the array together with the first arrival of \mathbf{LU} and \mathbf{R} data. Fig. 2c shows the computational wave front reaching the second row. The components of $\mathbf{V} = \mathbf{LUR}$ are stored in the memories at the $(p+q)^{\text{th}}$ step of this sequence. The indices i, j on the nodes of Fig. 2d represent the location of V_{ij} . Therefore, using the

same cylindrical arrays, the matrix-matrix operation $\mathbf{V}=\mathbf{LUR}$ can be computed in $O(p+q)$ time units while the matrix-vector operation $\mathbf{v}=\mathbf{H}\mathbf{u}$ takes $O(pq)$ time. We can clearly see the superiority in computational speed of the first linear operation over the last one. This implementation technique of 1D IIR filters could achieve a throughput rate of $1/(p+q)(t_m+t_a)$ much higher than the throughput rate of $1/(2pq)(t_m+t_a)$ of the Kung-type systolic array of Fig. 1.

At the last step, a separate collection network may be used to pipe out the V_{ij} results. Thus, the states $x_i(n+1)$, $1 \leq i \leq N$, of the filter together with the next input sample $e(n)$ will be used as direct entries of the array for the next wave front.

In the last discussion, the ability to dynamically switch and reconfigure the array implies added hardware complexity. These hardware complexity need careful evaluation in any specific design process.

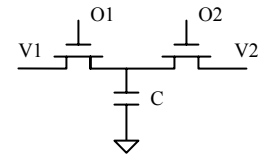
4 Design of processing elements by using switched-capacitor architectures

In this paper, we propose the use of switched-capacitor structures [13-17] to build the PEs of the systolic architectures. These last structures are mainly based on the switched-capacitor element of figure 3. This basic element can be used to construct adders, multipliers, and delay elements [13-17] which are the basic blocks of all types of processing elements of a systolic array.

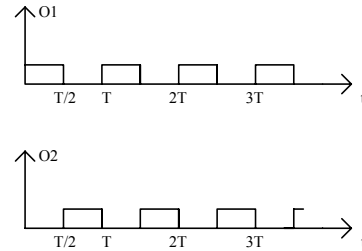
An interesting MOS circuit [13] which performs a function similar to that of a resistor takes advantage of the components and precision available in MOS technologies (Fig. 3a).

Each cylindrical-type PE of the systolic array of figure 2 is built from a Switched-Capacitor Multiplier/Adder, a one time-unit delay, and a memorization component [13-17]. The Switched-Capacitor Multiplier/Adder allows the computation $y_s = y_e + a_{ij}x_e$, the memorization component is used to load the a_{ij} coefficient of the filter during the first wave front, or to store the result $V_{ij} = V_{ij} + (\mathbf{LU})_{ik}r_{kj}$ locally at the PE,

and the one time-unit delay permits the transmission of the vertical input of the PE to its vertical output with one time-unit delay $x_s = x_e$.



(a) SC circuit

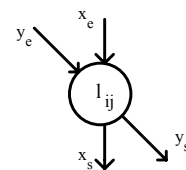


(b) Switch timing

Fig. 3 The Basic switched-capacitor element.

5 Conclusion

In this paper, we have presented and analyzed systolic architectures of the cylindrical-type that can be used to realize sampled-data recursive filters. All these structures are obtained in a straightforward manner from a matrix representation of the filters in the state-space domain. We have noticed in a previous work that a latency proportional to the filter order is the main disadvantage of the Kung-type systolic architectures. We have shown that the use of CTP technique together with the cylindrical structures leads to an improvement of computation throughput of these systolic arrays. Switched-capacitor techniques are proposed, in this paper, to build processing elements used in these structures.



(a) Operation of the cylindrical-type PEs
At the first wave front:

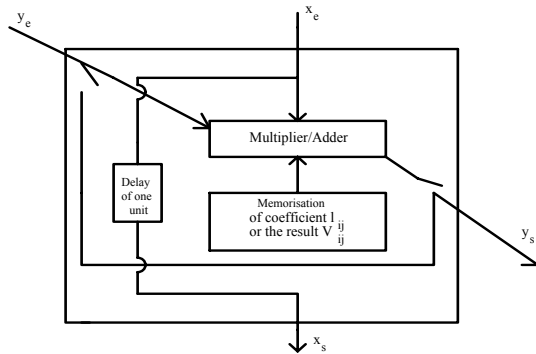
$$y_s = y_e + l_{ij} x_e$$

$$x_s = x_e \text{ (Delay of one time unit)}$$

At the second wave front:

$$\left. \begin{array}{l} y_s = y_e \\ x_s = x_e \end{array} \right\} \text{ (Delay of one time unit)}$$

$$V_{ij} = V_{ij} + (LU)_{ik} r_{kj}$$



(b) PE's Construction of the cylindrical-type
Fig. 4 PE's Construction of the cylindrical-type using SC techniques

References:

- [1] H. T. Kung, "Why systolic architectures?", *IEEE Computer*, Vol. 15, N°1, 1982, pp 37-46.
- [2] S. Y. Kung, K. S. Arun, R. J. Gal-Ezer, D. V. Bhaskar Rao, «Wavefront array processor: language, architecture, and applications», *IEEE Trans. comput., Special Issue on parallel and distributed computers*, Vol. C-31, N° 11, Nov. 1982, pp. 1054-1066.
- [3] W. A. Porter, J. L. Aravena, "Orbital architectures with dynamic reconfiguration", *Proc.IEE, part E*, Vol. 134, N°6, Nov.1987, pp. 281-287.
- [4] S. Jain, L. Song, K. K. Parhi, "Efficient semi-systolic VLSI architectures for finite field arithmetic", *IEEE Trans. On VLSI Systems*, Vol. 6, N° 1, Mar. 1998, pp. 101-113.
- [5] A. Härmä, "Implementation of frequency-warped recursive filters", *Signal Processing*, Vol. 80, 2000, pp. 543-548.
- [6] K. Z. Pekmestzi, N. K. Moshopoulos, "A bit-interleaved systolic architecture for a high-speed RSA system", *Integration : the VLSI Journal*, Vol. 30, N° 2, 2001, pp. 169-175.
- [7] D. Chikouche, R. E. Bekka, "Cylindrical architectures for 1-D recursive digital filters: a state space approach", *IEE Proc.-Comput. Digit. Tech.*, Vol. 145, No. 4, July 1998, pp.1-6.
- [8] D. Chikouche, R. E. Bekka, "Architectures rapides dynamiquement reconfigurables des filtres numériques récurrents 1-D et 2-D ", *Revue Traitement du signal*, vol. 16, N° 1, 1999, pp. 1-12.
- [9] D. Chikouche, R. E. Bekka, A. Boucenna, "Recursive filters using systolic architectures and switched capacitor techniques", *Proc. of the 9th IEEE International Conf. On Electronics, Circuits, and Systems ICECS 2002*, Dubrovnik, Croatia, Sept. 15-18, 2002.
- [10] D. Chikouche, R. E. Bekka, "Systolic architectures for 1D and 2D recursive filters", *Proc. of the 6th African Conference on Research in Computer Science CARI'02*, Yaounde, Cameroon, Oct. 14-17, 2002, pp. 175-182.
- [11] J. P. Ma, K. K. Parhi, E. F. Deprettere, "Pipelining of cordic based IIR digital filters", *Proc. Of IEEE Int. Conf. On Acoustics, Speech and Signal Processing*, Munich, April 1997, pp. 643-646
- [12] C. Souani, M. Abid, K. Torki, R. Tourki, "VLSI design of 1-D DWT architecture with parallel filters", *Integration : the VLSI Journal*, Vol. 29, N° 2, 2000, pp. 181-207.
- [13] K. Martin, A. S. Sedra, "Exact design of switched capacitor bandpass filters using coupled biquad structures", *IEEE Trans. Circuits Syst.*, CAS-27, June 1980, pp. 469-475.
- [14] D. J. Allstot, and W. C. Black, "Technological design considerations for monolithic MOS switched capacitor filtering systems", *Proc. IEEE*, Vol.71, Aug. 1983, pp. 967-986.
- [15] R. Gregorian, K. W. Martin, G. C. Temes, "Switched-Capacitor circuit design", *Proc. IEEE*, Vol.71, Aug. 1983, pp. 941-966.
- [16] D. Brodarac, D. Herbst, B. J. Hosticka, B. Hoefflinger, "A novel sampled-data MOS multiplier", *Electron. Lett.*, Vol. 18, 1982, pp. 229-230.
- [17] E. Kettel, W. Schneider, "An accurate analog multiplier and divider", *IRE Trans. Electronic Computers*, Vol. ED-7, 1961, pp. 269-274.