論文 94-19-12-10

Pipelining of Orthogonal Double-Rotation Digital Lattice Filters for High-Speed and Low-Power Implementation

Jin-Gyun Chung*, Kyoung-Bae Eum** *Regular Members*

고속 및 저파워 실현을 위한 직교 이중 회전 디지탈 격자 필터의 파이프라인화

正會員 鄭 鎭 均* 正會員 嚴 景 培**

要 約

직교 이중회전 디지탈 격자 필터는 고밀도 집적회로 실현에 아주 적합한 성질들을 가지고 있다. 예를 들면, 바로 옆의 프로세서들과만 연결되는 점, 규칙적인 구조를 갖는 점, 그리고 파이프라인화 할 수 있다는 점 등이다. 또한 이 필터들은 유한 워드 길이로 실현했을 때 좋은 수치적 성질을 갖는다는 것이 잘 알려져 있다. 비록 이 필터들은 "컷셋 분리 절차"를 이용하여 파이프라인화 될 수 있지만 이렇게 파이프라인화 된 필터는 귀환회로의 계산 시간에 의해 이 필터의 최대 데이타 처리속도가 제한된다는 단점이 있다.

본 논문에서는 귀환회로에 제한 받지 않고 원하는 만큼 데이타 속도를 높이거나 저파워 실현을 위한 직교 이중회전 디지탈 격자 필터의 새로운 파이프라인 방법을 제안하였다. 이 방법은 Schur 알고리즘, 필터 합성시 특정한 제약을 주는 필터 합성 방법, 그리고 다중 페이스 분해 방법에 근거하고 있다.

ABSTRACT

The ODR(orthogonal double-rotation) digital lattice filters have desirable properties for VLSI implementation such as local connection, regularity and pipelinability. These filters are also known to exhibit good numerical behavior for finite precision implementation. Although these filters can be pipelined by the cut-set localization procedure, it should be noted that the maximum sample rate obtained by this technique is limited by the feedback computations.

In this paper, a pipelining method for the ODR digital lattice filter is proposed, by which the sample rate can be increased at any desired level. It is also shown that the low-power CMOS digital implementation of ODR digital lattice filters can be done successfully using our pipelining method. The pipelining method is based on the properties of the Schur algoithm, constrained filter design methods, and the polyphase decomposition technique.

*미국 미네소타 주립대학교 전기공학과
**군산대학교 컴퓨터과학과

論文番號 : 94174
接受日字 : 1994年 7月 1日

I. Introduction

Any given digital filter transfer function can be realized in an infinite number of ways theoretically. From the implementation aspect, some structures may have lower complexity, while others may be pipelinable, and yet some others may consist of regular modules that can be designed in less time, etc. Much research has been carried out to search for different realization structures with various desirable properties and enhanced performance. Some major work includes *wave digital filters* [1], *basic and normalized lattice filters* [2-3], *scaled normalized lattice filters* [4-5], and *ODR digital lattice filters* [6].

The ODR digital lattice filters were developed for the realization of any stable, passive digital rational transfer function in a cascaded interconnection of orthogonal sections. Therefore, these filters possess desirable properties for VLSI implementation such as local connection, regularity and pipelinability. Each section of these fileters is realized involving only Givens rotations and storage elements, which enables the translation of the sensitivity arguments for analog lossless ladder realizations to the ODR digital lattice filters.

Achieving high speed in recursive filters is difficult because of the feedback loops. Although these filters can be pipelined by the cut-set localization procedure in [7], it should be noted that the maximum sample rate of these pipelined filters is limited by the feedback loop computations. A linear array is pipelinable when either all the left-directed or all the right-directed edges between modules carry at least one delay on each edge. If this condition is satisfied, the cut-set localization procedure can be applied to transfer some delays or a fraction of a delay to the opposite directed edges. For example, in the ODR digital lattice filter in Fig.1, the cut-set localization procedure can be applied to transfer one half of each delay on the right directed edges to the left directed edges. Then the half delays can be implemented by time rescaling. For example, using one clock cycle to represent a half delay, we can input data once every two clock cycles and generate the output data once every two clock cycles. With this transformation, the clock speed can be increased, but the sample rate cannot be increased, since multiple clock cycles are needed to process one sample. The maximum sample rate of this structure is limited by the feedback loop computation which involves three multiplications and three additions.

To increase the sample rate of IIR direct form filters at any desired level, the scattered look-ahead technique was proposed in [8]. Pipelining methods for basic lattice, normalized lattice, and scaled normalized lattice filters were proposed in [9-10]. In this paper, a pipelining method for the ODR digital lattice filters is proposed using the properties of the Schur algorithm [11], constrained filter design methods [9, 12-13], and the polyphase decomposition technique [14]. The pipelined filters can also be used for low-power applications [15].

II. Synthesis of ODR digital lattice filters

Consider an N-th order stable, passive IIR transfer function given by $H(z)=N_N(z)/D_N(z)$, where

$$N_N(z)=\sum_{i=0}^{N} n_i z^i, \text{ and } D_N(z)=\sum_{i=0}^{N} d_i z^i. \quad (1)$$

Then, the filter synthesis steps can be briefly summarized as follows [6]:

Step 1) For any N-th order stable, passive IIR transfer function, find $E_N(z)$ which satisfies the following relation:

$$E_N(z)^* E_N(z)=D_N(z)^* D_N(z)-N_N(z)^* N_N(z). \quad (2)$$

The superscript(*) operation is defined as

$$A(z)^*=A'(1/z'), \quad (3)$$

where ($'$) represents complex conjugate transpo-

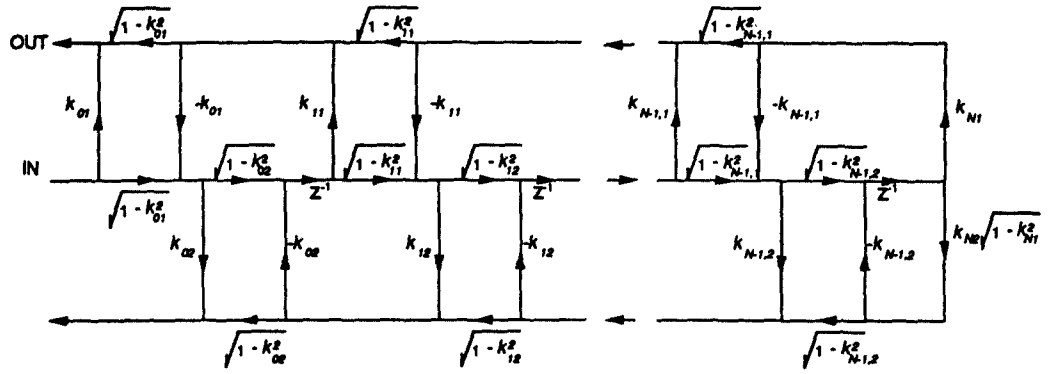


Fig. 1 Orthogonal double-rotation digital lattice filter structure

sition.

Step 2) Form $P_N(z) = \begin{bmatrix} N_N(z) \\ E_N(z) \end{bmatrix}$. (4)

The k-parameters for the first section are computed as follows :

$$k_0 = P_N(\infty) / D_N(\infty), \tag{5}$$

where $k_0 = \begin{bmatrix} k_{01} \\ k_{02} \end{bmatrix}$ (6)

Step 3) The new numerator and denominator polynomials are computed as follows :

$$D_{N-1}(z) = z^{-1}(1-k_0'k_0)^{-1/2}\,(D_N(z) - k_0'P_N(z)),$$

$$P_{N-1}(z) = (I - k_0k_0')^{-1/2}\,(P_N(z) - k_0D_N(z)). \tag{7}$$

Steps 2 and 3 are repeated until $N+1$ sets of k-parameters are obtained. Using the k-parameters obtained in step 2, the filter can be realized as in Fig. 1.

Notice that steps 2 and 3 are just a particular version of the Schur algorithm, which can be rewritten as follows :

Step 2-1)

$$\begin{bmatrix} D_N(z)^+ \\ N_{N-1}(z) \end{bmatrix} = \frac{1}{\sqrt{1-k_{01}^2}} \begin{bmatrix} 1 & -k_{01} \\ -k_{01} & 1 \end{bmatrix} \cdot \begin{bmatrix} N_N(z) \\ E_N(z) \end{bmatrix}, \tag{8}$$

where $k_{01} = N_N(\infty) / D_N(\infty)$.

Step 2-2)

$$\begin{bmatrix} zD_{N-1}(z) \\ E_{N-1}(z) \end{bmatrix} = \frac{1}{\sqrt{1-k_{02}^2}} \begin{bmatrix} 1 & -k_{02} \\ -k_{02} & 1 \end{bmatrix} \cdot \begin{bmatrix} D_N(z)^+ \\ E_N(z) \end{bmatrix} \tag{9}$$

where $k_{02} = E_N(\infty) / D_N(\infty)^+$.

The subscript of each polynomial denotes the degree of the polynomial. Therefore, through the steps 2-1 and 2-2, the degrees of $D_N(z)$, $N_N(z)$, and $E_N(z)$ are reduced by 1, respectively.

Ⅲ. Pipelining of ODR digital lattice filters

In the filter structure of Fig. 1, if k-parameters of section i(*i.e.*, k_{i1} and k_{i2}) are zero, then the feedback loops through the section are removed. If k-parameters of every odd numbered section are zero, then every feedback loop through odd numbered sections is removed, which means that every feedback loop has one more delay available. By redistributing the additional delays at the proper locations, we can achieve 2-level pipelining. This approach can be extended for general M-level pipelining. The pipelining approach in this section is based on this observation.

Let's denote the N-th order denominator in (1)

as follows :

$$D_N(z) = [\, d_N \;\; d_{N-1} \;\; d_{N-2} \;\; \cdots \;\; d_1 \;\; d_0 \,]. \qquad (10)$$

$N_N(z)$ and $E_N(z)$ can also be represented by the same way.

From step 2-1,

$$N_{N-1}(z) = \frac{[\, N_N(z) - k_{01} D_N(z) \,]}{\sqrt{1 - k_{01}^2}} \quad . \qquad (11)$$

With $k_{01} = n_N/d_N$, $N_{N-1}(z)$ can be computed as

$$\frac{[\, n_N d_N - n_N d_N \;\; n_{N-1} d_N - n_N d_{N-1} \ldots n_0 d_N - n_N d_0 \,]}{\sqrt{d_N^2 - n_N^2}} \quad .$$

It may be noted from the vector notation of $N_{N-1}(z)$ that the coefficients of $N_{N-1}(z)$ can be computed by $\frac{1}{\sqrt{d_N^2 - n_N^2}}$ times the $N+1$ determinants of 2×2 submatrices formed by the first column and each succeeding column in the following matrix :

$$\begin{bmatrix} d_N & d_{N-1} & d_{N-2} & \cdots\cdots & d_2 & d_1 & d_0 \\ n_N & n_{N-1} & n_{N-2} & \cdots\cdots & n_2 & n_1 & n_0 \end{bmatrix}.$$

The determinant of the first submatrix is always zero since the submatrix is formed by repeating the first column twice, which is obviously a dependent matrix. Therefore, the degree of $N_{N-1}(z)$ is reduced by 1 compared with that of $N_N(z)$. For each 2×2 submatrix, if a column is composed of all zero elements, the determinant of the submatrix is zero. Therefore, if $D_N(z)$ and $N_N(z)$ have j-consecutive zero coefficients between each two nonzero coefficients of nearest degree, then $N_{N-1}(z)$ is forced to have the same property. The coefficients of $D_N(z)^+$ in (8) can be computed from the same submatrices as :

i-th coefficient of

$$D_N(Z)^+ = \frac{1}{\sqrt{d_N^2 - n_N^2}} \cdot [\, \textit{product of the elements in the 1st row of i-th submatrix} - \textit{product of the elements in the 2nd row of i-th submatrix} \,]. \qquad (12)$$

Therefore, if $D_N(z)$ and $N_N(z)$ have j-consecutive zero coefficients between each two nonzero coefficients of nearest degree, then $D_N(z)^+$ is forced to have the same property. By (12), the coefficient of the highest degree term of $D_N(z)^+$ (i.e., $i=1$), and the constant term can be computed as follows :

$$d_N^+ = \sqrt{d_N^2 - n_N^2} \quad , \qquad (13)$$

$$d_0^+ = \frac{d_0 d_N - n_0 n_N}{\sqrt{d_N^2 - n_N^2}} \quad . \qquad (14)$$

From (13) and (14), it is obvious that the degree of $D_N(z)^+$ is not reduced by the step 2-1.

In step 2-2, with $k_{02} = e_N/d_N^+$, $E_{N-1}(z)$ and $zD_{N-1}(z)$ are computed by the same way as $N_{N-1}(z)$ and $D_N(z)^+$ are computed. Therefore, the degree of $E_{N-1}(z)$ is reduced by 1 compared with that of $E_N(z)$. On the other hand, the constant term of $zD_{N-1}(z)$ can be computed as $d_0 d_N - n_0 n_N - e_0 e_N / \sqrt{d_N^2 - n_N^2 - e_N^2}$.

The constant term of $zD_{N-1}(z)$ is always zero since, by (2), $E_N(z)$ satisfies the following condition :

$$e_0 e_N = d_0 d_N - n_0 n_N \, . \qquad (15)$$

Therefore, the degree of $D_{N-1}(z)$ is reduced by 1 compared with that of $D_N(z)$ through the steps 2-1 and 2-2. Above observations are summarized by the following theorem which is crucial for the pipelining of ODR digital lattice filters :

Theorem 1 : Let's assume the conditions of step 1 of section 2 are satisfied by N-th order polynomials $D_N(z)$, $N_N(z)$ and $E_N(z)$. If $D_N(z)$, $N_N(z)$ and $E_N(z)$ have j-consecutive zero coefficients between each two nonzero coefficients of nearest degree, then $D_{N-1}(z)$, $N_{N-1}(z)$ and $E_{N-1}(z)$ also have j-consecutive zero coefficients between each two

nonzero coefficients of nearest degree when those polynomials are obtained by the steps 2 through 3 of section 2.

If $D_N(z)$, $N_N(z)$ and $E_N(z)$ satisfy the *j-consecutive zero condition* of Theorem 1, it can be shown using step 2 that the k-parameters of consecutive j-sections become zero. The polynomials satisfying the j-consecutive zero condition can be obtained by the following procedure :

1) Given filter specifications and pipelining level M, design the filter using any constrained filter design method [8-9, 12-13], which constrains the denominator to be a polynomial in z^M rather than in z. Therefore, the transfer function can be represented by $H(z)=N(z)/D(z^M)$.

2) Decompose the obtained transfer function $H(z)$ using polyphase decomposition techinque :

$$H(z)=\sum_{i=0}^{M-1}\frac{z^i N^{(i)}(z)}{D(z^M)}, \tag{16}$$

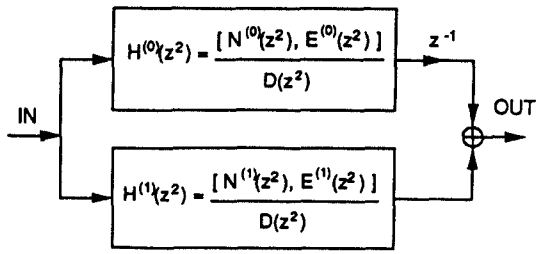


Fig. 2 Overall scheme of 2-level pipelined orthogonal double-rotation digital lattice filter

where $N^{(i)}(z)=\sum_{j=0}^{P} n_{i+jM}\cdot z^{jM}$, (17)

and P is an integer which satisfies the following condition :

$$N-M<i+PM\leq N. \tag{18}$$

3) For $D(z^M)$ and each $N^{(i)}(z)$, compute $E^{(i)}(z)$ which satisfies (2). Note that $E^{(i)}(z)$ is also a polynomial in z^M.

4) Since $\{D(z^M), N^{(i)}(z), E^{(i)}(z)\}$ satisfies the conditions of Theorem 1, it can be synthesized as an M-level pipelined ODR digital lattice filter. The final output is obtained by adding the outputs from each set $H^{(i)}(z)=\{D(z^M), N^{(i)}(z), E^{(i)}(z)\}$, for $i=0$ to $M-1$, by (16). For example, Fig. 2 shows the overall scheme for $M=2$.

Example 1 :

The design example in [6] is used for the synthesis of 2-level pipelined ODR digital lattice filter. A pipelinable transfer function which satisfies the conditions of Theorem 1 is synthesized using the constrained filter design method in [9] with $M=2$. The obtained transfer function is as follows :

$D(z^2)=[1\ 0\ -1.7399\ 0\ 1.2893\ 0\ -0.3468]$,
$N(z)=[0.0322\ 0.0623\ 0.0128\ -0.0174\ 0.0372\ 0.0564\ 0.0189]$

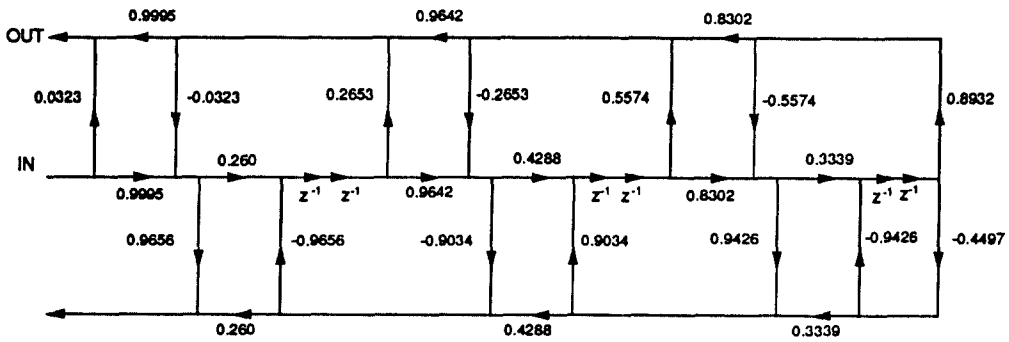


Fig. 3 Orthogonal double-rotation digital lattice filter implementation of $H^{(0)}(z)$ in Ex.1 (M=2)

The transfer function $H(z)$ is decomposed by the polyphase decomposition technique in (16). Since $M=2$, we have two sets of decomposed transfer functions :

$$\begin{bmatrix} N^{(0)}(z) \\ E^{(0)}(z) \end{bmatrix} = \begin{bmatrix} 0.0322 & 0 & 0.0128 & 0 & 0.0372 & 0 & 0.0189 \\ 0.9650 & 0 & -1.7402 & 0 & 1.3106 & 0 & -0.3598 \end{bmatrix},$$

$$\begin{bmatrix} N^{(1)}(z) \\ E^{(1)}(z) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.0623 & 0 & -0.0174 & 0 & 0.0564 \\ 0.9645 & 0 & -1.7396 & 0 & 1.3103 & 0 & 0.3594 \end{bmatrix}.$$

The k-parameters for $H^{(0)}(z)$ and $H^{(1)}(z)$ are computed by iterative applications of the steps 2-1 and 2-2 as follows :

-k-parameters for $H^{(0)}(z)$:

$$\begin{bmatrix} 0.0323 & 0 & 0.2653 & 0 & 0.5574 & 0 & 0.8932 \\ 0.9656 & 0 & -0.9034 & 0 & 0.9426 & 0 & -1 \end{bmatrix}.$$

-k-parameters for $H^{(1)}(z)$:

$$\begin{bmatrix} 0 & 0 & 0.2356 & 0 & 0.3617 & 0 & 0.9180 \\ 0.9645 & 0 & -0.9072 & 0 & 0.8745 & 0 & -1 \end{bmatrix}.$$

The first row of each k-parameter matrix corresponds to k_{i1}'s and the second row corresponds to k_{i2}'s. Column i of each k-parameter matrix corresponds to the *k-parameters* of i-th section. Notice that the k-parameters of alternate sections are zero. Fig.3 shows the implementation of the ODR digital lattice filter corresponding to $H^{(0)}(z)$. In a similar manner, $H^{(1)}(z)$ can be implemented and those two implementations are interconnected by the scheme as shown in Fig.2, which completes the filter design process. ◆

The process of retiming involves moving around the delays in a circuit such that the total number of delays in any loop remains unaltered, and the input-output behavior of the system is preserved. Removal of a fixed number of delays from each of the incoming edges of any node, and addition of the same fixed number of delays to each of the outgoing edges of the same node is a basic retiming operation. An example is shown in Fig.4.

By repeatedly applying the basic retiming operation, delays can be moved to the desired locations such that the critical path of the circuit is minimized. In Fig.2, there are two delays in each loop and these delays can be distributed such that the critical path is halved as compared with that of the nonpipelined filter. This results in an increase of clock speed by a factor of two.

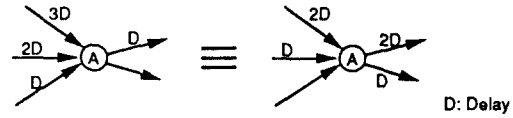


Fig. 4 The delays on the incoming edges are reduced by one, while the delays on the outgoing edges are increased by one.

To apply the polyphase decomposition technique, the denominator of a transfer function should be a polynomial in z^M. The transfer functions satisfying this condition can be obtained by applying the scattered look-ahead method to a transfer function whose denominator is a polynomial in z. To avoid the drawback of canceling zeros in the scattered look-ahead method, the pipelinable transfer function can be designed directly from the filter spectrum while the denominator is constrained to be a polynomial in z^M rather than in z. The constrained filter design method used in Example 1 is called the *modified Deczky's method*. In [9-10], it is shown that the transfer function designed by the modified Deczky's method requires less hardware than the scattered look-ahead method.

The modified Deczky's method first expresses the magnitude and group delay responses of a filter as functions of the radii and angles of the poles and zeros. Then the formulae for the partial derivatives of the magnitude and group delay are obtained with respect to the radius and the angle of a pole and a zero. These derivatives are used in the Fletcher-Powell algorithm to minimize the approximation error. To obtain a denominator in

z^M, the partial derivatives for a denominator are recomputed in powers of z^M and the equations are used in the Fletcher-Powell algorithm. Then, the denominator of the resulting transfer function is in terms of z^M.

Ⅳ. Low-power implementation of pipelined ODR digital lattice filters

The issue of low-power design is of great concern, particularly in high performance portable applications. Furthermore, as the density and size of the chips and systems continue to increase, the difficulty in providing adequate cooling might either add significant cost to the system or provide a limit on the amount of functionality that can be provided. In addition, many computation tasks currently require real-time processing. Once the real-time requirements are met, there is no advantage in increasing the computational throughput. This fact, along with pipelining, can be used to provide significant power savings in CMOS digital designs.

The power dissipation in a well-designed digital CMOS circuit can be approximated as

$$P=C_t V_{dd}^2 f_{clk}, \tag{19}$$

where C_t is the total switching capacitance, V_{dd} is the supply voltage and f_{clk} is the clock frequency. Due to the quadratic relationship of V_{dd} to the power consumption, reducing the supply voltage is clearly the key to the low-power operation even after taking into account the modifications to the system architecture. If V_{dd} is reduced, we should pay a speed penalty (i.e., increase in the propagation delay T_o) as can be seen from the following equation:

$$T_o=\frac{C_l V_{dd}}{k(V_{dd}-V_t)^2}, \tag{20}$$

where C_l is the capacitance along the critical path in the circuit, and V_t is the device threshold voltage and k is a process dependent parameter. Notice that T_o increases dramatically as V_{dd} approaches V_t. Also, the threahold voltage should be chosen properly by the requirement to retain adequate noise margins and the increase in the subthreshold currents.

For an M level pipelined system, the propagation delay T_p is

$$T_p=\frac{C_l}{M}\frac{V_{dd}}{k(V_{dd}-V_t)^2}. \tag{21}$$

Clearly, the pipelined system can be clocked M times faster than is necessary since the capacitance along the critical path has been reduced by M times compared with the nonpipelined system. Therefore, the supply votage can be reduced until T_p equals T_o. Since the pipelined system can be operated at a reduced supply voltage and at the same speed of the nonpipelined system, the pipelined system can achieve dramatic reductions in power consumption. In the following example, it is shown that the low-power CMOS digital implementation of ODR digital lattice filter can be done successfully using this pipelining method.

Example 2:

Consider the 2-level pipelined ODR lattice filter in Example 1. In this example, the additional delays are used only for low-power implementation. Therefore, the sample rate of the pipelined system is remained the same as that of the original system ($M=1$). Assume that the capacitance due to multipliers is dominant and that the capacitance due to the adders can be neglected. Also, assume the supply voltage V_o of the original system to be 5V and the CMOS threshold voltage to be 0.5V. Then, from (20) and (21), the supply voltage for the pipelined system, V_P, can be reduced to 2.94V. Therefore,

$$power\ saving=\left(\frac{m_P}{m_o}\right)\left(\frac{V_P}{V_o}\right)^2=0.638,$$

where $m_p=48$ and $m_o=26$ are the number of multipliers for the pipelined system and the original system, respectively. Therefore, the power con-

sumption of the 2-level pipelined system is only 63.8% of the original system. ◆

A direct implementation of the ODR digital lattice filters seems to indicate a formidable increase in hardware complexity when compared with conventional realizations. An efficient realization scheme of the ODR digital lattice filters was proposed in [6] using the CORDIC algorithm. A pipelined ODR digital lattice filter is composed of only Givens rotations and pure delays. Therefore, the CORDIC algorithm can also be used for implementing a pipelined ODR digital lattice filters.

V. Conclusions

It was shown that the ODR digital lattice filters can be pipelined at any desired level by the proposed pipelining method. Since the pipelined filters are composed of cascade connections of orthogonal sections, they exhibit good finite word-length properties. Due to these finite word-length properties, the pipelined ODR filters can be implemented using a smaller word-length than the conventional cascade or parallel form filters. The pipelined filters can be used for high sample rate applications or for low-power CMOS implementation applications.

The sample rate increase or the low-power implementations are obtained with the cost of increased hardware. Threrfore, future work might be to investigate a hardware minimization method for this proposed pipelining method.

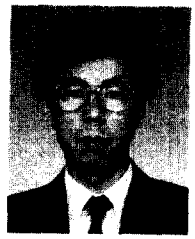
References

1. A. Fettweis, "Digital filters related to classical filter networks," *Arch. Electr* vol. 25, pp.79-89, Feb. 1971.
2. A. H. Gray, Jr. and J. D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. on Audio and Electroacoustics*, pp.491-500, Dec. 1973.
3. A. H. Gray, Jr. and J. D. Markel, "A normalized digital filter," *IEEE Trans. on speech, and signal processing*, vol. ASSP-33, pp.258-277, June 1975.
4. S. M. Lei and K. Yao, "A class of systolizable IIR digital filters and its design for scaling and minimum output roundoff noise," *IEEE Trans. on Circuits and systems*, pp.1217-1230, Oct. 1990.
5. J. G. Chung and K. K. Parhi, "The scaled normalized lattice digital filter," *Proceedings of 1993 IEEE ISCAS*, pp.483-486, Chicago (IL), May 1993.
6. S. K. Rao and T. Kailath, "Orthogonal digital filters for VLSI implementation," *IEEE Trans. on Circuits and systems*, pp.933-945, Nov. 1984.
7. S. Y. Kung, "On super computing with systolic/wavefront array processors," *Proc. IEEE*, vol. 72, pp.867-884, July 1984.
8. K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters-part I : pipelining using scattered look-ahead and decomposition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, pp.1099-1117, July 1989.
9. J. G. Chung and K. K. Parhi, "Design of pipelined lattice IIR digital filters," *Proceedings of the 25th Asilomar conference on Signals, Systems, and Computers*, pp.1021-1025, Nov. 4-6 1991, Pacific Grove(CA)
10. J. G. Chung and K. K. Parhi, "Pipelining of lattice IIR digital filters," *IEEE Transactions on Signal Processing*, Vol.42, pp.751-761, April 1994.
11. J. Schur, "Ueber Potenzreihen die im Innern des Einheitslreises beschrankt sind," *J. Reine Angewandte Mathematik*, vol.147, pp.205-232, Berlin, 1917.
12. M. A. Richards, "Application of Deczky's program for recursive filter design to the design of recursive decimators," *IEEE Trans.* on ASSP, Oct. 1982.
13. H. G. Martinez and T. W. Parks, "A class of infinite-duration impulse response digital filters for sampling rate reduction," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, pp.154-162, Apr. 1979.
14. M. Bellanger, G. Bonnerot and M. Coudreuse.

"Digital filtering by polyphase network : application to sample rate alteration and filter banks," *IEEE Trans. on Acoustics, Speech, and Signal Processing, vol.* ASSP-24, pp.109-114, Apr. 1976.

15. A. P. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of solid state circuits,* vol. 27, pp. 473-484, Apr. 1992.

鄭 鎭 均(Jin-Gyun Chung) 정회원
1985년 2월 : 全北 大學校 工科大學 電子工學科 卒業
1984년 12월~1988년 1월 : (주) 大宇 通信 研究員
1991년 12월 : 美國 미네소타 州立 大學校 大學院 電氣工學科 卒業(工學 碩士)
1994년 12월 : 美國 미네소타 州立 大學校 大學院 電氣工學科 博士科程 卒業 豫定



嚴 景 培(Kyoung-Bae Eum) 정회원
1984년 2월 : 全北 大學校 工科大學 電子工學科 卒業
1986년 2월 : 全北 大學校 大學院 電子工學科 卒業(工學 碩士)
1990년 8월 : 全北 大學校 大學院 電子工學科 卒業(工學 博士)
1989년 3월~1991년 3월 : 群山大學 電算 統計學科 專任講師
1991년 4월~1994년 12월 : 群山大學校 컴퓨터 科學科 助教授