



i

ſ

MICROCOPY RESOLUTION TEST CHART NATIONAL BUREAU OF STANDARDS-1963-A Jun Diego

march 1984

PIPELINED ORTHOGONAL DIGITAL LATTICE FILTERS

Sailesh K. Rao and Thomas Kailath Information Systems Laboratory Stanford University Stanford, CA 94305.



- (4) Communication: Communication between the processing elements should be restricted to immediate neighbours. A circuit that is modular and has only nearest-neighbour links is said to be systolic [2].
- (5) Numerical Stability: There are certain undesirable phenomena. e.g., limit-cycle and overflow oscillations, associated with the finite-precision implementation of linear systems that have been well-studied in the literature[3]. The circuit should be free of them. It should also be insensitive to slight variations in the multiplicative parameters in the implementation.

In the next section, a systolic array configuration is proposed for the circuit that, in general, satisfies properties (1),(3),(4). The pipelineability of such an array is considered and it will be shown that every strictly stable transfer function can be realized in several different ways that ensure that all the above desirable criteria are met.

2. BI-DIRECTIONAL LINEAR SYSTOLIC ARRAYS

A systolic array is said to be bi-directional if every processing element receives input from and provides output to each of its neighbours. (The external environment is a neighbour to the processing elements at the corners of the array.) The array is said to be linear in operation if each processing element performs only linear operations on its inputs. It is said to be linear in configuration if each processing element has only two neighbours. A systolic array that is linear both in operation and in configuration is linear. Such an array is time-invariant if the z-transform of the vector output sequence, $Y_i(z)$, of its ith processing element is related to the ztransform of its corresponding vector input sequence, $U_i(z)$, by:

$$Y_i(z) = \Sigma_i(z)U_i(z)$$
 (2)

where $\Sigma_i(z)$ is some rational matrix function of z, necessarily square, such that $\Sigma_i(\infty)$ has only finite elements, in order to ensure causality. A linear systolic array is said to have a *ternary signature* if it has either (a) two wires flowing from right to left and one from left to right, or (b) vice-versa. (See Fig.1)

The intent of this paper is to present a general framework for designing such arrays so that when they are suitably terminated at both ends with constant multiplier interconnections, they are realizations for H(z). The reasons for choosing ternary signature arrays as opposed to binary signature arrays are the following:

A pipelined orthogonal realization of H(z) in a binary signature array requires, in general, at least 2π delay elements and $(3\pi + 1)$ elementary Givens rotational

Approved for public releasest distribution unlimited.

ABSTRACT

AFOSR-TR- 84-0518

A .

An algorithm is presented for the design of systolic arrays that implement single-input single-output timeinvariant digital filters. The algorithm is then specialized to the case where the realized array consists only of orthogonal rotational modules and delay elements interconnected in such a manner as to render the circuit pipelineable.

1. INTRODUCTION

Consider a single-input single-output linear timeinvariant discrete-time system that allows a transfer function description:

$$H(z) = \frac{\sum_{i=0}^{n} N_i z^{n-i}}{z^n + \sum_{i=0}^{n} D_i z^{n-i}} = \frac{N(z)}{D(z)}$$
(1)

The system is said to be strictly stable if the function, H(z), in the complex variable, z, is analytic on the unit circle, Π , and in the region, E, where, |z| > 1. In other words, the zeros of the polynomial, D(z) should lie in the region D, where, |z| < 1. A strictly stable function is said to be a Schur function if it has magnitude less than or equal to unity on Π .

In this paper, the transpose of a matrix is denoted by the superscript, T, the complex conjugate transpose by an over-bar, and the para-Hermitian conjugate by the subscript \bullet . The para-Hermitian conjugate of a matrix function is, of course, defined as:

$$A_{\bullet}(z) = \bar{A}(1/\bar{z})$$

Given a transfer function, H(z), the synthesis problem is to obtain a hardware implementation of H(z). The quality of such an implementation using VLSI technology is then dependent upon the following criteria:

- Modularity: The circuit should preferably be a regular interconnection of similar processing elements.
- (2) Pipelineability: The maximum throughput achievable should be independent of the number of processing elements, n, in the array
- (3) Uniformity. It is desirable to design VLSI circuits that can implement digital filters with a variety of transfer functions of different orders by merely changing certain parameters of the circuit.

This work was supported in part by the Defense Research Projects Agency inder Contract MDA90179-C-0840, by the US Arry Tesearch Office: under Contract DAA9279-C-0840, by the US Arry Tesearch Scientific Research, Air Force Systems Command under Contract AF49 620-79-C-0058 Grant - AF-USK-SB-C.228





modules for its implementation. (See [1]). On the other hand, ternary signature arrays with n delay elements and (2n + 1) elementary Givens rotational modules will be derived in what follows. Another reason is that even if we synthesize the filter in a binary signature array with the attendant increase in hardware requirement, the speed at which the array can process the input sequence is only half the speed that is achievable using ternary signature arrays. Clearly, there is no tradeoff here.

J.

Consider a linear time-invariant systolic array with a ternary signature as shown in Fig. 1. Suppose that for some unknown constant termination at one end of the array. say the right end, the input to output transfer at the left end is given by: (with reference to Fig. 1.)

$$\begin{vmatrix} y_{02}(z) \\ y_{03}(z) \end{vmatrix} = \frac{1}{Q(z)} \begin{bmatrix} p(z) \\ R(z) \end{bmatrix} u_{01}(z) = \varphi(z) u_{01}(z) \quad (3)$$

The function, $\varphi(z)$, will be referred to as the generating function of the systolic array, for reasons that will become clear in what follows. The degree of the function, $\varphi(z)$, is the polynomial-degree of Q(z), assuming that the greatest common divisor of Q(z), P(z), R(z) is constant.

For a given transfer function, H(z), there are several possible choices of $\varphi(z)$. Suppose that the desig or wishes to use

$$u(z) = k_1 u_{01}(z) + k_2 y_{02}(z) \qquad (4a)$$

$$y(z) = y_{03}(z)$$
 (4b)

such that the transfer function from u(z) to y(z) is H(z). Then, $\varphi(z)$ should be such that P(z),Q(z),R(z) are polynomials satisfying the relationship:

$$R(z) = N(z) \tag{5a}$$

$$k_1Q(z) + k_2P(z) = D(z)$$
 (5b)

Clearly, the choice of P(z) and Q(z) is not unique. Moreover, the designer might wish to use different relationships in eqn.(4) in order to obtain the desired input and output terminals and the nonuniqueness is thus further enhanced.

We will get back to this problem of the choice of the generating function later. For the present, assume that $\varphi(z)$ is known.

The realization procedure, then, depends upon the solution to the following problem:

Given the (2×1) vector generating function, $\varphi(z)$, of degree, n, obtain a linear time-invariant systolic array with n processing elements and the constant termination at the right end of the array in Fig.1, such that the input to output transfer at the left

end of the array is $\varphi(z)$.

The following algorithm solves the above problem:

The Systolic Realization Algorithm:

It is required that the systolic array in Fig.1 with the i^{th} processing element defined by:

$$y_{i1} y_{i2} y_{i3} \Big|^{T} = \Sigma_{i}(z) \Big[u_{i1} u_{i2} u_{i3} \Big]^{T}$$
 (6)

is such that when the right end of the array is terminated with

$$\left(n-1 \right) \cdot 2 \left[u_{(n-1),3} \right]^T = \varphi_n(z) y_{(n-1),1}$$
 (7)

the signals at the left end of the array are related by:

$$\begin{bmatrix} y_{02} & y_{03} \end{bmatrix}^T = \varphi(z) u_{01}(z)$$
(8)

and that $\varphi_n(z)$ in eqn.(7) is a constant vector.

Step 1: Initialization: Let $\varphi_{c}(z) = \varphi(z)$.

For i = 0, 1, ..., n - 1, do

þ

þ

Step 2: Choose the complex scalar constants, α_i, β_i , the (2×2) nonsingular constant matrix, C_i , and the (1×2) constant vector, v_i^T , and obtain (we will restrict these choices later in order to ensure that the array is pipelineable and orthogonal. For the present, assume that they are free parameters subject to the designer's fancy and the equations below.)

$$\varphi_{i+1}(z) = \frac{C_i[\varphi_i(z) - \varphi_i(\beta_i)]}{v_i^{T}[\varphi_i(z) - \varphi_i(\alpha_i)]} p_i(z)$$
(9a)

such that the (3×3) constant matrix, Θ_{i} , defined by:

$$\Theta_{i} = \begin{bmatrix} -\nu_{i}^{T}\varphi_{i}(\alpha_{i}) & \nu_{i}^{T} \\ -C_{i}\varphi_{i}(\beta_{i}) & C_{i} \end{bmatrix}$$
(9b)

is nonsingular,

and

$$\varphi_i(\alpha_i), \varphi_i(\beta_i) \neq =$$
 (9d)

and $p_1(z)$ is chosen to be:

$$p_i(z) = \frac{z - a_i}{z - \beta_i}, \quad a_i, \beta_i \neq =, \qquad (9e)$$

$$p_i(z) = z - \alpha_i, \quad \beta_i = \cdots, \alpha_i \neq \cdots$$
 (9f)

Step 3: Form

$$E_{i}(z) = \begin{bmatrix} p_{i}^{-1}(z) & 0\\ 0 & I \end{bmatrix} \begin{bmatrix} -\nu_{i}^{T}(\varphi_{i}(\alpha_{i}) - \varphi_{i}(\beta_{i})) & \nu_{i}^{T}C_{i}^{-1}\\ \varphi_{i}(\beta_{i}) & C_{i}^{-1} \end{bmatrix} (10)$$

Comments: The constraint (9d) imposed in step 2 can be relaxed in certain cases. Details can be found in [1]. Constraint (9b) is necessary as shown below, and (9c) is required to ensure causality.

Proof of the algorithm: Using eqn.(10) in eqn.(6), it can be shown, after some simple algebra, that

$$\begin{vmatrix} y_{i1}(z) \\ u_{i2}(z) \\ u_{i3}(z) \end{vmatrix} = \begin{vmatrix} p_{i}^{-1}(z) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} u_{i1}(z) \\ u_{i1}(z) \\ \theta_{i1}(y_{i2}(z) \\ y_{i3}(z) \end{vmatrix}$$
(11)

Now, if,

$$\begin{vmatrix} y_{i2}(z) \\ y_{i3}(z) \end{vmatrix} = \varphi_{i}(z)u_{i1}(z)$$
(12)

Le., $\varphi_i(z)$ is the generating function for the i^{th} subarray that is the cascade of processing elements that are indexed *i* or greater in Fig.1., then, by substituting eqn.(12) in eqn.(11), using the defining relationship in eqn.(9a) and the interconnecting equations displayed in Fig.1, we have $\varphi_{i+1}(z)$ is the generating function for the $(i+1)^{th}$ sub-array.

The assumption that Θ_i is invertible ensures that the converse of this statement is true, i.e., if $\varphi_{i+1}(z)$ is the generating function for the $(i+1)^{th}$ sub-array and if the i^{th} processing element has a transfer defined by eqns(6,10), then $\varphi_i(z)$ is the generating function for the i^{th} sub-array. (The invertibility of Θ_i is necessary in order to state this [1].) Hence, if the right end of the array in Fig.1 is terminated according to eqn.(7), then the input to output transfer function at the left end of the array is $\varphi(z)$ as desired.

It is easy to check that if $\varphi_i(z)$ is of degree, n_i , then, $\varphi_{i+1}(z)$ is of degree n_i-1 at most. It has been proved in [1] that the degree is exactly n_i-1 . Hence, $\varphi_n(z)$ is a constant vector.

Comments: A remarkable property of the systolic realization algorithm is that almost every known canonical realization of digital filters, e.g., controller, controllability, cascade, parallel [5], all the different versions of the Gray-Markel lattice filter[6], wave-digital translations of passive analog Foster and Cauer canonical forms [7], the orthogonal digital filter realizations of Deprettere et al[8], to name a few, can be obtained by certain choices of the various parameters used in the algorithm. Moreover, for every realization obtained using the systolic realization algorithm, there is a dual realization that can be obtained using the duality theorem in [1], e.g., the observer form realization is dual to the controller form realization, etc.

Properties of The Systolic Realization Algorithm:

Some relevant properties of the systolic realization algorithm are given below. A detailed discussion of these and other properties together with their proofs can be found in [1].

(1) The systolic array obtained by the algorithm is in general a non-computable cascade. However, it can always be made computable by executing bilinear transformations on the functions, $\varphi_i(z)$, at eact, step of the algorithm.

(2) The array leads to a computable cascade as it is if

$$\alpha_1 = 0; \quad \beta_1 = \infty, \quad \text{for all } i \quad (13)$$

This results in $p_i(z) = z$ for all i.

(3) Pipelineability: The array leads to a pipelineable implementation if α_i , β_i are chosen as in eqn.(13) [1.9]. The throughput is maximum, all other factors being the same, if and only if this choice is adhered to [1]. Such arrays will be referred to as completely pipelineable arrays in what follows.

3. COMPLETELY PIPELINEABLE ORTHOCONAL ARRAYS

The systolic array in Fig.1 is said to be orthogonal if every processing element can be implemented as an interconnection of orthogonal rotational modules and x^{-1} blocks. If, in addition, the array is completely pipelineable, it is a completely pipelineable, orthogonal array. It has been shown in [1] that such arrays have very good numerical properties and therefore, satisfy all the desirable criteria mentioned above for VLSI implementation.

A necessary and sufficient condition for a completely pipelineable array to be orthogonal is that the constant matrix, Γ_i , defined by:

$$\Sigma_{1}(z) = diag[z^{-1}, 1, 1]\Gamma_{1}$$
(14)

is an orthogonal matrix for all i. Using the definition of orthogonality and eqn. (10), this reduces to the conditions:

$$\bar{C}_{i}C_{i} = \left[I - \varphi_{i}(\infty)\bar{\varphi}_{i}(\infty)\right]^{-1} \qquad (15a)$$

$$\overline{v}_i^T v_i^T = \varphi_i(\infty) \overline{\varphi}_i(\infty) (1 - \overline{\varphi}_i(\infty) \varphi_i(\infty))^{-1}$$
(15b)

$$\varphi_1(0)\overline{\varphi}_1(\infty) = 1 \tag{15c}$$

Eqn. (15) is solvable if and only if $|\varphi_i(\infty)| < 1$ and eqn.(15c) is true for $\varphi_i(z)$. The systolic realization algorithm then reduces to the Schur algorithm[10].

The question that now requires to be addressed is the following: For what choices of the generating function, can the above equations be solved for all i?. Theorem 1, stated below, a special case of which is originally due to Schur[10], clarifies the situation:

Theorem 1: A completely pipelineable orthogonal array exists for any generating function, $\varphi(z)$, if and only if:

[a] $\varphi(z)$ is composed of scalar rational Schur functions and

[b] $\varphi(z)$ satisfies the equation:

$$1 - \varphi_*(z)\varphi(z) = \frac{\sigma^2}{Q_*(z)Q(z)}$$
(16)

where σ is some real constant, and Q(z) is the polynomial defined in eqn.(3).

In addition, if in eqn.(16), σ is nonzero, then the termination, φ_n , is such that $\overline{\varphi}_n \varphi_n < 1$. Else, $\overline{\varphi}_n \varphi_n \neq 1$.

The proof for the above theorem can be found in [1], wherein a more general theorem is stated that defines the conditions for the existence of orthogonal arrays as such.

Choice of the generating function:

Given a (2x1) function, $\varphi(z)$, that satisfies eqn.(15), and the restrictive choices of the parameters, C_i , α_i , β_i , ψ_i^2 , as defined in eqn.(15), the systolic realization algorithm yields a completely pipelineable orthogonal array. The next step is to obtain this generating function $\varphi(z)$, from the given scalar transfer function, H(z). Again, there are several ways in which this choice can be made.

(1)Firstly, the terminating conditions at the left end of the array have to be chosen.

(2)Secondly, the constant σ in eqn.(16) has to be chosen.

The restrictions for making these choices and the general properties of the resulting arrays can be found in [1]. For the present, suppose the terminating conditions are given as in eqn.(4,5) and that the constant $\sigma = 0$ in eqn.(16). Even then, there are two constant parameters, k_1, k_2 available to the designer, and only two particular choices of these parameters are considered below:

(1) The Direct Embedding:

Suppose $k_2 = 0$. Then, P(z),Q(z),R(z) have to satisfy the relationships:

$$R(z) = N(z) \qquad (17a)$$

$$Q(z) = D(z)/k_1$$
 (17b)

 $P(z)P_{\bullet}(z) = D(z)D_{\bullet}(z)/k_1^2 - N(z)N_{\bullet}(z)$ (17c) From the well-known spectral factorization theorem, it can be derived that eqn.(17c) is solvable if and only if

$$|k_1| \le \frac{1}{|H(e^{jw})|}, \quad \text{for all } 0 \le w \le 2\pi \quad (17d)$$

The resulting pipelined orthogonal array together with some simulation results have been discussed in depth in [11], wherein it has also been shown that the solution to eqn.(17c) is trivially obtainable when H(z) is a Butterworth, or a Tchebycheff, or an Elliptic function, for $k_1 = 1$.

(2) The Darlington Embedding:

Consider the choice, $k_1 = k_2 = 1$. Then,

$$R(z) = N(z) \tag{18a}$$

(19b)

Q(z) + P(z) = D(z)

then, if
$$Q(z)$$
, $P(z)$ are written as:

$$Q(z) = (D(z) + C(z))/2$$
 (15c)

$$(z) = (D(z) - C(z))/2$$
 (19d)

then

$$C(z)D_{*}(z) + C_{*}(z)D(z) = 2N(z)N_{*}(z)$$
 (18e)

Eqn.(18) is precisely the one used by Deprettere et al[8] for obtaining the generating function for their binary signature orthogonal array. The systolic realization algorithm is then analogous to the Darlington synthesis procedure for passive analog filters.

For this termination, however, the designer has to be careful to ensure that the resulting structure is computable, by scaling the transfer function such that $R(\infty)/Q(\infty) = 0$.

A general methodology for obtaining the generating function so that the resulting array is orthogonal is available in [1].

Discussion: In eqn.(15), the matrix, C_i can be chosen to be lower or upper triangular, in which case, the processing element can be implemented using two elementary Givens rotational modules[1].

In [1], a processor configuration termed the Universal Schur filter has been proposed based upon the results presented here, that is capable of implementing timeinvariant systems of arbitrary order, if the processing elements have sufficient memory. The arithmetic unit of the processing element implements the CORDIC algorithm[4] and the logical unit consists of a few counters and gates. The operation of the circuit is assumed to be asynchronous, with the necessary handshaking provided using binary semaphores. A description of the processing element in Concurrent Pascal is given. A VLSI circuit that has at least one such processing element and has access to sufficient memory is capable of implementing any strictly stable linear system in such a manner that the circuit is free of limit-cycle and overflow oscillations and has low coefficient sensitivity. The throughput of the circuit is then directly proportional to the number of processing elements in the circuit, provided this number is an integer submultiple of n. The throughput is not influenced by having more than n processing elements in the circuit. The circuit is also capable of implementing vector transfer functions of arbitrary order.

CONCLUSIONS

In this paper, the systolic realization algorithm for synthesizing digital filters on a VLSI circuit has been presented. The problem of implementing digital filters on a systolic array in a pipelineable fashion has been solved in a numerically robust manner. For a more complete treatment of the VLSI synthesis of digital filters, the interested reader is referred to [1].

REFERENCES

[1] Sailesh K.Rao and Thomas Kailath. 'Digital Filtering in VLSI', Technical Rept., Information Systems Lab, Dept. EE, Stanford Univ., Jan. 1984.

[2] H.T.Kung, 'Why Systolic Architectures?,' IEEE Computer, vol. 25, No.1, pp37-48, Jan. 1982.

[3] L.R.Rabiner and B.Gold, Theory and Applications of Digital Signal Processing, Prentice-Hall Inc., Englewood Chiffs, New Jersey, 1975.

[4] J.E.Volder, 'The CORDIC Trigonometric Computing Technique,' IRE Trans. on Electronic Computers, vol. EC-8, pp330-334, Sep. 1959.

[5] T.Kailath. Linear Systems, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1980.

[6] A.H.Gray, Jr., and J.D.Markel, 'A Normalized Digital Filter Structure,' *IEEE Trans Acoust*, Speech, Signal Processing, vol.ASSP-23, pp.486-494, Oct.1975.

[7] A.Fettweis, 'Digital Filter Structures Related to Classical Filter Networks,' A.E.U. band 25, pp.79-89, 1971.

[8] E.Deprettere and P.Dewilde. Orthogonal Cascade Realization of Real Multiport Digital Filters, Tech. Rep., Network Theory Section, Dept. of Elec. Engg., Delft Univ. of Tech., Delft, The Netherlands, 1980.

[9] H.V.Jagadish, T.Kailath, J.A.Newkirk, and R.G.Mathews, 'Pipelining in Systolic Arrays,' Conf. Rec. Seventeenth Asilomar Conference on Circuits and Systems, Pacific Grove, 1983.

[10] I.Schur, 'Ueber Potenzreihen, die im Innern des Einheitskreises beschrankt sind,' J. fur die Reine und Angewandte Mathematik, vol. 147, pp.205-232, Berlin, 1917.

[11] Sailesh K.Rao and Thomas Kailath, 'Orthogonal Digital Filters for VLSI Implementation', to appear in the IEEE Trans. on Circuits Syst.



UMCLASSIFIED SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE					
14. REPORT SECURITY CLASSIFICATION	16. RESTRICTIVE MARKINGS				
UNCLASSIFIED					
28. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT			
2 DECLASSIFICATION/DOWNGRADING SCHEDULE		Approved for public release; distribution unlimited.			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 84-0518			
Sa NAME OF PERFORMING ORGANIZATION Stanford University	6b. OFFICE SYMBOL (If applicable)	7. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research			
5c. ADDHESS (City. State and ZIP Code) Department of Electrical Engineering Stanford CA 94305		7b. ADDRESS (City, State and ZIP Code) Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332			
B. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
AFOSE N1		AF0SR-83-0228			
8c. ADDRESS (City, State and ZIP Code)		10. SOURCE OF FUN	IDING NOS.		
		PROGRAM	PROJECT	TASK	WORK UNIT
Bolling AFB DC 20332		ELEMENT NO.	NU.	NO.	NU.
TITLE Include Security Classification		61102F	2304	A6	
PIPELINED ORTHOGONAL DIGITAL LA	Į				
2. PERSONAL AUTHOR(S) S.K. Rao and T. Kailath				ha	
134 TYPE OF REPORT 136. TIME COVERED		14. DATE OF REPOR	IT (Yr., Mo., Day)	15. PAGE CO	DUNT
Technical FROM TO		MAR_1984		4	
S. SUPPLEMENTARY NOTATION ICASSP, MAR 1984, San Diego, California.					
7. COSATI CODES 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)					
FIELD GROUP SUB. GR.					i
3. ASSTRACT (Continue on reverse if necessary and identify by block number) An algorithm is presented for the design of systolic arrays that implement single-input single-output time-invariant digital filters. The algorithm is then specialized to the case where the realized array consists only of orthogonal rotational modules and delay elements interconnected in such a manner as to render the circuit pipelineable.					
DISTRIBUTION/AVAILABILITY OF ABSTRA		21 ABSTRACT SECU	RITY CLASSIFI	CATION	
CLASSIFIED/UNLIMITED & SAME AS HPT					
2: NAME OF RESPONSIBLE INDIVIDUAL		1225 TELEPHONE NU	JMBER de i	22c. OFFICE SYMI	BOL
CPT Brian W. Woodruff		(101) 767-	50 '7	101	
DE FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE					

-

