TIME-RECURSIVE ARCHITECTURES AND WAVELET TRANSFORM

Emmanuel Frantzeskakis   John S. Baras   and   K. J. Ray Liu

Electrical Engineering Department, Institute of Systems Research
University of Maryland, College Park, MD 20742

Abstract

The time-recursive computation has been proved as a particularly useful tool in real-time data compression and in transform domain adaptive filtering, with applications in the areas of audio, radar, sonar and video. Unlike the FFT based ones, the time-recursive architectures require only local communication. Also, they are modular and regular, thus they are very appropriate for VLSI implementation and they allow high degree of parallelism. In this paper, we propose an architectural framework for parallel time-recursive computation. We consider a class of linear operators that consists of the discrete time, time invariant, compactly supported, but otherwise arbitrary kernel functions. We define a shift property of the linear operators and reveal its relation with the time-recursive implementation. We demonstrate the potential of the proposed framework by designing a time-recursive architecture for the Discrete Wavelet Transform.

1   INTRODUCTION

In many signal processing applications the key computation consists of a *mapping operator* $[h_0 \; h_1 \; \cdots \; h_{N-1}]$: $x(\cdot) \to X(\cdot)$, which operates on the semi-infinite sequence of scalar data $x(\cdot)$ and produces the sequence $X(\cdot)$ as follows:

$$X(t) = \sum_{n=0}^{N-1} h_n x(t + n - N + 1), \quad t = 0, 1, \cdots. \quad (1)$$

A *time-recursive implementation* of a mapping operator $[h_n \; h_1 \; \cdots \; h_{N-1}]$ is the one that is based on an update computation of the type

$$X(t+1) = \mathcal{U}(X(t), x(t+1)).$$

For example, if we have $[h_n = 1, n = 0, 1, \cdots, N - 1]$, then $X(t)$ will be the sum of the last N values in the input stream. The recursive algorithmic implementation of this operator will be simply the computation

$$X(t+1) = X(t) + x(t+1) - x(t - N + 1).$$

The time-recursive computation has been proved as a particularly useful tool in real-time data compression [1, 2, 3] and in transform domain adaptive filtering [4, 5, 6], with

applications in the areas of audio, radar, sonar and video. There is a common infrastructure among the mapping operators that are involved in these diverse applications. The unifying feature is a *shift property* we discuss in the following Section. We also show how this property dictates the time-recursive architectural design. In Section 3, we design a time-recursive architecture for the Discrete Wavelet Transform (DWT). We conclude with Section 4.

2   ARCHITECTURAL FRAMEWORK

We can specify a mapping operator $[h_0 \; h_1 \; \cdots \; h_{N-1}]$ with a function $f(\cdot)$, for which the values at the points $0, 1, \cdots, N - 1$ are the prescribed coefficients: $h_n = f(n)$, $n = 0, 1, \cdots, N - 1$. In the sequel, we will use the term *kernel function* or simply *kernel* for this function $f(\cdot)$. Furthermore, we will call *kernel group* a vector of kernel functions

$$\mathbf{f}(\cdot) = [f_0(\cdot) \; f_1(\cdot) \; \cdots \; f_{M-1}(\cdot)]^T.$$

Shift Property: *A kernel group* $\mathbf{f}(\cdot)$ *satisfies the shift property (SP), if it satisfies the (matrix) difference equation*

$$\mathbf{f}(n - 1) = \mathbf{R}\mathbf{f}(n), \quad n = 1, 2, \cdots, N, \quad (2)$$

with specified final condition $\mathbf{f}(N)$, *where* $\mathbf{R}$ *is a constant matrix of size* $M \times M$.

Lemma 1 *A recursive implementation of a kernel group* $\mathbf{f}(\cdot)$ *is feasible if this kernel group satisfies the shift property.*

Proof: (2) gives:

$$f_p(n - 1) = \sum_{q=0}^{M-1} r_{pq} f_q(n),$$

for $n = 1, 2, \cdots, N$, $p = 0, 1, \cdots, M - 1$, where $r_{pq}, p, q = 0, 1, \cdots, M - 1$ are the elements of the matrix $\mathbf{R}$. Let

$$X_p(t) = \sum_{n=0}^{N-1} f_p(n) x(t + n - N + 1), \quad (3)$$

for $p = 0, 1, \cdots, M - 1$. Suppose this is available at the time instant $t+1$. For the quantities $X_p(t+1), p = 0, 1, \cdots, M-1$

we have:

$$X_p(t+1) = \sum_{n=0}^{N-1} x(t+n+1-N+1)f_p(n) =$$
$$\sum_{n=1}^{N} x(t+n-N+1)f_p(n-1) =$$
$$\sum_{n=1}^{N} x(t+n-N+1)\sum_{q=0}^{M-1} r_{pq}f_q(n) =$$
$$\sum_{q=0}^{M-1} r_{pq}\left(\sum_{n=1}^{N} x(t+n-N+1)f_q(n)\right)$$

and therefore we obtain:

$$X_p(t+1) = \sum_{q=0}^{M-1} \{r_{pq} \\ [X_q(t) - x(t-N+1)f_q(0) + x(t+1)f_q(N)]\}, \quad (4)$$

$p = 0, 1, \cdots, M-1$. If we assume knowledge of the boundary values $\{f_q(0), f_q(N), q = 0, 1, \cdots, M-1\}$, (4) specifies the algorithm that performs the update computation we were after. Furthermore, note that if R is nonsingular, knowledge of $f(0)$ yields $f(N)$. $\quad \square$

Corollary 1 *A kernel group $f(\cdot)$ that satisfies the shift property can be implemented recursively as follows:*

1. *Compute the matrix R by evaluating $f(n-1)$ and using (2).*

2. *Evaluate $f(n)$ at the points $n = 0$ and $n = N$.*

3. *At each time instant t evaluate (4).*

Note that the first two steps of the above procedure belong in the initialization phase (off-line computation).

The issue of specifying a family of kernel groups that satisfy the shift property is addressed by lemma 2:

Lemma 2 *The shift property (SP) is satisfied by:*

1. *The singleton kernel group $[cb^n]$, where b and c are non-zero free parameters.*

2. *The kernel group $\left[c_{00}b^n + c_{01}b^{-n}, c_{10}b^n + c_{11}b^{-n}\right]^T$, where b is a non-zero parameter and the coefficients are free parameters, such that $c_{00}c_{11} - c_{01}c_{10} \neq 0$.*

3. *The kernel group $\left[c_0, c_1 n, \cdots, c_Q n^Q\right]^T$, where Q is an arbitrary positive integer and the coefficients are non-zero parameters.*

4. *The union of two kernel groups that satisfy SP.*

5. *The cartesian product of two kernel groups that satisfy SP.*

The proof of this lemma can be found in [7]. The kernel functions of the Short Term Fourier Transform, the Discrete Cosine Transform and the Modulated Lapped Transform [8] belong in the class of kernels specified by lemma 2. Efficient time-recursive architectures for these transforms have been designed in [9], in a systematic way. On table 1, we provide the relevant cost metrics. In the following Section, we address the problem of the time-recursive architecture design for the Discrete Wavelet Transform (DWT).

3 TIME-RECURSIVE ARCHITECTURE FOR THE DWT

We consider the implementation of a pair of finite impulse response filters (FIR),

$$H = [h_{N-1} \ h_{N-2} \ \cdots \ h_0]$$

and

$$G = [g_{N-1} \ g_{N-2} \ \cdots \ g_0],$$

that are used in the implementation of the DWT, in a time-recursive way. Implementing the filters H and G is equivalent to implementing the pair of mapping operators

$$\begin{bmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ g_0 & g_1 & \cdots & g_{N-1} \end{bmatrix}. \quad (5)$$

In order to proceed with the architecture design we need to determine the kernel group associated to the pair of mapping operators (5).

Lemma 3 *The size of the smallest kernel group that can be used to implement the pair of mapping operators (5) in a time-recursive way is equal to the minimal order of the partial realization of the 1-input 2-output LTI system with the N first Markov parameters being equal to the coefficients of the specified operator [1].*

Proof: Given the pair of mapping operators (5) we can have the following coefficient expansion:

$$[h_n \ g_n]^T = cA^n b, \quad n = 0, 1, \cdots, N-1, \quad (6)$$

where the sizes of A, b and c are $M \times M, M \times 1$ and $2 \times M$ respectively [10, 11]. Let

$$f(n) = A^n b \quad (7)$$

be a kernel group of size M. Since $f(n-1) = A^{n-1}b = A^{-1}f(n)$, this kernel group satisfies the shift property with

$$R = A^{-1} \quad \text{and} \quad f(0) = b. \quad (8)$$

From (6) and (7) we get a *linear decomposition* of the mapping operator coefficients $[h_n \ g_n]^T = cf(n)$. Therefore, the time recursive implementation of the mapping operator can be based on the the kernel group $f(\cdot)$. In our construction, the size of the kernel group M is equal to the order of the realization $\{A, b, c\}$. Thus, minimizing the number of the decomposition kernels is equivalent to minimizing the order of the the partial realization of the LTI system, for which the first N Markov parameters are equal to the coefficients of our operator. $\quad \square$

We can proceed now with the architecture design as follows. We specify the partial realization $\{A, b, c\}$ of minimal order M for the linear system, so that

$$[h_n \ g_n]^T = cA^n b, \quad n = 0, 1, \cdots, N-1 \quad (9)$$

---

[1]For a tutorial text on the linear time invariant (LTI) systems, the partial order realization and the Markov parameters of an LTI system one may refer to [10].

[10, 11]. We bring the triplet $\{A, b, c\}$ in the modal canonical form [10]. Since H and G specify a lossless system the magnitude of all the eigenvalues of the system matrix A is equal to 1 [12]. For the sake of concreteness, suppose that the order of the system is $M = 3$. The format of the matrix A will be as follows:

$$A = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & \beta \end{bmatrix},$$

where α takes values in the interval $[0, 2\pi)$ and β equals either 1 or -1. The $M \times 1$ vector b and the $2 \times M$ matrix c do not have any particular structure.

By substituting the above expression of A in (9) and expanding the matrix notation we obtain

$$[h_n \quad g_n]^T$$

$$\begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix} \begin{bmatrix} \cos\alpha n & \sin\alpha n & 0 \\ -\sin\alpha n & \cos\alpha n & 0 \\ 0 & 0 & \beta^n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

and consequently

$$\begin{bmatrix} h_n \\ g_n \end{bmatrix} = \begin{bmatrix} c_{00}b_0 + c_{01}b_1 \\ c_{10}b_0 + c_{11}b_1 \end{bmatrix} \cos\alpha n + \begin{bmatrix} -c_{01}b_0 + c_{00}b_1 \\ c_{11}b_0 + c_{10}b_1 \end{bmatrix} \sin\alpha n + \begin{bmatrix} c_{02}b_2 \\ c_{12}b_2 \end{bmatrix} \beta^n. \quad (10)$$

The kernel groups we need to implement are

$$f_0(n) = \begin{bmatrix} f_{00}(n) \\ f_{01}(n) \end{bmatrix} = \begin{bmatrix} \cos\alpha n \\ \sin\alpha n \end{bmatrix}$$

and

$$f_1(n) = f_{10}(n) = \beta^n.$$

For the kernel group $f_0(\cdot)$ we have $f_0(n-1) = R f_0(n)$ with

$$R = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}.$$

We also have

$$\begin{bmatrix} f_{00}(0) \\ f_{01}(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} f_{00}(N) \\ f_{01}(N) \end{bmatrix} = \begin{bmatrix} \cos\alpha N \\ \sin\alpha N \end{bmatrix}.$$

The resulted architecture implies module M_0 in figure 1. On the other hand, for the singleton kernel group $f_1(\cdot)$ we have $f_1(n-1) = R f_1(n)$ with $R = \frac{1}{\beta}$ and also $f_{10}(0) = 1$, $f_{10}(N) = \beta^N$. The associated architecture is demonstrated by module M_1 in figure 1. No multipliers are needed for the implementation of $f_1(\cdot)$, since all the parameters involved have unit magnitude. The architectural implementation of the given pair of mapping operators for the case where $M = 3$ is shown in figure 1.

For the general case of a system of an arbitrary order M we need to implement M kernel functions. Among these functions no more than two are of the form of $f_1(\cdot)$ seen above (since only two distinct such functions exist with $\beta = 1$ and $\beta = -1$) and they are implemented by module M_1. The rest of the kernel functions will group into

pairs of cosine-sine functions specified by the parameter α, as dictated by $f_0(\cdot)$ in the above example, and they can be implemented by module M_0.

The implementation of module M_0 in figure 1 requires 2 multipliers, 3 adders and one CORDIC processor that will evaluate the rotation involved. For the implementation of module M_1 we need 2 adders. We implement the desired pair of mapping operators as two weighted sums of the outputs of the above described parts. If the size of the associated kernel group is M the cost of this interconnection is $2M$ multipliers and $2(M-1)$ adders. The overall cost of the design is not higher from $3M$ multipliers. $\lfloor 7M/2 \rfloor$ adders and $M/2$ CORDIC processors.

Let us consider now the implementation of the pair of wavelet filters H and G, whose coefficients are given on table 2 [13]. The lengths of the filters H and G are 9 and 7 respectively. The size of the kernel group we have to implement (that is the order of the associated linear system) is $M = 6$. The architecture involves 2 copies of module M_0 and 2 copies of module M_1. The values of the parameters α and β, as well as the output weights are given on table 2. The resulted architecture is shown on figure 2.

For the inverse Discrete Wavelet Transform (IDWT) we have to implement the mirror filters $\tilde{H}$ and $\tilde{G}$ of G and H respectively [13]. The architectural implementation is obtained if we simply replace the parameter α with $\pi - \alpha$ in the M_0 modules and the parameter β with $-\beta$ in the M_1 modules in figure 2. The size of the kernel group is $M = 6$ and the implementation cost is 15 multipliers, 20 adders and 2 CORDIC processors.

4 CONCLUSION

The *shift property* dictates the common infrastructure of the time-recursive computation in a variety of applications. The time-recursive approach yields architectural implementations that are modular, regular and require local communication, thus they are very appropriate for VLSI implementation. The time-recursive implementation of the DWT is discussed in detail. The same design procedure can be applied for implementing an arbitrary lossless system [12], thus it establishes the generic nature of the time-recursive architectural framework. The implementation cost depends on the eigenvalue decomposition of this system [9]. In terms of cost efficiency, representative examples among the time-recursive architectures are the STFT, the DCT and the MLT [9].

References

[1] K.J.R. Liu and C.T. Chiu. Unified Parallel Lattice Structures for Time-Recursive Discrete Cosine/Sine/Hartley Transforms. To appear, IEEE Transactions on Signal Processing, May 1993.

[2] C.T. Chiu and K.J.R. Liu. Real-Time Parallel and Fully Pipelined Two-Dimentional DCT Lattice Structures with Application to HDTV Sysytems. IEEE Transactions on Circuits and Systems for Video Technology, 2(1):25-37, March 1992.

[3] K.J.R. Liu, C.T. Chiu, R.K. Kolagolta, and J.F. Jaja. Optimal Unified Architectures for the Real-Time Computation of Time-Recursive Discrete Sinusoidal Transforms. *Submitted to IEEE Transactions on Circuits and Systems for Video Technology*, 1992.

[4] R.R. Bitmead and B.D.O. Anderson. Adaptive Frequency Sampling Filters. *IEEE Transactions on Circuits and Systems*, 28(6):524–534, June 1981.

[5] G.A. Clark, M.A. Soderstrand, and T.G. Johnson. Transform Domain Adaptive Filtering Using a Recursive DFT. In *Proc. IEEE ISCAS*, pages 1113–1116, June 1985.

[6] N.R. Murthy and M.N.S. Swamy. On the Computation of Running Discrete Cosine and Sine Transforms. *IEEE Transactions on Signal Processing*, 40(6):1430–1437, June 1992.

[7] E. Frantzeskakis, S.J. Baras, and K.J.R. Liu. Time-Recursive Computation and Real-Time Parallel Architectures, Part I: Framework. *Submitted to IEEE Trans. on Signal Processing*, Jan. 1993.

[8] H.S. Malvar. Lapped Transforms for Efficient Transform/Subband Coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(6):969–978, June 1990.

[9] E. Frantzeskakis, S.J. Baras, and K.J.R. Liu. Time-Recursive Computation and Real-Time Parallel Architectures, Part II: Methodology and Applications. *Submitted to IEEE Trans. on Signal Processing*, Jan. 1993.

[10] T. Kailath. *Linear Systems*. Prentice Hall, London, 1980.

[11] S.Y. Kung. *Multivariable and Multidimentional Systems: Analysis and Design*. PhD thesis, Stanford University, June 1977.

[12] P.P. Vaidyanathan and Z. Doganata. The Role of Lossless Systems in Modern Digital Signal Processing: A Tutorial. *IEEE Transactions on Education*, 32(3):181–197, Aug. 1989.

[13] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image Coding Using Wavelet Transform. *To appear in IEEE Transactions on Signal Processing*

[14] K.J.R. Liu. Novel Parallel Architectures for Short Time Fourier Transform. *To appear in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 1993.

	multiplications	additions	CORDICs
DCT	$2N - 1$	$3N + 2$	-
STFT	-	$3N$	$N - 1$
MLT	$2N + 3$	$3N + 3$	$N - 1$

Table 1: Operator counts for time-recursive architectures of some N-point block transforms.

h_n	g_n	α or β	$weight_h$	$weight_g$
0.0267	0.0000	1.0	0.1781	-0.0032
-0.0168	0.0456			
-0.0782	-0.0287	-1.0	-0.0056	0.1778
0.2668	-0.2956			
0.6029	0.5575	1.1085	-0.0881	-0.0235
0.2668	-0.2936		-0.3089	-0.0781
-0.0782	-0.0287			
-0.0168	0.0456	2.0929	-0.0575	-0.1511
0.0267	0.0000		0.0998	0.2673

Table 2: Example of wavelet filter coefficients and the associated architecture parameters.
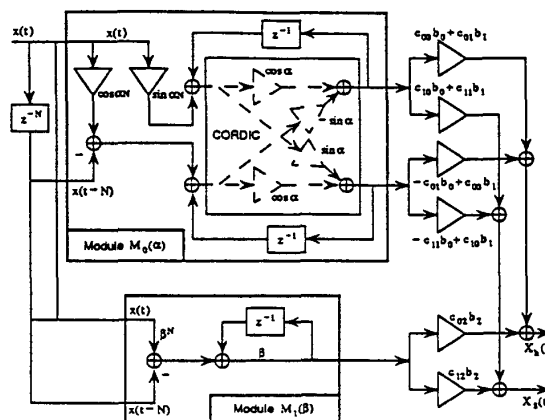

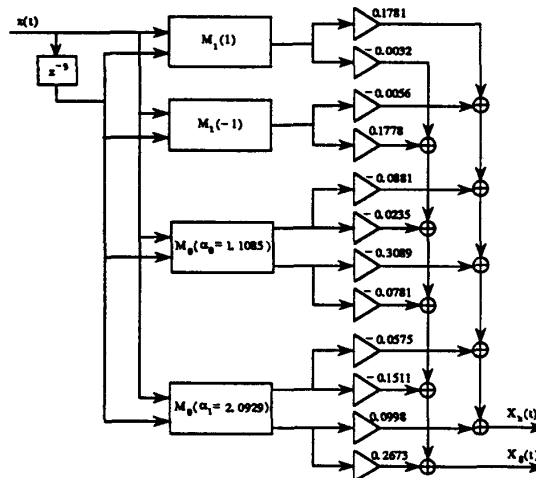Figure 1: The architectural modules used for DWT.

Figure 2: Time-recursive architecture for the DWT filters specified in table 2.