BLOCK PROCESSING STRUCTURES FOR FIXED POINT DIGITAL FILTERING

Claude Alain Wambergue*
T.R.T.
5, Avenue Reaumur
F.92350. Le Plessis Robinson
FRANCE

and

Richard A. Roberts
Department of Electrical Engineering
University of Colorado
Boulder, Colorado 80309

*Now with: Thomson C.S.F.
1, rue des Mathurius, F.92222. Bagneux, FRANCE

ABSTRACT

This work compares block processing structures using as a figure of merit the product of the chip area needed to implement the structure in NMOS and the reciprocal of the word rate.

It considers both distributed arithmetic schemes and conventional multiplier implementation of block filters and compares these structures with conventional filter structures. By fixing the output signal quality of the filters one is able to find the best block processing configurations in terms of the assumed figure of merit.

INTRODUCTION

Block state-space structures for multi-input, multi-output filtering possess two primary advantages. The inherent parallelism of the block structure increases data throughput and also reduces the noise caused by the round off of internal multiplies. These advantages accrue from the extra degree of freedom one obtains in processing a vector input instead of a scalar.

Block structures, however, require more hardware resources which increases as the block length of the vector input increases. There are several forms that one can use in block processing which involve parallel and cascade substructures. How should we best employ these degrees of freedom in block processing to maximize data throughput and minimize hardware complexity?

Criteria such as the number of multiplies, the data throughput rate, or the noise properties are commonly used to evaluate filter structures. We shall use a figure of merit [1] equal to the product of chip area needed to fabricate the filter and the reciprocal of word rate to compare filter structures. Furthermore we shall fix the output signal quality. Under these assumptions we find the best block processing configuration.

1. Block State Equations and Roundoff Noise

Consider a time-invariant digital filter of order n, with single input sequence $\{u(k)\}$, single output sequence $\{y(k)\}$, and its state variable description:

$$(1)\qquad \begin{aligned} x(k+1) &= A\,x(k) + B\,u(k) \\ y(k) &= C\,x(k) + D\,u(k) \end{aligned}$$

If we divide the input sequence and the output sequence into blocks of length L, defining

$$u(k) = [u(k),\, u(k+1),\, \dots ,\, u(k+L)]^T$$

$$y(k) = [y(k),\, y(k+1),\, \dots ,\, y(k+L)]^T$$

and computing states at time (k+1) with (1), as a function of states at time k and u(k), and y(k) expressed on the same terms, we obtain the block state equations:

$$(2)\qquad \begin{aligned} x(k+1) &= A'\,x(k) + B'\,u(k) \\ y(k) &= C'\,x(k) + D'\,u(k); \quad k = 0, L, 2L, \dots \end{aligned}$$

where $A' = A^L$ is n x n

$B' = [A^{L-1}B,\, A^{L-2}B,\, \dots ,\, AB,\, B]$ is n x L

$C' = [C,\, CA,\, \dots ,\, CA^{L-1}]^T$ is L x n

and

$$D' = \begin{bmatrix} D & O & D & \cdots & O \\ CB & D & O & & \cdot \\ CAB & CB & D & & \cdot \\ \cdot & \cdot & \cdot & \ddots & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ CA^{L-2}B & CA^{L-3}B & CA^{L-4}B & \cdots & D \end{bmatrix}$$

is L x L, lower triangular and Toeplitz.

Comparing (1) and (2) several observations can be made [2]. Among them, that the Block-structure computes L output samples at a time, therefore increasing the data throughput rate capability.

CH 1746-7/82/0000 - 0498 $ 00.75 © 1982 IEEE

Also, that when we derive a block structure from a good (i.e., ℓ_2-scaled, minimum noise) simple state-space realization, the properties of the latter are retained (and often improved). In particular there is a reduction in the noise caused by the roundoff of internal multiplies. In fact, one can show that

$$(3)\quad (\sigma^2_{y_j})_{ave} = \{1 + \frac{1}{L}\sum_{k=0}^{\infty} CA^k(CA^k)^T\}\,\sigma_0^2$$

This equation states that in block processing the contribution to the average roundoff noise from internal roundoff noise sources, σ_0^2, is reduced by a factor of L, the block length. There is a distribution of the internally generated roundoff noise over the L output samples.

2. Block Structure Implementations

There are other advantages to block processing. Barnes and Shinuaka [2] point out the use block realizations for the elimination of all autonomous limit cycles. In parallel substructures, different parallel subfilters could be realized with different block lengths so that subfilters associated with poles close to the unit circle could use longer block lengths to control roundoff noise and limit cycles. Block structures can reduce the number of multiplies per output sample [3]. An nth order state space structure requires $(n+1)^2$ multiplies and $n(n+1)$ additions per output samples. The block structure results in a savings of

$(L-1)(2n^2-L)/2L$ multiplies if $1<L<2n^2$

and

$(L-1)(2n^2-2n-L)/2L$ additionsif $1<L<2n(n-1)$.

Thus the block structure is computationally more efficient than single-input-output state space structures for $1<L<2n(n-1)$.

The optimal value of the block length L depends on the order of the subfilters, call it m, that make up a realization [4]. The optimal values for L are: $L = n\sqrt{2}$ for a full state space structure and $L = m\sqrt{2}$ for cascaded or parallel substructures. In the case of a filter realized with second-order subfilters, the optimal block length is L=3 and represents about 20% fewer multiplies per output sample vis-a-vis a simple cascade of second-order state space subfilters.

In addition to the above advantages the block structure includes an inherent parallelism in its operation that can be exploited by the use of distributed arithmetic [4,5]. Distributed arithmetic is a hardware implementation technique which replaces conventional multipliers with precalculated stored partial products.

The purpose of this paper is to determine on the basis of hardware complexity and data throughput which block structures are best and how they compare to other well-known structures.

3. A Comparison of Implementations

We shall evaluate all the implementations in the sequel for the same output signal quality, i.e., for the same output roundoff noise. Each implementation will have a word length B that is adjusted so that the output roundoff noise is identical for all the implementations. Under this constraint we shall define a figure of merit [1] F_2 as

$$(4)\quad F_2 = \frac{\text{nMOS area needed to fabricate filter}}{\text{word rate of filter}}$$

This figure of merit depends on a particular technology, however, we argue that relative sizes of the areas for various logic functions are approximately the same for all technologies.

We shall consider one example in some detail - a sixth-order Butterworth lowpass filter with a cutoff frequency of 2% $f_{s/2}$, implemented in second-order subfilters. The subsections have the following transfer functions

$$H_1(z) = \frac{(z+1)^2}{z^2-1.964z+.96802},$$

$$H_2(z) = \frac{(z+1)^2}{z^2-1.9112z+.91498},$$

$$H_3(z) = \frac{(z+1)^2}{z^2-1.8819z+.88563}.$$

Calculating the noise gains of each section [6] we can determine the word lengths necessary to give the same output roundoff noise for the various realizations. These results are summarized in Table 1.

From these word length values, we can now proceed to calculate the area needed to fabricate each structure in an nMOS chip. We use Table 2 to obtain the required areas adding in some percentage for the interconnection areas and the bonding pads. We use 50 nanoseconds as a memory access time and also as a multiply time. (Our word lengths are nominally 12 bits.)

Table 3 summarizes our results. Shown in Table 3 are results for both multiplier and distributed arithmetic implementations. As the block length L increases, a distributed arithmetic structure requires large amounts of memory. Partitioning the inner product length into sums of smaller lengths helps in reducing these memory requirements.

4. Discussion of Results

There are two implementation techniques evaluated in Table 3. "Multiplier structures" are fabricated using isolated monolithic multipliers. "Distributed arithmetic structures" use table look-up of precomputed partial products. Using F_2 as a figure of merit does not necessarily imply that the block length for "computational efficiency" (L=3) will be the best block length for minimizing F_2. For the multiplier structures, in fact, L=5 is a better block length.

For this example the cascaded structures are better than paralleled structures. This is clearly related to the lower noise level for the cascaded sections which results in a smaller word length B. For higher values of L, the large memory requirement is responsible for the rapidly increasing value of F_2. The best structure from Table 3 is a distributed arithmetic implementation of cascaded section-order sections with a block length of L=3.

TABLE 1

	Word Length B for Normal 2nd Order Sections in Parallel		Normal 2nd Order Sections in Cascades, Optimal Ordering	
Block Length L	Multiplier Structure	Distributed Arithmetic Structure	Multiplier Structure	Distributed Arithmetic Structure
1	10	10	8	9
2	9	10	8	8
3	9	9	8	8
4	9	9	8	8
5	9	9	7	8
6	9	9	7	8
7	8	9	7	7
8	8	8	7	7

TABLE 2

Function Description	nMOS Area Needed
Bonding pad	$(0.090\ \mathrm{mm})^2$
8 bit parallel register	$(0.065\ \mathrm{mm})^2$
8 bit adder	$(0.29\ \mathrm{mm})^2$
32x16 bit ROM	$(0.36\ \mathrm{mm})^2$
16 bit x 16 bit multiplier	$(1.96\ \mathrm{mm})^2$
1 bit full adder	$(0.031\ \mathrm{mm})^2$

TABLE 3

a. Second Order Sections in Parallel

	Multiplier Structure			Distributed Arithmetic Structure No Partioning of the Inner Products			Distributed Arithmetic Structure Partioning into 2 Sums		
L	Area (mm^2)	Output Word Rate (MHz)	F_2	Area (mm^2)	Output Word Rate (MHz)	F_2	Area (mm^2)	Output Word Rate (MHz)	F_2
1	33.245	20	1.662	0.746	1.82	0.411			
2	49.820	40	1.245	1.059	3.64	0.291			
3	72.913	60	1.216	1.353	6.	0.225			
4	99.444	80	1.243	1.958	8.	0.245	2.868	8.	0.358
5	129.243	100	1.292	2.991	10.	0.299	3.219	10.	0.322
6	162.378	120	1.353	4.87	12.	0.406	3.831	12.	0.319
7	176.825	140	1.263	8.427	14.	0.602	4.421	14.	0.316
8	212.223	160	1.326	13.656	17.78	0.768	4.638	17.78	0.261

b. Normal Second-order Sections in Cascade (Sections Optimally Ordered)

	Multiplier Structure			Distributed Arithmetic Structure No Partitioning of the Inner Products			Distributed Aritmetic Structure Partioning into 2 Sums		
L	Area (mm^2)	Output Word Rate (MHz)	F_2	Area (mm^2)	Output Word Rate (MHz)	F_2	Area (mm^2)	Output Word Rate (MHz)	F_2
1	26.689	20	1.334	0.681	2.0	0.340			
2	44.343	40	1.109	0.867	4.44	0.195			
3	64.929	60	1.082	1.216	6.67	0.182			
4	88.462	80	1.106	1.756	8.89	0.198	2.568	8.89	0.289
5	100.65	100	1.006	9.678	11.11	0.241	2.881	11.11	0.259
6	126.417	120	1.053	4.353	13.33	0.327	3.427	13.33	0.257
7	154.781	160	1.106	6.617	17.5	0.378	3.484	17.5	0.199

REFERENCES

[1] M. Arjmand and R. A. Roberts, "On the Speed and Complexity of Distributed Arithmetic Structures for Digital Signal Processing," 13th Asilomar Conference, November 1979.

[2] C. W. Barnes and S. Shimaka, "Finite Word Effects in Block-state Realizations of Fixed-point Digital Filters," IEEE Trans. Circuits and Syst., Vol. CAS-27, No. 5, May 1980.

[3] P. Ananthalrishna and S. K. Mika, "Block-state Recursive Digital Filters with Minimum Roudoff Noise," Proceedings of Int. Conf. in Acoustics, Speech and Signal Processing, April 1980, pp. 81-84.

[4] J. Zeman and A. G. Lindgren, "Fast Digital Filters with Low Roundoff Noise," IEEE Trans. Circuits and Syst., Vol. CAS-28, No. 7, July 1981.

[5] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters," IEEE Trans. Acoustics, Speech and Signal Proc., Vol. ASSP-22, No. 6, December 1974.

[6] W. L. Mills, C. T. Mullis, R. A. Roberts, "Low Roundoff Noise and Normal Realizations of Fixed Point IIR Digital Filters," IEEE Trans. on Acoustics, Speech and Signal Proc., Vol.ASSP-23, No.4, Aug.1981.