ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

ARCHITECTURE DRIVEN FILTER TRANSFORMATIONS

Manish Sharma and Naresh R. Shanbhag

Coordinated Science Laboratory/ECE Dept.
University of Illinois at Urbana Champaign, Urbana, IL-61801.

ABSTRACT

In this paper, we present the *Sum of Powers-Of-Two* (SPOT) algorithm transformation that results in a high-speed IIR filter architecture by forcing the first few coefficients of the denominator polynomial to powers of two or sums of powers of two. The SPOT transform achieves the same result as achieved by conventional pipelining techniques such as scattered look-ahead and minimum order augmentation but with significantly smaller pipelining overhead and similar sensitivity to coefficient quantization. For typical examples, the SPOT transform roughly saves 30% hardware complexity over existing techniques. Architectures for implementation of the transformed filter transfer functions have also been described.

1. INTRODUCTION

Pipelining is widely used to obtain high speed digital filter implementations. However, the inner feedback loops in an IIR filter present a pipelining bottleneck. Traditionally this problem has been overcome via techniques such as *look-ahead* pipelining [1] [3] and *minimum order augmentation* [2]. These techniques transform a given serial IIR filter into a pipelined filter by adding additional poles (and cancelling zeros to preserve the input-output relationship) so that the first few coefficients in the denominator of the pipelined transfer function (TF) are zero. Note that in this process a pipelining hardware overhead has to be incurred. In the *clustered lookahead* technique [1], the pipelining overhead is M, where M is the pipelining level and N is the filter order. However, stability of the pipelined filter is not guaranteed. The *scattered lookahead* transformation [1] guarantees stability at the expense of a higher pipelining overhead of MN ($Nlog_2M$ if decomposition is employed). The pipelining overhead in the look-ahead technique appears as an increase in the order of the numerator polynomial. The number of non-zero coefficients in the denominator polynomial remains the same. Minimum order augmentation also guarantees stability but at the expense of additional non-zero terms in the denominator.

In this paper, we exploit the fact that speedup can also be achieved by reducing the computational complexity of the inner feedback loops so that pipelining of these loops is no longer required. This can be done by transforming the filter TF such that the first few coefficients in the denominator of the resulting function are either powers of two or sums of a few powers of two so that multipliers are replaced by shifters and skeletal multipliers (multipliers composed of shifters and a few adders). This method imposes lesser constraints on the transformation and therefore may result in better solutions in terms of the hardware overhead required for the same speedup as compared to the other techniques.

In this paper, we present the *Sum of Powers Of Two* (SPOT) algorithm transformation that systematically transforms a given serial transfer function (without altering the input-output mapping) into a transfer function that achieves similar speed-ups as pipelined transfer functions but with much lower overhead. Note, that the SPOT transform is different from powers-of-two filter design techniques [4] as the latter start with a filter design specifications and generate a transfer function with powers-of-two coefficients.

Section 2 presents the SPOT transform in detail. Section 3 describes possible architectures for the implementation of SPOT transformed filter functions. Some examples are given in Section 4.

2. THE SPOT TRANSFORM

Consider an IIR filter transfer function as given in (1):

$$H(z) = \frac{B(z)}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \tag{1}$$

The goal of an M-Level SPOT transform is to modify the original transfer function to the one given in (2):

$$H_T(z) = \frac{B_T(z)}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} \cdots + \hat{a}_M z^{-M} \cdots + \hat{a}_X z^{-K}} \tag{2}$$

where $\hat{a}_1, \hat{a}_2 \cdots \hat{a}_M$, the first M coefficients in the denominator polynomial, are either powers of two or sums of powers of two.

The desired transfer function (2) can be obtained by multiplying the denominator and numerator polynomials of the original transfer function (1) by a suitable polynomial $D(z)$ of degree $L = K - N$ as shown below,

$$H_T(z) = \frac{B(z)D(z)}{A(z)D(z)} = \frac{B_T(z)}{A_T(z)}$$

$$D(z) = 1 + \sum_{i=1}^{L} d_i z^{-i}, \qquad L = K - N$$

$$A_T(z) = 1 + \sum_{i=1}^{K} \hat{a}_i z^{-i} \tag{3}$$

In order for the resulting filter to be stable, all the roots of the polynomial $D(z)$ should lie inside the unit circle. The problem, therefore, is to determine the polynomial $D(z)$ with the smallest degree that meets the above mentioned requirements (note that in general the smallest possible value of L is M).

Let $X_{d1}, X_{d2}, \cdots X_{dL}$ be the roots of the polynomial $D(z)$ and $X_{a1}, X_{a2}, \cdots X_{aN}$ be the roots of $A(z)$ (i.e., the poles of the

0-7803-5482-6/99/$10.00 ©2000 IEEE

original filter). Then the roots of the polynomial $A_T(z)$ (i.e., the poles of the transformed filter) will be

$$X_{d1}, X_{d2}, \cdots X_{dL}, X_{a1}, X_{a2}, \cdots X_{aN}$$

Now, it can be shown that the roots of any polynomial in general, and that of $A_T(z)$ in particular, satisfy the following set of simultaneous non-linear equations,

$$\sum_{i=1}^{L} X_{di}^k + \sum_{i=1}^{N} X_{ai}^k = -\left(k\hat{a}_T + \sum_{i=1}^{k-1} \hat{a}_i P_{k-i}\right)$$
$$k = 1, \cdots, M \qquad (4)$$

where P_k is defined as the sum of the k^{th} power of the poles of $A_T(z)$

$$P_k = \sum_{i=1}^{L} X_{di}^k + \sum_{i=1}^{N} X_{ai}^k \qquad (5)$$

The above set of equations (4) can be derived using the fact that the n-th coefficient of a polynomial is equal to the sum of all products of its roots taken n at a time. Now, in the set of equations (4) the only known quantities are X_{ai}'s, the poles of the original filter transfer function. To restate, our goal is to determine the poles of $D(z)$, X_{di}'s, (and hence the polynomial $D(z)$ itself) that satisfy (4) under the constraint $|X_{di}| < 1 \;\; \forall i$ keeping the degree, L, as small as possible. Transferring the summation in (4) to the RHS we get

$$\sum_{i=1}^{L} X_{di}^k = -k\hat{a}_k - \left(\sum_{i=1}^{k-1} \hat{a}_i P_{k-i} + \sum_{i=1}^{N} X^k ai\right) = C_k$$
$$k = 1, \cdots, M \qquad (6)$$

Therefore, if we fix $\hat{a}_1, \cdots, \hat{a}_M$, the first M coefficients of $A_T(z)$, to some powers of two (or sums of powers of two), the set of equations (6) can be solved to get the desired polynomial $D(z)$. This should be done in a manner so that a solution to (6) exists within the constraints on the magnitude of X_{di}'s for as small a value of L as possible. In order to achieve this goal, the following heuristic has been used:*The smaller the value of the constants C_k's the more likely it is that a bounded solution exists for a smaller value of L.* The basic intuitive idea behind this heuristic is as follows: think of X_{di}'s as vectors in the two-dimensional complex plane each of length at most one. Now, if the constants C_k's are large then a larger number vectors (hence larger L) will be needed to satisfy the set of equations (6). Based on this heuristic and from (6) and (5) we can now fix the coefficients $\hat{a}_1, \cdots, \hat{a}_M$ to

$$\hat{a}_i = -\frac{1}{k} Q_h \left(\sum_{j=1}^{k-1} \hat{a}_j P_{k-j} + \sum_{j=1}^{N} X_{aj}^k\right)$$

$$P_k = -\left(k\hat{a}_k + \sum_{j=1}^{k-1} \hat{a}_j P_{k-j}\right) \qquad (7)$$
$$i = 1, \cdots, M \qquad k = i$$

where the function $Q_h(y)$ quantizes the argument y to a sum of at most h powers of two. (by convention $Q_0(y) \equiv 0$). Therefore, the coefficients can be determined iteratively starting from $\hat{a}_1$. The parameter h can either be fixed to a certain value depending on the level of complexity desired in the M innermost feedback loops of the filter, or, it can be increased in steps. Different choices of h will result in different architectures as will be described in Section 3.

Having determined C_k's we can now determine $D(z)$ by solving (6). Since the value of L for which a bounded solution exists is not known a-priori, we start with the smallest possible value of $L = M$. For this particular value of L, the number of variables is equal to the number of equations and a closed form expression can be derived for the coefficients of the polynomial $D(z)$

$$d_k = -\frac{1}{k}\left(C_k + \sum_{j=1}^{k-1} d_j C_{k-j}\right) \qquad (8)$$
$$k = 1, \cdots, M$$

The polynomial $D(z)$ thus obtained is checked for stability. If some roots of $D(z)$ lie outside the unit circle, then the value of L is increased by one and again an attempt to find a bounded solution is made. For $L > M$ the equation (6) can be solved in a least square sense by minimizing the mean square error function f

$$f = \sum_{i=1}^{M} e_i^2$$
$$\textit{where } e_i = \left(\sum_{j=1}^{L} X_{dj}^i\right) - C_i \qquad (9)$$
$$\textit{under the constraints } |X_{di}| < 1 \quad \forall i$$

If the constrained minimum of f, f_{min}, is almost equal to zero then a solution to (6) has been found, otherwise L is again incremented and this process is continued till a bounded solution is obtained. Thus, in this manner we can determine the smallest degree polynomial, which when multiplied to the denominator and numerator of the original transfer function will result in the desired transformation.

3. FILTER ARCHITECTURES

Figure 1 delineates an architecture for efficiently implementing the recursive section of an IIR filter using M-level SPOT transformation. The feedback loops are partitioned into three different sections as shown in figure 1. The first section consists of the innermost M_P feedback loops. Each coefficient in this section is a power of two (thus no pipelining is required) with an additional constraint that the first coefficient is zero. This is required because an additional delay element is needed to partially pipeline the adder tree described below. The second section consists of the next M_{SP} feedback loops. Each coefficient in this section is a sum of M_P powers of two. Since $M_P - 1$ additional delay elements are available in these loops therefore the skeletal multipliers (consisting of shifters and adders) required for implementing these coefficients can be fully pipelined (figure 2). The last section consists of the remaining loops composed of multipliers. Each multiplier in this section can have $M = M_P + M_{SP}$ pipelining levels. The outputs of all the three sections are then added using the partially

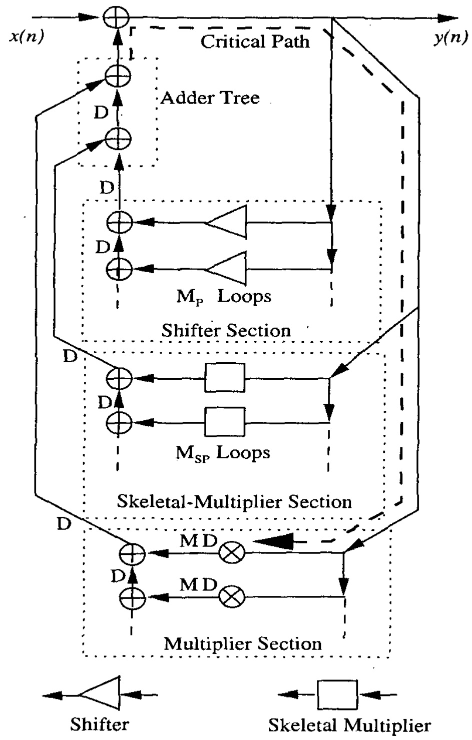

Figure 1: Architecture for implementing a filter using SPOT transform

Table I:
COEFFICIENT VALUES FOR A TENTH ORDER LOW PASS ELLIPTIC FILTER WITH STOP BAND EDGE AT 0.4 SAMPLING FREQUENCY, PASSBAND RIPPLE OF 0.5 AND STOPBAND ATTENUATION OF 40dB

$b_0 = 0.0416$	$a_0 = 1.0000$
$b_1 = -0.0221$	$a_1 = -3.6951$
$b_2 = 0.1586$	$a_2 = 9.4066$
$b_3 = -0.0374$	$a_3 = -15.9195$
$b_4 = 0.2522$	$a_4 = 21.0099$
$b_5 = -0.0292$	$a_5 = -21.0668$
$b_6 = 0.2522$	$a_6 = 16.7477$
$b_7 = -0.0374$	$a_7 = -10.1227$
$b_8 = 0.1586$	$a_8 = 4.6058$
$b_9 = -0.0221$	$a_9 = -1.4077$
$b_{10} = 0.0416$	$a_{10} = 0.2436$

Table II:
COEFFICIENT VALUES FOR $D(z)$ FOR SPOT TRANSFORMATION WITH $M_P = 6$, $M_{SP} = 0$

$d_0 = 1.0000$	$d_1 = 3.6951$
$d_2 = 4.2472$	$d_3 = -1.1450$
$d_4 = -6.6183$	$d_5 = -3.6390$
$d_6 = 4.4450$	$d_7 = 7.0404$
$d_8 = 2.1783$	$d_9 = -2.5897$
$d_{10} = -2.5277$	$d_{11} = -0.2261$
$d_{12} = 0.8708$	$d_{13} = 0.5876$
$d_{14} = 0.1385$	

pipelined added tree. All the above criterion can be easily incorporated in the SPOT transform by choosing the value of h in Q_h appropriately (Section 2).

The critical path delay of this architecture is

$$T_{critical} = 2T_{adder} + (T_{mult}/M)$$

where T_{adder} is the delay of an unpipelined adder and T_{mult} is the delay of an unpipelined multiplier. As a comparison, the critical path delay for an architecture based on the other transformations described in Section 1 (for M levels of lookahead) is

$$T_{critical} = T_{adder} + (T_{mult}/M)$$

Since, in almost all cases, the multiplier delay is much larger than the adder delay, the two techniques achieve similar speedups. In the next section we give examples to show that the filter functions derived using SPOT transforms require much less hardware overhead as compared to the other transforms.

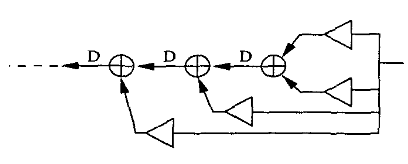

Figure 2: A fully pipelined skeletal multiplier for multiplying with a sums of powers of two coefficient

4. RESULTS

A tenth order elliptic low pass filter is taken as the original transfer function that we want to implement. The coefficients of the numerator and denominator polynomials of the original filter TF are given in Table I.

Example 1: In the first example we have assumed that $M_P = 6$ and $M_{SP} = 0$, i.e., there is no skeletal multiplier section in the filter. The coefficients of the polynomial $D(z)$ (multiplying $D(z)$ to the numerator and denominator of the original TF gives

Table III:
COEFFICIENT VALUES FOR $D(z)$ FOR SPOT TRANSFORMATION WITH $M_P = 4$, $M_{SP} = 2$

$d_0 = 1.0000$	$d_1 = 3.6951$
$d_2 = 6.2472$	$d_3 = 6.2452$
$d_4 = 4.1260$	$d_5 = 2.1349$
$d_6 = 1.2156$	$d_7 = 0.6651$
$d_8 = 0.1801$	

Filter Specifications	Transformation Parameters	Hardware Overhead (Largest Pole Radius)		
		SPOT	Scattered Lookahead	MOA
Lowpass (Elliptic) $N = 10$, $W_p = 0.4$, $R_s = 40$, $R_p = 0.5$	$M = 6$, $M_P = 4$ $M_{SP} = 2$	10 AMP + 11 A (0.97)	30 AMP (0.9981)	42 AMP (0.97)
Lowpass (ChebyshevII) $N = 8$, $W_p = 0.8$, $R_s = 40$	$M = 6$, $M_P = 3$ $M_{SP} = 3$	18 AMP + 11 A (0.92)	24 AMP (0.9247)	30 AMP (0.97)
Lowpass (Butterworth) $N = 6$, $W_p = 0.3$	$M = 6$, $M_P = 3$ $M_{SP} = 3$	14 AMP + 11 A (0.7177)	18 AMP (0.8085)	18 AMP (0.7807)
Highpass (Elliptic) $N = 6$, $W_p = 0.4$, $R_s = 40$, $R_p = 0.5$	$M = 6$, $M_P = 4$ $M_{SP} = 2$	6 AMP + 11 A (0.6727)	18 AMP (0.9698)	14 AMP (0.97)
Highpass (Butterworth) $N = 10$, $W_p = 0.3$	$M = 6$, $M_P = 3$ $M_{SP} = 3$	10 AMP + 11 A (0.8882)	30 AMP (0.8805)	18 AMP (0.8998)
Highpass (ChebyshevII) $N = 8$, $W_p = 0.4$, $R_s = 40$	$M = 6$, $M_P = 3$ $M_{SP} = 3$	6 AMP + 11 A (0.6691)	24 AMP (0.8921)	14 AMP (0.8938)
Bandpass (Elliptic) $N = 10$, $W_p = [0.5\ 0.7]$, $R_s = 30$, $R_p = 0.5$	$M = 6$, $M_P = 4$ $M_{SP} = 2$	14 AMP + 11 A (0.97)	30 AMP (0.9849)	30 AMP (0.9695)
Bandstop (Elliptic) $N = 6$, $W_p = [0.5\ 0.6]$, $R_s = 40$, $R_p = 0.5$	$M = 6$, $M_P = 4$ $M_{SP} = 2$	6 AMP + 11 A (0.8090)	18 AMP (0.9623)	22 AMP (0.97)
Bandstop (ChebyshevII) $N = 8$, $W_p = [0.1\ 0.2]$, $R_s = 40$	$M = 6$, $M_P = 4$ $M_{SP} = 2$	22 AMP + 11 A (0.9676)	24 AMP (0.9540)	34 AMP (0.97)

Table IV: SUMMARY OF RESULTS

the transformed TF) are given in Table II. It can be seen that 22 additional terms are introduced into the filter TF (14 in the numerator and 8 in the denominator) as a result of the transformation and 22 additional multiplier-adder pairs are required to implement these additional terms. In addition $(M_P - 2) + (M_{SP} - 1) + 2 = 5$ adders are required (in the shifter section, skeletal multiplier section and the adder tree respectively).

Example 2: In the second example the parameters used are: $M_P = 4$ and $M_{SP} = 2$. The coefficients of the polynomial $D(z)$ are given in Table III. In this case the number of additional terms introduced are 12, out of which 2 are skeletal multiplies. Therefore, the additional hardware required is: 10 multiplier-adder pairs and 6 adders for the two skeletal multipliers (each coefficient in the skeletal multiplier section is a sum of upto four powers of two) and 5 more adders (as described in Example1).

For the purpose of comparison we have also implemented a transformation technique similar to minimum order augmentation [2]. For six levels of lookahead (thus similar speedup as in the examples above) minimum order augmentation requires 42 additional terms (42 extra multiplier-adder pairs in hardware) and scattered lookahead with 3x2 decomposition requires 30 additional terms.

Similar experiments were performed for a number of other filters with different specifications. These results are summarized in Table IV. The first two columns give the filter specifications (N=Filter order, R_s=Stopband attenuation, R_p=Passband Ripple, W_p=Cutoff frequency) and the transformation parameters respectively. The last three columns give the hardware overhead for each transformation (AMP:adder-multiplier pairs, A:adders, MOA:Our implementation of minimum ordered augmentation). The figure in brackets in each of the last three columns is the largest magnitude of the extra poles added to the filter because of the transformation. This parameter is important because a filter whose poles are nearer to the unit circle will be more sensitive to coefficient word length truncation. It can be seen that, in almost all cases, the radius of the largest magnitude pole added to the original transfer function by the SPOT transform is less than or equal to that in the other transformation techniques.

5. CONCLUSIONS

In this paper we have presented the SPOT algorithm transformation for overcoming the difficulties in pipelining IIR filters. We have shown through examples that this technique can result in similar speedups as compared to the lookahead based techniques but requiring much less hardware overhead. The same examples also show that the SPOT transform does not result in higher coefficient sensitivity to rounding.

6. ACKNOWLEDGMENT

The results presented in this paper were developed as part of the term project for ECE497NS:VLSI in Signal Processing and Communications (Instructor: N. Shanbhag) during Fall 1998 semester.

7. REFERENCES

[1] K.K. Parhi and D.G. Messerchmitt, "Pipeline interleaving and parallelism in recursive digital filters," *IEEE Trans. Acoust. Speech and Signal Proc.*, vol 37, pp. 1099-1117, July 1989.

[2] Y. Lim and B. Liu, "Pipelined recursive filter with minimum order augmentation," *IEEE Trans. Signal Processing*, vol. 40, pp. 1643-1651, J uly 1992.

[3] H.H. Loomis and B. Sinha, "High Speed recursive digital filter realization," *Circuits, Systems Signal Processing*, vol. 3, pp. 267-294, Sept. 1984.

[4] Lim, Yong-Ching. Yang, Rui. Li, Dongning. Song, Jianjian, "Signed power-of-two (SPT) term allocation scheme for the design of digital filters," *Proceedings - IEEE International Symposium on Circuits and Systems*, v 5 1998.