ARITHMETIC TRANSFORMATIONS FOR INCREASED MAXIMAL SAMPLE RATE OF BIT-PARALLEL BIRECIPROCAL LATTICE WAVE DIGITAL FILTERS

Henrik Ohlsson, Oscar Gustafsson, and Lars Wanhammar

Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden
E-mail: {henriko, oscarg, larsw}@isy.liu.se

ABSTRACT

In this paper we propose an arithmetic transformation of first-order allpass lattice wave digital filter sections implemented with bit-parallel, carry-save arithmetic which increases the maximum sample rate. Such filter sections are the critical component of bireciprocal lattice wave digital filters which are efficient filter structures for sample rate conversion with factors of two. The proposed transformation reduces the sample period bound with up to 33% for CSDC coefficients. This is obtained with a coefficient with four nonzero bits, and, dependent on the value of the adaptor coefficient, at the expense of up to 28% more carry-save adders. An example is presented where an adaptor with a coefficient with two nonzero bits is implemented. This results in a reduction of the sample period bound with 25% with a 14% increase of the number of carry-save adders needed.

1. INTRODUCTION

Bireciprocal lattice wave digital filters (BLWDFs), or half-band filters, is a class of digital filters suitable for efficient implementation of interpolators and decimators for sample rate conversion with factors of two using the polyphase representation of the transfer function [1]. This results in a structure where all filtering may be performed at the lower sample rate. Compared to FIR filters, BLWDFs have a low arithmetic complexity which yield low power consumption and low chip area. However, since BLWDFs are recursive structures there is a bound on the maximal sample rate [2]. Thus, to use BLWDFs in high speed applications this bound should be increased. Also, any excess speed obtained can be traded for reduced power consumption through power supply voltage scaling [3].

A first-order allpass lattice wave digital filter section, which is the limiting recursive component in a BLWDF, is shown in Fig. 1 where the loop limiting the sample rate is indicated. For a polyphase representation of the BLWDF for interpolation and decimation we only have to consider filter sections with one delay element in the recursive loop though there are two delay elements in the original structure.

The maximal sample rate is determined by the latency of the operations in the loop divided by the number of delay elements in the loop. For the first-order lattice wave digital filter section the minimum sample period, T_{min}, is

$$T_{min} = \frac{2T_{add} + T_{mult}}{1} \qquad (1)$$

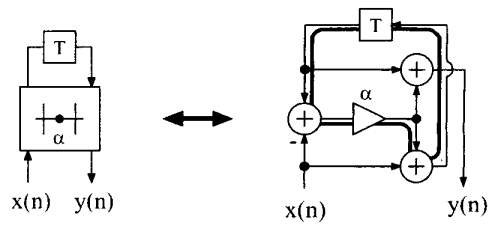

Figure 1. Sample rate bound for first-order filter section with a symmetric two-port adaptor.

where T_{add} is the latency for an addition and T_{mult} is the latency for a multiplication. The main contributor to this bound is T_{mult}. For a bit-parallel implementation the latency of the multiplication depends on the number of partial products to be added. Each partial product is generated by a nonzero bit in the coefficient. Thus, to reduce T_{min} for a bit-parallel implementation, the number of nonzero bits in the coefficient word should be as low as possible. This is obtained for CSDC coefficients. The average number of nonzero bits in a CSDC number is approximately $W_{coeff}/3$ where W_{coeff} is the coefficient word length, compared to $W_{coeff}/2$ for the binary representation. Thus, CSDC representation yields a significant reduction of the number of nonzero bits in a coefficient [4].

One method for reducing the number of nonzero bits in the coefficient further is to cascade several filters and thereby reduce the stopband requirement on each filter stage [5]. This results in shorter coefficient word lengths for each filter stage which yields a reduced number of nonzero bits. Efficient cascaded polyphase structures for decimation and interpolation where the filtering is performed at the lower sample rate have been proposed in [6].

The sample period bound may be reduced even further by arithmetic transformations of the filter structure. In [7] some transformation performed on a bit-serial first-order lattice wave digital filter structure that reduces the latency of the critical loop with one addition is presented. In this paper similar arithmetic transformations are applied on such filter structures implemented with bit-parallel arithmetic.

2. CARRY-SAVE IMPLEMENTATION

Carry-save arithmetic, which is a redundant arithmetic, is efficient for implementation of many DSP algorithms since time consuming carry propagation may be avoided [8]. One such case, where carry-save arithmetic is especially efficient, is the implementation of wave digital filters [9]. The use of

0-7803-6685-9/01/$10.00©2001 IEEE

carry-save adders (CSAs) instead of carry-propagation adders reduces the latency for the computations in the loop significantly.

A CSA adds three operands and yields as result a sum vector and a carry vector. The latency for a CSA is equal to the latency of one full adder. CSAs can be combined into adder trees which are efficient for adding several operands with a low overall latency. This is efficient for implementation of multiplications where several number of partial products need to be added. A tree with a minimum height is the Wallace tree [10]. A drawback with the Wallace tree is that it yields an irregular routing, which has a number of drawbacks, e.g., irregular wire routing, which causes different loads on the full adders and glitching which contributes to large power consumption. Instead an overturned-stair adder tree may be used. Such a tree has a regular routing and has the same height as a Wallace tree when adding up to 18 inputs [11]. The latency for a multiplier based on a Wallace tree is dependent on the height of the adder tree, i.e., the number of partial products in the coefficient. The latency for such multipliers can therefore be expressed as a number of T_{add}.

To complete a multiplication implemented with carry-save arithmetic a vector merging adder (VMA) is needed at the output of the adder tree to merge the sum and the carry vector. However, when implementing the symmetric two-port adaptor with CSAs, the VMA can be placed outside the recursive loop, at the nonrecursive output of the adaptor. Hence, no carry propagation is required in the recursive part and T_{min} can be expressed as a number of T_{add}. It is also possible to pipeline the VMA to reach T_{min}. However, implementing the adaptor with CSAs means that the data inside the adaptor is represented with two data vectors. This makes it difficult to identify if an overflow occurs. Overflow detection and correction is necessary in order to guarantee overflow stability for wave digital filters [12]. In [13] a scheme for detection and correction of overflows that guarantee the stability in carry-save implementation of WDFs was introduced.

Another difference between a carry-propagation implementation and a carry-save implementation is the possible occurrence of carry overflows. When performing a carry-save addition there is a possibility for an error to occur in the final sum due to the discarding of carry bits. A solution for this has been proposed in [8] where the full adder used for the most significant bits of each CSA can be modified to correct any possible errors in the carry bits. This can be done without introduction of any extra delay in the full adder.

3. PROPOSED IMPLEMENTATION

The symmetric two-port adaptor used in the first order allpass section and shown in Fig. 1 is described with the following equations

$$B_1 = \alpha(A_2 - A_1) + A_2 \tag{2}$$

$$B_2 = \alpha(A_2 - A_1) + A_1 \tag{3}$$

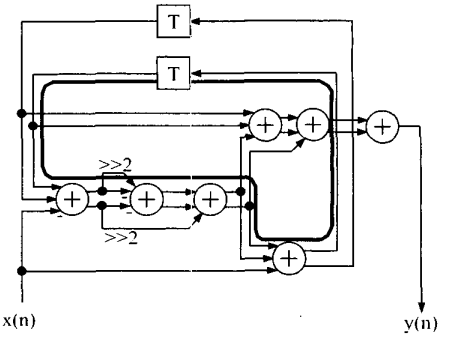

Figure 2. CSA implementation of the original structure for $\alpha = -0.75$.

Implementation of the adaptor with carry-save arithmetic using these equation yields a structure as shown in Fig. 2, where $\alpha = -0.75 = -1.01_{CSDC}$. From this figure we also see that the operations in the critical path consists of two carry-save adders and a Wallace tree for the multiplication. In this example we obtain a T_{min} of $4T_{add}$.

One method to reduce the latency for the critical loop would be to combine all additions in the loop into one Wallace tree. This is possible by rewriting (2) and (3) as

$$B_1 = (1 + \alpha)A_2 - \alpha A_1 \tag{4}$$

$$B_2 = (1 - \alpha)A_1 + \alpha A_2 \tag{5}$$

This is a numerically equivalent representation of the adaptor. Thus, the method used for detection and suppression of overflows in a carry-save WDF previously discussed is still valid. The reordering of the arithmetic operations and the quantization operation in the adaptor corresponding to (4) and (5) is shown in Fig. 3. When implementing the transformed adaptor using carry-save arithmetic T_{min} may be reduced with up to two T_{add}. The cost for the 1-α multiplication depends on the number of nonzero bits in the 1-α coefficient. This number is, depending on the value of α, either equal, one more, or one less than the number of nonzero bits in α.

In Fig. 4 the same filter section as in Fig. 2 has been implemented using the proposed structure. In this example, where we have two nonzero bits in the coefficient, we need to increase the height of the adder tree for the α multiplication with one to add the result from the 1-α multiplication. Thus, for this example, T_{min} is reduced with one T_{add} to $3T_{add}$ compared to the original structure. In this case there is no need for a VMA in the 1-α multiplication since there are two inputs available in the adder tree in the loop. The number of carry save adders used for the proposed structure is seven compared to six for the original structure. Thus, a reduction of T_{min} with 25% is obtained with a 14% increase of the number of CSAs for $\alpha = -0.75$.

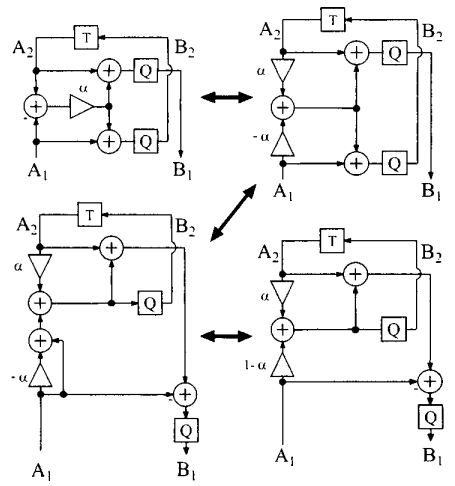

Figure 3. Reordering of operations in a first-order filter section.

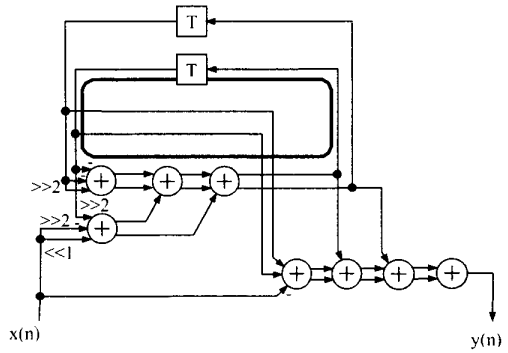

Figure 4. CSA implementation of the proposed structure for $\alpha = -0.75$.

4. RESULTS

The proposed structure for implementation of first-order all-pass LWDF sections has been evaluated for the number of carry-save adders needed and for the difference in T_{min} for different CSDC coefficients. The evaluation presented in this paper considers CSDC coefficients with up to 10 nonzero bits. With 10 nonzero bits in the CSDC coefficient, all coefficients with word lengths of up to 19 bits can be represented.

Table 1 presents T_{min} for both the original structure and for the proposed structure. The later has been evaluated both with and without a vector merging adder at the output of the 1-α multiplication. This only makes a difference when we have four, six or nine nonzero bits in the coefficient. This is because the Wallace tree for the α multiplication for such α has one free input only and the height of the adder tree for the multiplication must then be increased by one if there are two extra inputs in the adder tree.

In Table 2 the number of carry-save adders needed for the original and for the proposed structure is compiled. The number of CSAs needed gives an estimation of the extra hardware resources needed for the proposed structure. Also, the total number of CSAs is dependent on the value of the coefficient. For $2/3 < \alpha < 1$, the coefficient 1-α consists of one nonzero bit less than α. For $-1 \le \alpha < -2/3$ and $1/3 < \alpha < 2/3$ the number of nonzero bits are equal in α and 1-α. Finally, for $-2/3 < \alpha < 1/3$ one more nonzero bit appears in 1-α than in α. The results presented in Table 1 and Table 2 are also valid for overturned stair adder trees for coefficients with up to eight nonzero bits.

Figure 5 shows a comparison of the reduction of T_{min}, both for the case with a VMA at the input multiplication and for the case without a VMA. The case with a single nonzero bit in the coefficient seems to be much more efficient, with a 50% re-

Number of nonzero bits in coefficient	Original	Proposed (without input VMA)	Proposed (with input VMA)
1	2	1	1
2	4	3	3
3	5	4	4
4	6	5	4
5	7	5	5
6	7	6	5
7	8	6	6
8	8	6	6
9	8	7	6
10	9	7	7

Table 1. Comparison of T_{min} (in T_{add}) between the original and the proposed structure.

Number of nonzero bits in coefficient	Original	Proposed $\frac{2}{3} < \alpha < 1$	Proposed $-1 \le \alpha < -\frac{2}{3}$, $\frac{1}{3} < \alpha < \frac{2}{3}$	Proposed $-\frac{2}{3} < \alpha < \frac{1}{3}$
1	4	4	5	6
2	6	6	7	8
3	8	9	10	11
4	10	12	13	14
5	12	15	16	17
6	14	18	19	20
7	16	21	22	23
8	18	24	25	26
9	20	27	28	29
10	22	30	31	32

Table 2. Comparison of CSAs needed for the original and the proposed structures.

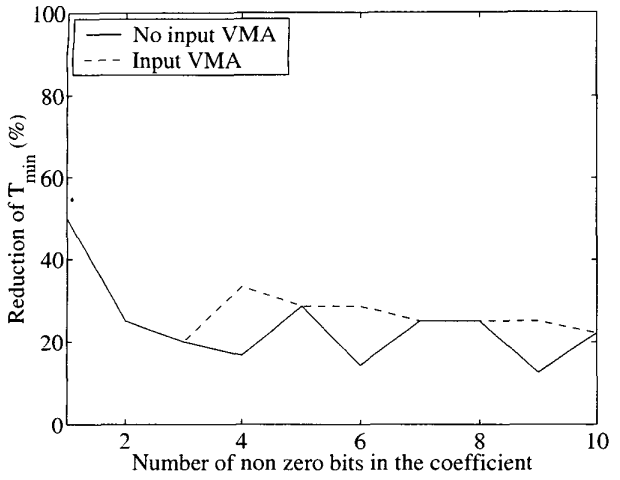

Figure 5. Reduction of T_{min} compared to the original solution.

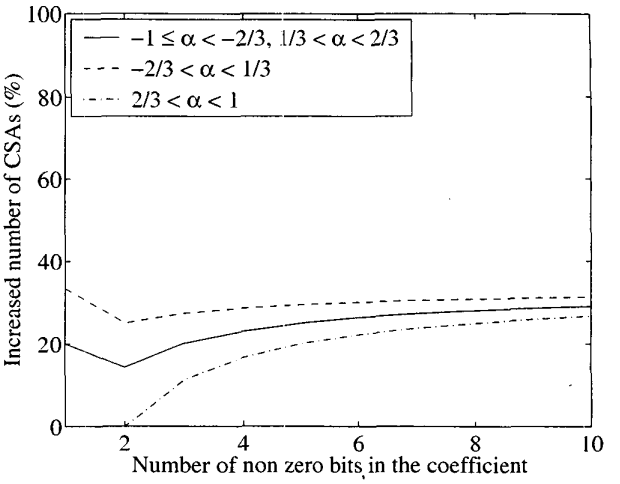

Figure 6. Increased number of CSAs compared to the original solution.

duction of T_{min} and no increased area. However, for that case the needed pipeline registers in the nonrecursive parts should be considered as well, yielding a larger area. For the other cases, a reduction of T_{min} with up to 33% can be obtained. This is the case when we have four nonzero bits in the coefficient and we use a VMA in the 1-α multiplication.

In Fig. 6 the cost for the proposed structure compared to the original structure, measured in carry-save adders, is presented. We can see that the extra adders needed increases as the number of nonzero bits in the coefficient increases. We can also see the influence the value of α has on the cost.

5. CONCLUSIONS

In this paper we have presented an arithmetic transformation of a first order lattice wave digital filter section for reduction of the sample period bound. The transformation has been evaluated for implementation with carry-save arithmetic. The evaluation gave that the sample period bound may be reduce with up to 33% compared to the original structure depending on the number of nonzero bits in the coefficient. The reduction of T_{min} is at the expense of some extra hardware. This amount depends on the value of the adaptor coefficient as well as the number of nonzero bits in the coefficient.

An example of a first order allpass filter section was presented with an adaptor coefficient of $\alpha = -0.75 = -1.01_{CSDC}$. This resulted in a reduction of the sample period bound by 25% with a 14% increase of the number of carry-save adders.

6. REFERENCES

[1] Vaidyanathan P.P.: *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[2] M. Renfors and Y. Neuvo, "The maximum sampling rate of digital filters under hardware speed constraints," *IEEE Trans. Circuits Syst.*, vol. 28, pp 196–202. Mar. 1981.

[3] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, no. 4, pp. 498–523, April 1995.

[4] L. Wanhammar, *DSP Integrated Circuits*, Academic Press, 1999.

[5] J. Yli-Kaakinen and T. Saramäki, "Design of low-sensitivity and low-noise recursive digital filters using a cascade of low-order lattice wave digital filters," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 7, pp. 906–914, July 1999.

[6] H. Johansson and L. Wanhammar, "High-speed recursive filter structures composed of identical all-pass subfilters for interpolation, decimation and QMF banks with perfect magnitude reconstruction," *IEEE Trans. Circuits Syst. II*, vol. 46, pp 16–28, Jan. 1999.

[7] K. Palmkvist, M. Vesterbacka, and L. Wanhammar, "Arithmetic transformations for fast bit-serial VLSI implementations of recursive algorithms," *NORSIG 96*, Espoo, Finland, Sept. 24-27, 1996, pp. 391–394.

[8] T. G. Noll, "Carry save architectures for high-speed digital signal processing," *J. VLSI Signal Processing*, vol. 3, pp. 121–140, 1991.

[9] J. Pihl, "Design automation with the TSPC circuit technique: a high-performance wave digital filter," *IEEE Trans. VLSI Syst.*, vol. 8, no. 4, pp 456–460, Aug. 2000.

[10] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comp.*, vol. 13, pp. 114–117, Feb. 1964.

[11] Z. Mou and F. Jutand, "'Overturned-stairs' adder trees and multiplier design," *IEEE Trans. Computers*, vol. 41, pp. 940–948, Aug. 1992.

[12] Fettweis A.: "Wave digital filters: Theory and practice," *Proc. IEEE*, vol. 74, no2, pp. 270–327, Feb. 1986.

[13] U. Kleine and T. G. Noll, "On the forced response stability of wave digital filters using carry-save arithmetic," *Arc. Elektr. Übertragungstechnik (AEÜ)*, vol. 41, no. 6, pp. 321–324, Nov/Dec. 1987.