Low power Comb Decimation Filter Using Polyphase Decomposition For Mono-Bit $\Sigma\Delta$ Analog-to-Digital Converters

Y. Dumonteix, H. Aboushady, H. Mehrez and M. M. Louërat
Université Paris VI, Laboratoire LIP6
4 Place Jussieu, 75252 Paris Cedex 05, France
Yannick.Dumonteix@lip6.fr , Hassan.Aboushady@lip6.fr

Abstract

A power efficient multirate multistage Comb decimation filter for monobit $\Sigma\Delta$ A/D converters is presented. Polyphase decomposition in all stages with high decimation factor in the first stage, are used to reduce the frequency of the input signal. Several implementations have shown that proper choice of the decimation factor of the first stage can reduce power consumption by more than 30% The multistage architecture makes the total decimation factor easily programmable and suitabale for multi-standard applications.

1 Introduction

Comb filters, shown in Fig.1(a), are widely used in the decimation filter of $\Sigma\Delta$ A/D converters. These filters operate at maximum sampling frequency before any decimation takes place. The power consumption of Comb filters is then very high. The transfer function $H(z)$ of a Comb filter of order k and for a decimation ratio M is defined by

$$H(z) = \left(\frac{1 - z^{-M}}{1 - z^{-1}}\right)^k. \qquad (1)$$

These filters are usually implemented using the IIR-FIR technique [1][2]. In this case, the Comb filter, which is actually an FIR filter, is decomposed into two filters: an IIR filter $H_1(z) = [1/(1 - z^{-1})]^k$ and an FIR filter $H_2(z) = [1 - z^{-M}]^k$. Using the commutative rule [2], the second filter is transferred after decimation, Fig1(b), and then operates at a much lower sampling frequency $\frac{f_s}{M}$. To ensure stability, the minimum wordlength at the input of the IIR filter is fixed to $(B_{in} + k\ log_2 M)$ bits, where B_{in} is the number of bits at the input [1]. The major drawback of this architecture is that the IIR filter is operating at maximum sampling frequency f_s and with a large wordlength. This increases drastically power consumption and limits the highest operating frequency of the decimation filter.
Equation (1) can be written in the following form:

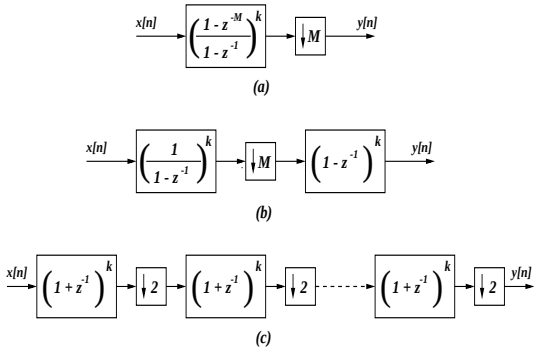


Fig. 1: (a) Comb Filter. (b) IIR-FIR Implementation. (c) Cascade of FIR decimating by 2.

$$H(z) = \prod_{i=0}^{(log_2 M)-1} \left(1 + z^{-2^i}\right)^k. \qquad (2)$$

Applying the commutative rule on this representation gives, as shown in Fig.1(c), $log_2 M$ cascade of identical FIR filters $H_i = [1 + z^{-1}]^k$, each followed by a decimation factor of 2. This kind of structure has the advantage of not having any stability problems and at the output of each stage the wordlength of each filter stage is limited to $(B_{in} + k\ i)$ bits. It has been shown in [3], that for high order and high decimation ratio Comb filters, the FIR-cascade implementation consumes less power than the IIR-FIR implementation. Polyphase decomposition [4] can be used to bring the decimation by 2 to the input of each filter [5]. Thus reducing the operating frequency of the cascade filter number i to $\frac{f_s}{2^i}$.

In this paper, we present a different representation of the Comb filter. This representation allows us to exploit the polyphase decomposition in order to perform higher decimation factors at the input of the first stage. Although coefficients, resulting from this decomposition, require expensive multiplication operations and larger wordlength, the overall power consumption is lower. This is due to the significant reduction of the operating frequency. We show that, an optimum decimation factor exists that compromises the added com-

plexicity of the polyphase decomposition with the reduction of the operating frequency.

In the first section, we show how we propose to apply polyphase decomposition with high decimation ratio in the first stage. The second section is dedicated to the filter implementation. We present also a method to estimate the power consumption, area and sampling frequency. We conclude, in the two last sections, by a comparative study on the effect of the decimation factor of the first stage on the overall circuit performance.

2 Polyphase Comb Decimation Filters

As shown in Fig.2, we propose to decompose the Comb decimation filter into a first stage FIR filter $H_1(z)$ with a decimation factor M_1, followed by a cascade of FIR $[1+z^{-1}]^k$ filters with a decimation factor 2. The reason behind choosing this representation is that we would like to decimate as much as possible in the first stage. The following stages are kept with the minimum decimation ratio 2 because, as will be shown later, when the wordlength of the input signal is high, reducing the sampling frequency does not compensate for the added complexity of the polyphase decomposition. In the following, we will explain how polyphase decompostion is applied to Comb decimation filters. Equation (1) can be written in the following form:

$$H(z) = H_1(z)\ \ H_2(z) \tag{3}$$

where,

$$H_1(z) = \left(\sum_{i=0}^{M_1-1} z^{-i}\right)^k$$

$$H_2(z) = \prod_{i=0}^{(log_2 \frac{M}{M_1})-1} \left(1+z^{-2^i}\right)^k.$$

The expansion of $H_1(z)$ results in an FIR filter of order $k(M_1-1)$

$$H_1(z) = \sum_{n=0}^{k(M_1-1)} h(n) z^{-n}. \tag{4}$$

The coefficients of this filter are integers and symmetrical $h(n) = h(N-1-n)$, where $N = k(M_1-1)$. Applying polyphase decomposition on the filter of equation (4), we get

$$\begin{aligned} H_1(z) &= \sum_{n=0}^{k(M_1-1)} h(nM_1) z^{-nM_1} \\ &= z^{-1} \sum_{n=0}^{k(M_1-1)} h(nM_1+1) z^{-nM_1} \qquad (5) \\ &\vdots \\ &= z^{-(M_1-1)} \sum_{n=0}^{k(M_1-1)} h(nM_1+M_1-1) z^{-nM_1}. \end{aligned}$$

Efficient polyphase implementation of $H_1(z)$ is shown

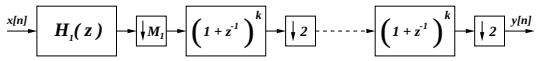


Fig. 2: Cascade of FIR with High Decimation Factor in the 1st Stage.

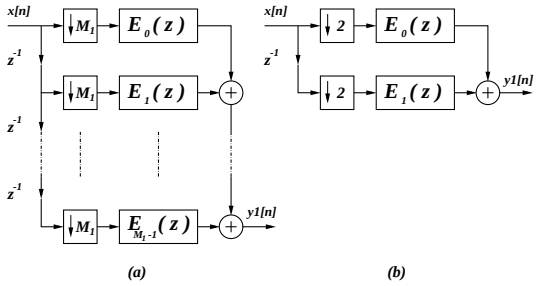


Fig. 3: (a) Polyphase Decomposition of the 1st Stage Decimation Filter $H_1(z)$. (b) Polyphase Decomposition of Subsequent Filters decimating by 2.

in Fig.3(a). As we can see, decimation takes place before filtering, so multiplications and additions are performed at a sampling frequency M_1 times lower than the frequency of the input signal. The subsequent filters decimating by 2 are nothing but a special case of the general case described above. The implementation of these filters is shown in Fig.3(b).

To first order, the average power consumption, P, of a digital signal processing system may be expressed as

$$P \ \propto \ \sum_j \ N_j \ W \ f_s \tag{6}$$

where N_j is the number of operations of type j (corresponding to addition or multiplication) performed per sample, W the wordlength and f_s is the sample frequency. In IIR-FIR and FIR-cascade implementations, we have a low number of operations. These filters do not require any multiplications because all the coefficients are equal to 1. On the other hand a large part of the filtering operations occur at high frequency. Polyphase decomposition introduces more complex coefficients into the filter, but the reduction of the sampling frequency can reduce the overall power consumption. Note that the wordlength at the output of each subfilter is $B_{in} + klog_2 M_1$.

In order to find the decimation factor M_1 that achieves minimum power consumption, several implementations with different values for M_1 have been implemented and are presented in the following sections.

3 Filter Implementation

3.1 General Architecture

The choice of the FIR architecture to implement the polyphase filters has an important impact on power consumption. FIR filters are implemented either in a

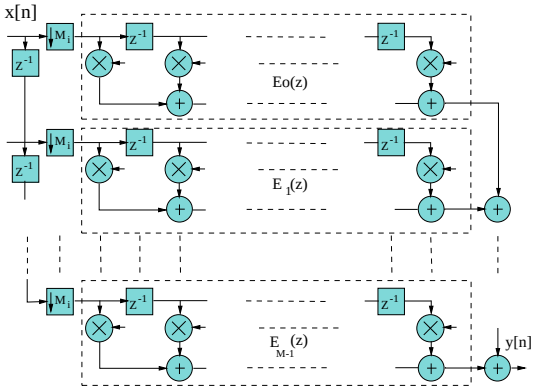


Fig. 4: General Direct-Form Architecture for One Stage of the Comb Decimation Filter.

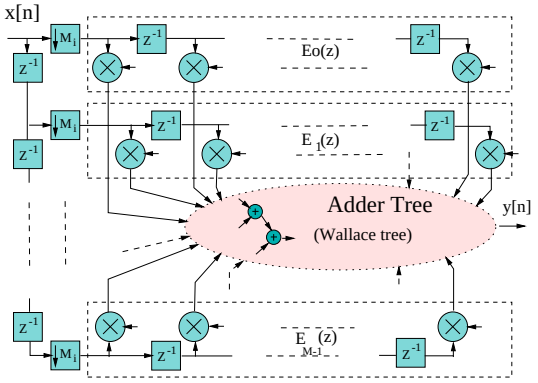


Fig. 5: Adder Tree Compaction Using Wallace Tree.

direct-form or a transposed-form. Each of these two forms has some drawbacks :

- The direct-form has a long critical path which limits the maximum sampling frequency of the filter. The number of combinational logic stages is high which increases the glitches impact on power comsumption.
- The transposed-form requires larger word-length for the intermediate registers, which increase the power comsumption and the area. The fact that the input signal is distributed over a large number of multipliers increases power consumption due to high input capacitance and complex routing.

Since the use of polyphase decomposition has highly reduced the operating frequency of the filter, the critical path is not a problem anymore. Thus we have chosen the direct-form implementation. Both architectures have been implemented. Simulation results have confirmed that the direct structure is better in terms of power consumption.

Fig.4 shows that the general architecture for one stage of the comb decimation filter. All the subfilters E_0, E_1, ..., E_{M_1-1} resulting from the polyphase decomposition, operate at the same sampling frequency.

One way of reducing the required hardware, to implement these filters, is to gather all additions from the different subfilters into one adder tree Fig.5. This adder tree is also used in the multipliers to sum all the partial products. In fact, partial products resulting from different multiplications can be gathered with the addition operation in the same adder tree. The Wallace tree [6] is an efficient realization of the adder tree. This technique is usually used in the implementation of high speed multipliers [6, 7]. Note that, we have only one Wallace tree for each stage of the complete polyphase filter. This has significantly reduced the overall power consumption.

Although each subfilter resulting from the polyphase decomposition is not always symmetric, the method described above has allowed us to exploit the symmetry that exists between the coefficients of the different subfilters.

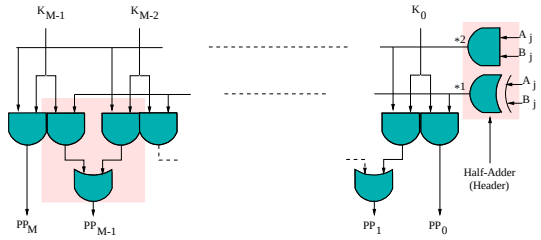
3.2 Multipliers : Partial products calculation

An analysis of existing multipliers shows that the power comsumption, the critical path and the surface are directly related to the number of partial products to be added. We must therefore focus on limiting the number of terms to sum. The partial products are reduced by using the symmetry in the Comb filters and by considering that the coefficients are constants.

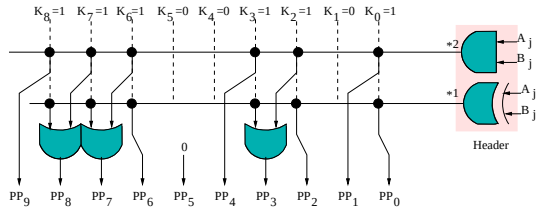
The symmetry of the coefficients makes it possible to divide by two the number of multiplications. For that, it is just necessary to add both multiplied operands by the same coefficient before the multiplication : $A*K+B*K=(A+B)*K$; where K is the coefficient $h[n]$, A the input with delay n and B the input with delay $N-1-n$ before applying the polyphase decomposition.

The addition $(A+B)$ can be performed directly in the multiplication by using the technique presented in [7]. This technique is illustrated in Fig.6(a). In this figure, the calculation of a general partial product line : $(A_j+B_j)*K$, is depicted. The idea is based on the decomposition of each pair A_j and B_j, into two signals : $*1$ and $*2$. Each decomposition requires one Half-Adder. The multiplication with K is carried out at the same time with $*1$ and $*2$ (AND gates). As these two signals are exclusives, the two partial product lines can be recombined in only one line (OR gates). The number of partial products is divided by two.

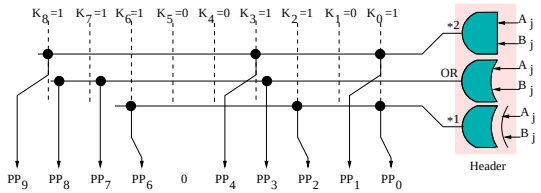
Since the coefficients K are constants, all the AND gates can be replaced either by zero, by $*1$ or by $*2$. The example $K=111001101_{|b}$ is given Fig.6(b). The OR gates can be replaced by one single gate in the header, as shown in Fig.6(c). Apart from the headers, this technique does not require any hardware to produce the matrix of partial products. For each pair of



(a) Principle



(b) Example with $K = 111001101$



(c) Degradation and Header

Fig. 6: Multiplication by a Constant K : Partial Product Line $(A_j + B_j) * K$.

$\downarrow M_1$	max size coeff. (Bit)	coeff. number	W_o	direct impl. (PP_1)	symmetric impl. (PP_1)	
2	4	6	6	10	10	-
4	8	16	11	50	41	(-18%)
8	12	36	16	178	139	(-22%)
16	16	76	21	440	346	(-21%)
32	20	156	26	1184	919	(-22%)

Table 1: Reduction of the Number of Partial Products for the First Stage of decimation.

multiplication : $(A + B) * K$, the number of partial products is equal to $W * PPL$ where W is the input wordlength and PPL the number of partial products for a partial product line : $(A_j + B_j) * K$. For a stage i of decimation, the total number of partial products is : $PP_i = W_i * pp_i$, where W_i is the input wordlength and pp_i is the sum of the PPL_k for all the coefficients K_i.

3.3 Analysis and Estimation of Performances

Table 1 presents the evolution of the number and the maximum wordlength of the coefficients, the wordlength of the output (W_o), as well as the number of generated partial products : PP_1, for the first stage of decimation. The values are given in function of the decimation factor : M_1. PP_1, is given for a direct implementation of the multiplications, and an implementation taking advantage of the symmetry of the coefficients. The average gain is 21%. This gain is not more significant because the direct implementation can also profit from the fact that for each zero bit in the coefficients, the partial product is zero.
This architecture gives similar results as an implementation using the radix 4 Booth algorihm to decompose the different coefficients.

Comb filter Polyphase decompo.	Number of Partial Products (PP) Stage 1	2	3	4	5	A	PWR	f_s
2 2 2 2 2	10	60	110	160	210	550	50	0.60
4 2 2 2	41	110	160	210		521	40	0.75
8 2 2	139	160	210			509	33	1.11
16 2	346	210				566	28	1.90
32	919					919	29	3.25

Table 2: Delay, Area and Power Consumption Estimations for $M = 32$ and a fifth order filter

In order to estimate the power consumption (PWR), the area (A) and the sampling frequency (f_s), we shall assume that the harware required for the multiplications is dominant. In this case, the number of partial products can be used as a basis for the estimation. The PWR, A and f_s equations are :

$$PWR = \sum_{i=1}^{S} \left(\frac{PP_i}{\prod_{j=1}^{i} M_j} \right) \quad , \qquad A = \sum_{i=1}^{S} PP_i$$

$$f_s = Min \left[\frac{M_1}{log_2(PP_1)}, ..., \frac{\prod_{j=1}^{i} M_j}{log_2(PP_i)}, ..., \frac{M}{log_2(PP_S)} \right]$$

where S is the number of stages of decimation and $log_2(PP_i)$ is an estimation of the number of combinational logic stages necessary for the sum of PP_i.
The evaluations show that the higher decimation must be carried out in the first stage. The value on M_1 depends of M and the filter order.
The table 2 summarizes the various results for a total decimation of 32 and a fifth order filter. The growth of the input of each stage is 1, 6, 11, 16 and 21 bits, in the case of the decimation 2 2 2 2 2. The best results are obtained for a first decimation factor $M_1 = 16$, worst performances for $M_1 = 2$.
The evaluation technique described above is general and independant from the partial product calculation method. It only requires the knowledge of the number and the distribution of the partial products.

4 Simulation Results

To study the effect of the decimation factor of the first stage M_1 on the overall performance of the circuit, Comb filters with a decimation factor of 32 for monobit $\Sigma\Delta$ A/D converters have been implemented. These filters have been realized using the polyphase decomposition described in section 2 and with the architecture described in section 3.

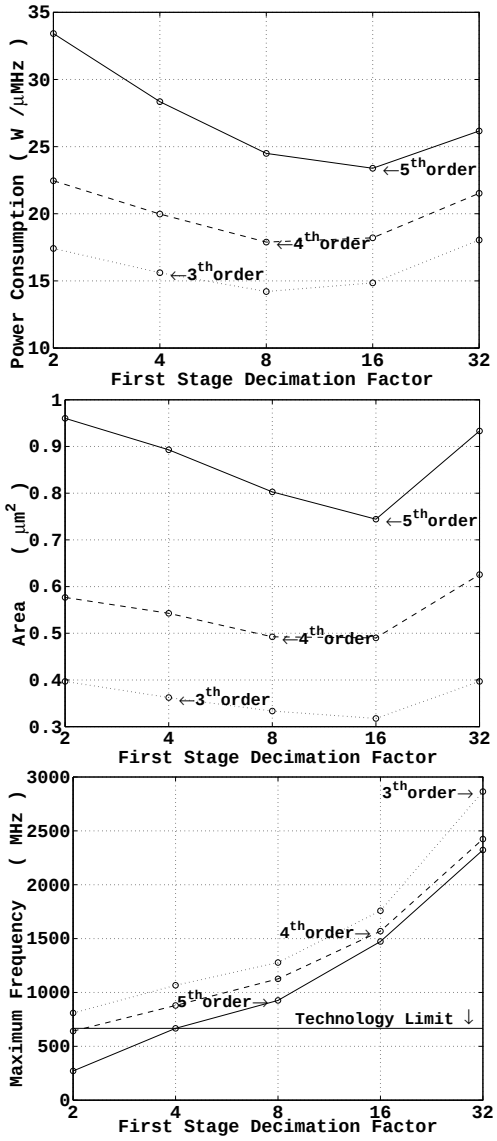


Fig. 7: Performances of the Polyphase Comb Filter in Function of the Decimation Factor of the 1st stage

Comb filters of order 3, 4 and 5 have been realized, each with a different decimation factor $(M_1 = 2, 4, 8, 16, 32)$ in the first stage. The circuits have been implemented using a 0.35 μm low cost standard technology. Three criteria have been chosen for comparison : power consumption, area and maximum sampling frequency. Simulation results for monobit $\Sigma\Delta$ A/D are shown in Fig.7. Minimum power consumption and area are achieved for a decimation factor $M_1 = 16$. The worst performances are obtained for $M_1 = 2$. These results confirm the estimations section 3. Comparing the two implementations for $M_1 = 16$ and $M_1 = 2$:

- the power consumption is reduced between 20% and 30% depending on the filter order.
- the area is reduced between 15% and 20%.
- the sampling frequency is multiplied between 3 and 5 times. In fact, the sampling frequency is limited by the intrinsic propagation delay of the D Flip Flop.

Although the main purpose of this method was to achieve low-power consumption, significant improvements regarding area and maximum sampling frequency have also been obtained.

5 Conclusion

A low-power implementations of a Comb decimation filter for monobit $\Sigma\Delta$ A/D converters have been presented. A multi-stage polyphase structure with maximum decimation factor in the first stage has been used. The proper choice of the decimation factor of the first stage can significantly improve the power consumption, area and maximum sampling frequency.
Gathering all the partial products additions into one adder tree has greatly reduced the power consumption and the hardware required for the circuit. For further improvement of the performances, a technique based on the symmetry in the filter have been proposed to implement the different multiplications.
We also presented a method to estimate the power consumption, area and sampling frequency.
This technique and the proposed implementation can be extended to the Comb decimation filter for multibit $\Sigma\Delta$ A/D converters.

References

[1] E. B. Hogenauer. "An Economical Class of Digital Filters for Decimation and Interpolation". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29(No. 2):155–162, April 1981.

[2] S. Chu et al. "Multirate Filter Designs Using Comb Filters". *IEEE Transactions on Circuits and Systems*, vol. CAS-31:913–924, November 1984.

[3] Y. Gao et al. "A Comparison Design of Comb Decimators for Sigma-Delta Analog-to-Digital Converters". *Analog Integrated Circuits and Signal Processing*, vol. 22(No. 1):51–60, January 2000.

[4] M. G. Bellanger et al. "Digital Filtering by Polyphase Network: Application to sample-Rate Alteration and Filter Banks". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24(No. 2):109–114, April 1976.

[5] Y. Gao et al. "Low-Power Implementation of a Fifth-Order Comb Decimation Filter for Multi-Standard Transceiver Applications". *International Conference on Signal Processing Applications & Technology, ICSPAT*, October 1999.

[6] J-M. Muller. *"Elementary functions, Algorithms and Implementation"*. Birkhauser Boston, 1997.

[7] Y. Dumonteix and H. Mehrez. "A Family of Redundant Multipliers Dedicated to Fast Computation For Signal Processing". *International Symposium on Circuits & System, ISCAS*, May 2000.