LOW–POWER DECIMATION FILTER ARCHITECTURES FOR SIGMA–DELTA ADC's

Özge Gürsoy Orkun Sağlamdemir Mustafa Aktan Selçuk Talay Günhan Dündar
e–mail: [ozge.gursoy, orkun.saglamdemir, aktanmus, talays, dundar]@boun.edu.tr
Boğaziçi University, Department of Electrical and Electronics Engineering,
34342, Bebek, Istanbul, Turkey

Key words: Sigma–Delta ADC, low–power, decimation filter.

ABSTRACT

In this work, recent low–power decimation filter architectures of Sigma–Delta analog–to–digital converters are presented and an improved version is proposed. Most of the available studies are focused on the analog part of Sigma–Delta architectures. However, the digital part can also be designed as low–power. This work presents a low–power and area design of the decimation filter section in the digital part.

I. INTRODUCTION

Sigma–Delta analog–to–digital converters (ADC) [1] are gaining more and more interest due to their various properties, which allow them to be integrated into systems within various different applications. They are suitable for many applications ranging from medium speed telecommunication to low speed, high resolution instrumentation systems. They utilize oversampling for shaping the quantization noise out of band. Thus, these architectures may provide up to 24 bits of resolution. However, in order to achieve such performance, the oversampling ratio should be high, which limits the input frequency.

Sigma–Delta ADC's may provide high resolution in small silicon area. Also, the power consumption is better compared to ADC architectures with similar performance values. Sigma–Delta ADC's contain a few amplifiers or comparators if they are single bit architectures. For other ADC's, such as flash or pipeline, the number of active elements is larger. Thus, power consumption values tend to be large. However, since Sigma–Delta ADC's utilize oversampling, high clock rates are common and this increases the power consumption.

There is much research for low–power Sigma–Delta ADC's which aim to achieve lowest power consumption especially for telecommunication applications. However, the majority of these studies are mainly focused on decreasing the power consumed in the modulator part or in other words, analog part. There are reasonable achievements in this research area and low–power SD modulators are available in the literature. On the other hand, the whole ADC architecture can be further optimized regarding the power consumption.

SD ADC's contain digital filters which perform decimation and low pass filtering. The decimation filter downsamples the output and the following low pass filter filters the out of band noise. These filters can also be optimized regarding the power consumption. There are some approaches for decreasing the power consumption.

This work presents two decimation filter architectures and proposes an improvement over them. The filters were described by VHDL and synthesized with Leonardo in 0.35μm AMS technology.

The following sections present one common and one power optimized architecture. The next section introduces some improvements for further optimization of power consumption. Then simulation results are presented and the last section concludes the work.

II. CIC DECIMATION FILTERS

The CIC filter, which is the abbreviation for the Cascaded Integrator–Comb, was proposed by

Hogenauer for decimation and interpolation [2]. In the case of Sigma–Delta Modulation, the decimation application is considered. This filter is presented in Figure 1.

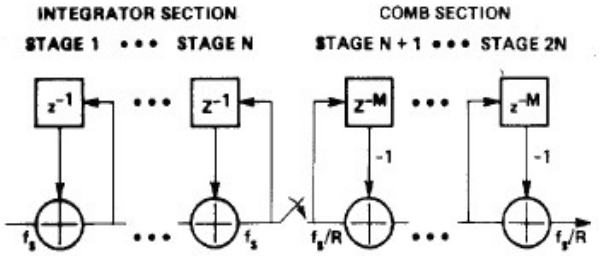

Figure 1. CIC Decimation Filter.

In the first block of CIC decimation filters, there are N–integrator stages whose transfer functions are of the form $H_I(z)=(1-z^{-1})^{-1}$. The downsampler stage follows the N^{th} integrator stage. Finally, N comb stages are cascaded. The transfer function of a single comb stage is $H_C(z)=1-z^{-M}$. M is the delay term in the comb structure. When this transfer function is referenced to the high frequency part it becomes $H_C(z)=1-z^{-RM}$ because of the downsampling ratio R, which is also the conversion ratio for converting high frequency sampled data to low frequency data. The overall transfer function of the CIC filter becomes

$$H(z) = H_I^N(z)\, H_C^N(z)=[(1-z^{-RM})/(1-z^{-1})]^N = (1+z^{-1}+\ldots+z^{RM-1})^N \quad (1)$$

As it is seen from the architecture of the filter, multipliers are not used. This is the most important advantage of the filter. On the issue of the frequency characteristics of the filter, it is verified in [2] that zeros exist at the multiples of f=1/M where f is referenced to the low frequency of f_s/R. Aliasing occurs around every multiple of f in the band limited by $\pm f_c$, where f_c is the cut–off frequency of the filter. Maximum aliasing occurs around the first aliasing band [2].

Since unity feedback is utilized in the integrator blocks, register overflow problem must be considered. For elimination of this problem, register growth is calculated and the architecture is repeated by the calculated value. The formula for this is derived in [2] as $G_{max}=(RM)^N$. According to this calculated register growth the most significant bit at the filter output, $B_{max,}$ is the smallest integer not less than the value (N.log RM + Bin — 1). This B_{max} is the most significant bit of every stage of the filter if truncation or rounding is not used in simplifying the circuit. If B_{max} becomes too large, truncation and rounding may be used in some cases, which is dealt in [2] widely. As will be demonstrated, in our case it was not appropriate to use truncation.

In the case of implementation of the $sinc^4$ filter using CIC approach, specifications for the filter were: (1) B_{in} number of input bits is 1 since the output of the Sigma–Delta modulator is 1 bit. (2) R which is the conversion ratio is 4 (3) Number of output bits 9 (4) $f_c=f/8$.

The transfer function of the $sinc^4$ is

$$H(z)=[(1/4).(1+z^{-1}+z^{-2}+z^{-3})]^4.$$

To make this equal to (1) without considering the constant ¼ :

$$(RM-1)N=(4M-1)N=12 \ .(R=4).$$

For M=1, N=4 and for M=2, N=12/7. Since N is an integer N=4 is chosen which means that both integrator and comb stages will be N=4 stages and differential delay in the comb stage will be M=1. Using $B_{in}=1$, N=4, M=1, R=4, B_{max} is 8. So the number of output bits is 9, which is in compliance with the specifications. Since power is an important issue, whether truncation can be used or not is investigated via the formula $B_{2N+1}=B_{max}-B_{out}+1$ where B_{2N+1} is the number of bits to be truncated after the N^{th} comb stage. Since $B_{max}=8$, $B_{out}=9$ B_{2N+1} is 0, which means that truncation can not be used in this case.
Having calculated the specifications of the architecture, the structures in Figure 2 were used for single integrator and comb stages.

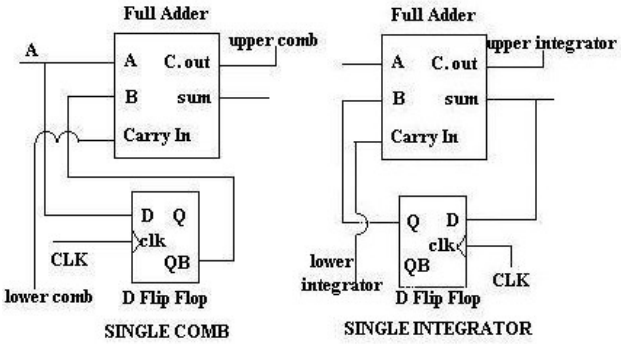

Figure 2. Structures used for realizing single integrator and comb stage.

Connecting these 1 bit structures, the overall filter was implemented as in Figure 3. Note that down samplers were realized using D flip–flops clocked at the lower frequency.

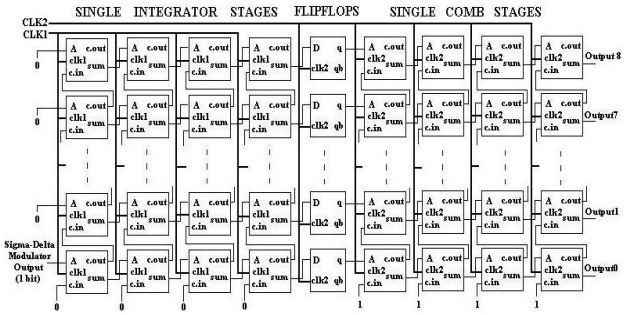

Figure 3. Overall design of the filter.

III. POWER OPTIMIZED SINC^4 FILTER

An alternative technique [3], whose implementation is given below, results to be even more power efficient than the usual approach based on comb filters. This approach can be applied to any other sinc stage, whenever a single–bit quantization is used in the Sigma–Delta modulator, resulting in a significant power reduction, with respect to a standard approach [4]. This work is focused on the power optimization of the sinc^4 filter, which is the first block of the digital decimation filter.

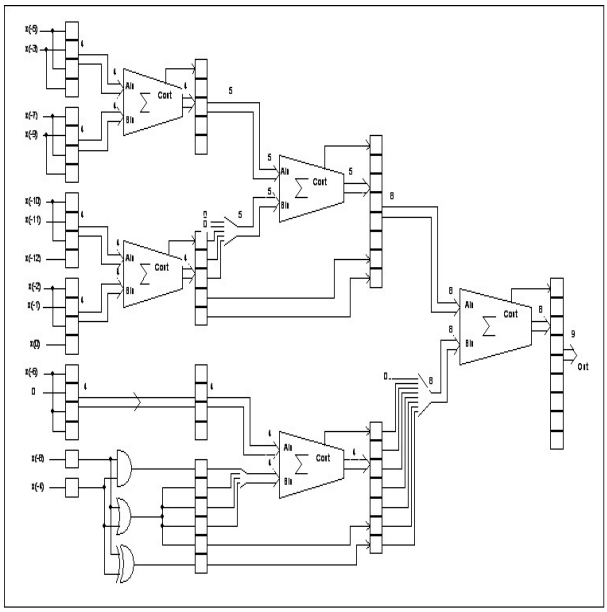
Figure 4. Filter block diagram

The transfer function of the sinc^4 filter is the following:

$$H(z) = \sum_{i=0}^{M-1} h_i . z^{-i} = \left\{ \frac{1}{N} \cdot \sum_{i=0}^{N-1} z^{-i} \right\}^4$$

where M is the filter order and N is the decimation factor.

The output sample of the filter at time t_0, y_0, as a function of the input samples x is expressed as follows: (in our case it holds N=4)

$$y_0 = x(0)+4x(-1)+10x(-2)+20x(-3)+31x(-4)+\ldots$$
$$\ldots+40x(-5)+44x(-6)+40x(-7)+31x(-8)+\ldots$$
$$\ldots+20x(-9)+10x(-10)+4x(-11)+x(-12)$$

where x(–i) represents the input value sampled i clock cycles before t_0.

This sinc^4 filter has 13 inputs (x(–12),…,x(0)) and 9 outputs (y1,…, y9). This circuit requires the last 13 samples of the input signal, which are stored in a shift register clocked at 8kHz [4]. Power consumption reduction for the proposed technique occurs due to the following reasons. This alternative approach of the sinc^4 filter requires a total of 65 1–bit memory cells, three 4–bit binary adders, one 5–bit binary adder, one 8–bit binary adder and three logic gates. A standard implementation based on the CIC architecture requires at least 7 registers and 7 adders or subtractors. Therefore this alternative architecture reduces the hardware complexity. Furthermore the integrators in the CIC approach run at the incoming rate of 8kHz, while, all blocks in this technique run at 2kHz directly [4].

IV. PERFORMANCE IMPROVEMENTS

The sinc filter in figure 4 is a pipelined implementation of the filter. Registers are inserted in between adders so as to minimize the delay and extra power consumption introduced by glitches due to long critical paths between inputs and outputs. However, these registers do consume power and occupy a big amount of area thereby reducing the gain achieved by pipelining. In our work, we have removed the pipelining registers in between the adders without touching the ones at the inputs which serve for the down conversion of the incoming data from 8kHz to 2 kHz, i.e decimation.

We have synthesized three sinc^4 filters, namely CIC approach, direct approach [4], and the direct approach without pipelining (our approach) in AMS 0.35 μm technology. The comparison of the designed filters in terms of area is given in table. The given values are the total area values of the gates after the synthesis. It is clear that the routing area should also be considered. However, since the routing area can be expressed as a fraction of the total gate area, the ratio between the given values in Table 1 does not change considerably.

Table 1. Comparison of filters in terms of area.

Method	Area (um^2)
CIC	44353
Direct w/ pipelining	27828
Direct w/o pipelining	17163

For the direct methods the 12 bit shift register working at 8 kHz which is not shown in figure 4 is also included.

V. POWER SIMULATION RESULTS

All designs were defined by VHDL and functional simulations were carried by ModelSim. After functional verification, designed filters were synthesized by Leonardo which is part of the MentorGraphics suite.

For the designed filters described in the previous section, power simulations were done using an event driven simulator with a variable delay model which can handle glitches. The supply voltage for the technology used is 3.3V. Power simulation results of the filters are given in Table 2. Inputs given to the system for simulation purposes may be categorized in two classes: DC values and Sigma–Delta outputs. The values given in the tables are the results gathered from stimulus achieved from Sigma–Delta modulator outputs.

As can be seen from the table, our approach gives the best power performance. The improvement in power of our approach over the approach in [4] is smaller compared to the improvement in area which is due to the increased glitching power caused by removing the pipeline registers.

Table 2. Power simulation results of designed filters .

Method	Power (nW)
CIC	168.6
Direct w/ pipelining	22.4
Direct w/o pipelining	18.8

VI. CONCLUSION

In this work, different decimation filters were presented. Their performances regarding area and power values were compared and some improvements were presented. Although the improvements seem to be somewhat obvious, there is no work describing such improvements. Throughout this work only decimation part was discussed. The digital filters connected to the Sigma–Delta modulators should also contain low–pass filters. However, the performances of these filters were not considered here.

The presented work should give an idea of power and area optimization for decimation filters for designers. Also, the presented work shows up to which point area and power can be optimized. The standard filter designs perform far worse than the optimized designs. This issue has great importance since nearly all power aware Sigma–Delta designs try to minimized the modulator power. Although the power in the filter is very small compared with the modulator consumption, further improvement in the digital part results with longer battery life and cheaper silicon implementation. Since, power aware designs are at the edge of technology limits, every small improvement has great significance.

Another contribution of this work is that the power simulation method used in this work estimates the power consumption better. Thus, values given for the power consumption in [2] and [4] are optimistic values. Our approach presents more realistic values. Hence, the claims presented in [4] were validated with more exact simulations and the actual improvements were observed.

REFERENCES

1. S. R. Norsworthy, R. Schreier & G. C. Temes, *Delta–Sigma Data Converters: Theory, Design, and Simulation*, IEEE Press, 1998
2. E.B. Hogenauer,"An Economical Class of Digital Filters for Decimation and Interpolation", *IEEE Transactions on Acoustic,Speech and Signal Processing*, Vol. ASSP–29, No. 2 pp. 155–162, Apr. 1981.
3. J.C. Candy, "Decimation for Sigma Delta Modulation", IEEE Transactions on Communications, Vol. COM–34, No. 1, pp.72–76, Jan. 1986
4. A. Gerosa and A. Neviani, "A Low–Power Decimation Filter For A Sigma–Delta Converter Based On A Power–Optimized Sinc Filter" ,Proceedings of the 2004 International Symposium on Circuits and Systems, Vol 2, 23–26 pp:II – 245–8, May 2004