WAVE DIGITAL DECIMATION FILTERS IN OVERSAMPLED A/D CONVERTERS

E. Dijkstra, L. Cardoletti, O. Nys,
C. Piguet and M. Degrauwe

CSEM, Centre Suisse d'Electronique et de Microtechnique S.A.
Maladiere 71, 2007 Neuchatel, Switzerland

ABSTRACT

In this paper a digital decimation filter system for oversampled A/D converters is addressed. In order to achieve very low passband ripples for data acquisition converters a cascade of a comb-filter and some bireciprocal equiripple Wave Digital Decimation filters is applied. The complexity of the filters is minimized by taking into account the noise shaping of the modulator.
A design example has been given.

I Introduction

Since the advent of VLSI technology, the demand for performant A/D converters is constantly growing. Oversampled converters are very well suited for VLSI technologies since a high resolution can be achieved without the need of highly matched components in the analog part of the converter[1]. The large digital part of the converter is a digital low-pass decimation filter whose function is to filter out the noise shaped quantization noise and to compress sample rate again to a more convenient frequency.

The digital decimation filter can be realized in several ways. In previous work[2] we discussed a design methodology for very high order FIR decimation filters in Sigma-Delta converters. Such filter realizations become however unrealistic for very high performance A/D converters. In this paper we will, therefore, focus on a cascade of a simple comb-filter and some wave digital decimation filters. The principle motivation of this cascade is to be able to realize efficiently b-bits data acquistion converters with pass-band ripples smaller than $2^{-(b+1)}$.

Section II will give an overview of the system to be considered. Thereafter, in section III we will discuss the filter transferfunctions which are needed. The complexity of the filters has been minimized by taking into account the noise shaping of the modulator. Finally, in section IV we will give a typical design example for a 16 bits data acquisition Sigma-Delta converter, followed by the conclusions.

II System overview

As stated in the introduction several filter algorithms could be used to filter out the quantization noise. In this paper we will focus on the data acquisition converter system of fig.(1). In this scheme, a first filtering and decimation is performed by a comb-decimation filter.
Their transferfunction can be expressed as:

$$\left(\sum_{i=0}^{N-1} z^{-i}\right)^n = \left(\frac{1}{N} \quad \frac{1-z^{-N}}{1-z^{-1}}\right)^n$$

where N=decimation factor of the comb filter
n=order of the comb filter (usually 2 for a first order modulator and 3 for a second order modulator)

As Candy showed[4] such filters are attractive candidates for a first decimation stage since they can be efficiently implemented. One such an implementation using modulo arithmetic has been described in [5].
The comb filter introduces a sin(x)/x attenuation in the frequency domain. This behaviour is the main reason to sample down only to about 4 times the Nyquist rate with the combfilter.

The sin(x)/x corrector in fig(1) corrects the sin(x)/x attenuation of the combfilter. Fig(2) shows the flowgraph of the corrector we used which is basically a lossy Wave Digital Filter (WDF) [6]. The performance of the corrector depends on the position in the cascade chain of filters. The best correction is obtained at the highest sample rate, thus directly after the combfilter. On the other hand, from an implementation point of view it would be better to operate this filter at the lowest possible rate. Therefore, the definite position of the corrector will depend on the application and the desired correction performance.
The coefficients of the corrector are determined by applying a Newton-Raphson iteration algorithm which minimizes the remaining passband ripple after correction. In this way remaining passband ripples can be achieved which are smaller than 10^{-5} dB.

Reprinted from *IEEE Proc. ISCAS'88*, pp. 2327–2330, June 1988.

After the sin(x)/x corrector the chain of fig(1) continues with a cascade of bireciprocal equiripple Wave Digital lattice Filters[7]. Between each WDF a decimation by a factor of two is performed. The advantages or this cascade are in this context threefold:

a). The passband behaviour and the coeffients sensibility in the passband are excellent for WDF structures. As explained before, for data acquisition converters this is of prime importance. The "poorer" stopband properties are less important because we do not need very severe filter characteristics due to the first prefiltering by the comb filter.

b). For bireciprocal WDF's complexity is reduced by the possibility of an interchangeability between sample rate compression and filtering function. Concretely, this means that decimation (by a factor of two) can be performed before filtering and therefore each WDF will operate at only half of its sample frequency. Furthermore, bireciprocal WDF strucures do use only (N-1)/2 multiplications for a filter order N[7].

c). Bireciprocal WDF's are inherently scaled in an optimum way for sinusoidal signals[7].

In the next section we will discuss the filter characteristics of the WDF's which are needed and the different design trade-off's which have to be made.

III The required filter complexity for the WDF's

The main system parameters of interest are the imposed passband ripple and the tolerated loss in SNR due to the non-ideal filtering. With these parameters one should try to fix the filter specifications leading to a minimum overall filter complexity. In this section we will review successively the loss in SNR, the WDF filter specifications, the finite wordlength effects and some architectural issues.

III.1 Loss of SNR due to non-ideal filtering

Following the same reasoning as in [2], the converter will loose due to non-ideal filtering

$$\text{SNR-loss} = 10 \log [1+P2/P1]$$

where P1=power of the quantization noise in the baseband
P2=power of the quantization noise after filtering outside the baseband

Assuming that the quantization noise injected into the comparator of the modulator was white, Candy[4] derived formula's for the noise spectral density after the combfilter. Knowing this spectal density we can calculate for different filter parameters the resulting loss in SNR.

Obviously, as discussed in [2,3] the loss in SNR should be compensated by an "overdimensioning" of the oversampling rate pushing more quantization noise outside the baseband. Similar performance trade-off's between the analog loop and the complexity of the decimation filter as proposed for FIR filters [3] are valid for the in this paper presented system.

III.2 The filter specifications

Starting from the tolerated SNR loss and the passband ripple to be achieved we will determine in this subsection the optimum filter specifications leading to a minimum complexity. We will demonstrate this for a cascade of two bireciprocal WDF's but the results can be easily generalized. Fig.(3). defines the parameters which should be fixed.

Due to the noise shaping most of the noise is rejected into higher frequencies. This means that in order to conserve a reasonnable SNR we should require

$$as1 > as2 \quad [dB] \qquad (1)$$

For bireciprocal WDF this inherently means:

$$ap1 < ap2 \quad [dB] \qquad (2)$$

Furthermore in order to satisfy the passband ripple we require:

$$ap1 + ap2 = ap \quad [dB] \qquad (3)$$

The first WDF should normally avoid the aliasing of the [Fcl/2-Fmax, Fcl/2] band into the baseband, but due to the fact that only bireciprocal filters are considered, the passband edge continues until Fcl/8 and therefore the second filter can not filter out any noise in the [Fmax, Fcl/8] range. Hence, our Fsl choice is fixed as:

$$Fs1 = 3/4 * Fcl/2 \qquad (4)$$

Due to the bireciprocal WDF option the parameters as1 and as2 are related to respectively ap1 and ap2. This together with (3) and (4) implies that Fs2 and ap2 remain the only real degrees of freedom.

The algorithm of fig.(4) scans the different possible values of Fs2 and ap2. Each time, the orders n1 and n2 of both WDF's and the resulting SNR loss are outputted. For a given maximum loss of SNR we can sort out the satisfying couples (n1,n2). It turns out that for a large majority of applications the couple which minimizes the chip complexity can easily be found by minimizing (n1+n2).

III.3. Finite wordlength effects

The finite wordlength of the coefficients affect obviously the transferfunctions of the sin(x)/x corrector and the WDF's. Due to the excellent sensitivity properties of

WDF's the number of bits on which the coefficients should be coded can be kept surprisingly low while still satisfying the transferfunction.
The inevitable wordlength truncation after each multiplication introduces extra noise sources, which degrade again the SNR.

III.4. Architectural issues

The comb filter in the chain of fig (1) can be efficiently implemented by using the algorithmic decomposition proposed in [5], whereas the sin(x)/x corrector and the WDF's can efficiently be implemented by the microprogram controlled datapath of fig(5). This datapath essentially consists of a programmable barrel shifter cascaded with an addition unit. Compared to e.g.[8] the architecture has been slightly changed (i.e. two slave-accumulators + some multiplexers) in order to be able to handle more efficiently the rather complicated flowgraph of WDF's.

IV A design example

In this design example we will consider the design of a digital decimation filter for a 16 bits Sigma-Delta A/D converter. Fig.(6) resumes the main characteristics which have been imposed.

In this example we will first sample down the oversampling frequency by the combfilter with a factor 78. This means that the output of the combfilter operates at 32 kHz. Since the imposed pass-band ripple is severe, the sin(x)/x corrector should be placed directly after the combfilter thus operating also at 32 kHz. In this case all coefficients of the sin(x)/x corrector can be coded on 11 bits. The resulting passband ripple after correction is 3.7E-6 dB.

After this correction a cascade of two WDF's has been dimensionned. Fig.(7) gives the best solution for the tolerated SNR loss. Thus, by taking N1=5 and N2=9 (orders should always be impair) the filter characteristics can be realized. Figs. (8) and (9) give a graphical representation[9] of the imposed filter specifications and the transferfunctions for coefficients coded on 8 bits.

V Conclusions

The proposed cascade of filters allows to design efficiently decimation filters for oversampled Sigma-Delta data acquisition converters. The needed filter specifications are derived from the noise spectral density function at the output of the modulator in view of minimizing the complexity of the filters.

References:

[1] M.W. Hauser, P. Hurst, R. Brodersen: "MOS ADC filter combination that does not require precision analog components". Proc. of ISSCC 85, pp 80-81.

[2] E. Dijkstra, M. Degrauwe, J. Rijmenants, O. Nys: "A design methodology for decimation filters in Sigma Delta A/D converters". ISCAS 87 pp.479-482.

[3] O. Nys, M. Degrauwe, E.Dijkstra. "A CAD tool for oversampled CMOS A/D converters", Proc. of 1987 Symp. on VLSI circuits, May 1987, Karuizawa, pp 113-114

[4] J.C. Candy "Decimation for Sigma-Delta modulation" IEEE Trans. on comm. Vol COM-34, No.1, Jan 1986, pp 72-76.

[5] E. Dijkstra, O. Nys, C. Piguet, M. Degrauwe "On the use of modulo arithmetic combfilters in Sigma Delta converters" to be published at ICASSP 88, New York.

[6] G. Lucioni, "Alternative method to magnitude truncation in WDF", IEEE Trans. on circuits and systems, Vol CAS-33

[7] L. Gazsi, "Explicit formulas for lattice Wave Digital Filters", IEEE Trans. on circuits and systems, Vol CAS-32, No. 1, Jan 1985, pp. 68-88.

[8] A. Rainer, "Adder based digital signal processor architecture for 80 ns cycle time". Proc. ICASSP 84, pp.16-9.1-4, San Diego, March 1984

[9] L. Gazsi, "Reference manual Falcon", Ruhr University, Bochum, West-Germany.

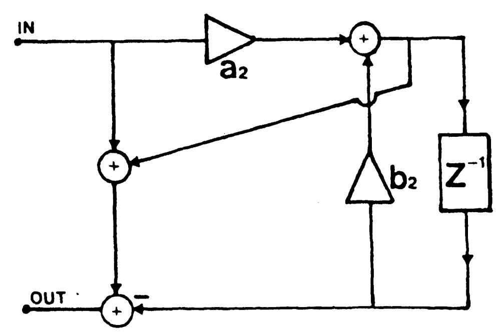


fig.2 Flow-graph of the sinX/X corrector.

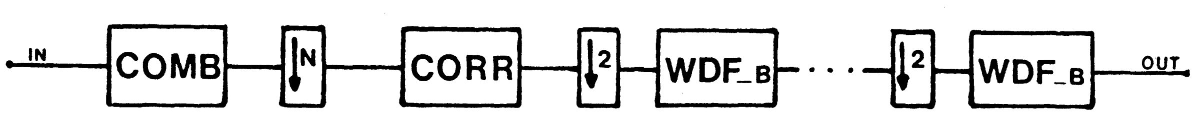
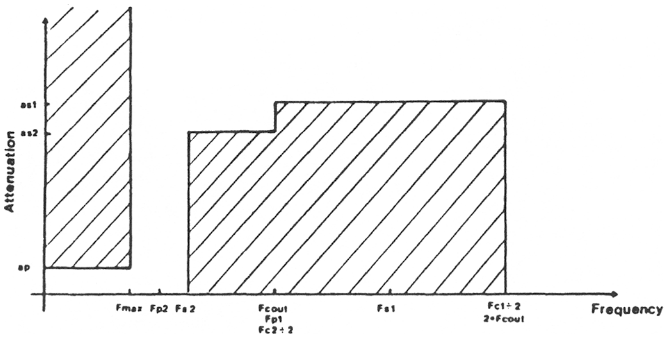


fig.1 Decimation filter structure to be considered.



Fmax = maximum signal input frequency.
Fcout = output rate of the converter.
Fci = sample frequency of i-th WDF.
Fsi = stopband-edge of i-th WDF.
Fpi = pass band-edge of i-th WDF.
This corresponds for bireciprocal WDF's to Fci/4.
The attenuation at this frequency is 3 dB.
api = passband ripple of i-th WDF in dB.
ap = sum of api.
asi = stopband attenuation of i-th WDF in dB.

fig.3 parameter definition for the filterspecifications.

User's specifications :

```
     Fmax=?            Fcout=?
     SNR_loss=?        ap=?
```

Algorithm :

```
Fp2:=Fcout/2; Fp1:=Fcout;
Fs1:=3/2*Fcout; Fs2:=Fp2+epsilon;
while Fs2<(Fcout-Fmax) do
      begin
        ap2:=ap/2;
        while ap2<ap do
              begin
                ap1:=ap-ap2;
                as1:=fct(ap1);
                as2:=fct(ap2);
                order_n1:=fct(ap1,as1,Fs1);
                order_n1:=fct(ap2,as2,Fs2);
                SNR_loss:=fct(ap1,ap2,as1,as2,Fs1,Fs2);
                ap2:=ap2+epsilon2;
              end;
        Fs2:=Fs2+epsilon1;
      end;
```

fig.4
Algorithm determining the possible couples (n1,n2).

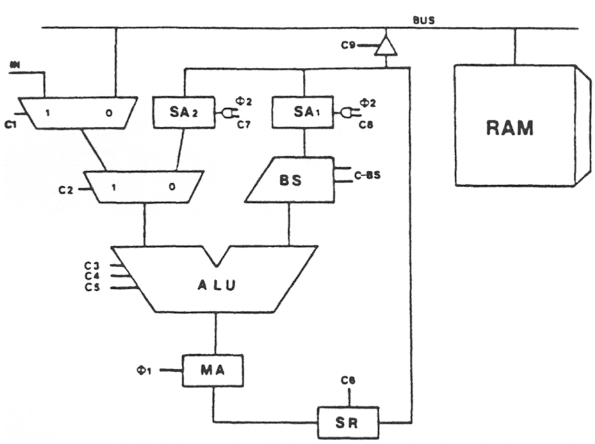


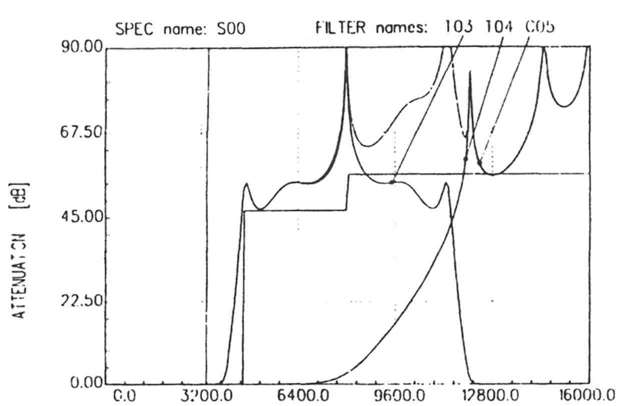
Fig. 5. Architecture of the datapath

nbr_of significant bits	16	bits
passband ripple	10E-4	dB
tolerated SNR-loss	5	dB
max. freq. of input signal	3.4	kHz
output rate	8	kHz
order of analog loop	2	
oversamping frequency calculated with program described in [3]	2496	kHz

fig 6. characteristics of the 16 bits data acquisition A/D converter

Fs1	12	kHz
Fs2	4.6	kHz
ap1	3.4E-5	dB
ap2	6.6E-5	dB
as1	51	dB
as2	48	dB
n1	4.18	
n2	7.15	
SNR_loss	4.6	dB

fig. 7. optimum filter specifications leading to a minimum complexity

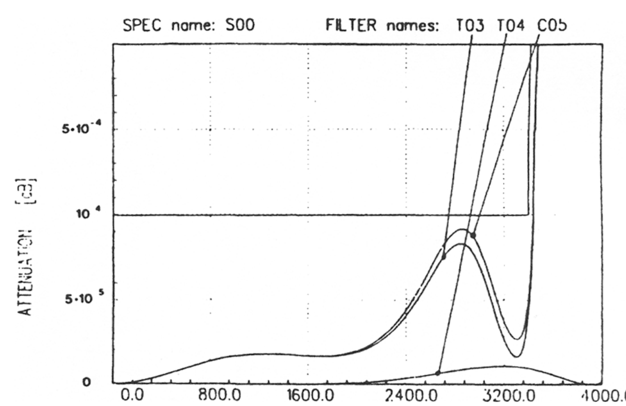


T04 - Transfertfunction of first WDF with coefficients coded on 8 bits.

T03 - Transfertfunction of second WDF with coefficients coded on 8 bits.

C05 - Cascade of T03 and T04.

fig.8 Transfertfunction of the WDF's cascade for a 16 bits A/D converter.



T04 - Passband behaviour of first WDF with coefficients coded on 8 bits.

T03 - Passband behaviour of second WDF with coefficients coded on 8 bits.

C05 - Cascade of T03 and T04.

fig.9 Passband behaviour of the WDF's cascade for a 16 bits A/D converter.

A DESIGN METHODOLOGY FOR DECIMATION FILTERS IN SIGMA DELTA A/D CONVERTERS

E. Dijkstra, M. Degrauwe, J. Rijmenants, O. Nys

CENTRE SUISSE D'ELECTRONIQUE ET DE MICROTECHNIQUE S.A.
Maladière 71, 2007 Neuchâtel, Switzerland

ABSTRACT

A design methodology is presented for FIR decimation filters in sigma delta A/D converters. The realized filter transfer function takes into account the noise shaping in order to minimize filter complexity. A simulated annealing optimization is performed to minimize the wordlength of the filter coefficients. This reduces typically the wordlength with two bits compared to classical approaches.

The methodology has been integrated in a CAD-tool for the design of high performance analog circuits. This permits to make rapidly trade-off's between the performance requirements of the analog loop and the complexity of the decimation filter. A design example has been given.

I. INTRODUCTION

Decimation is the process of sample rate reduction in digital filters. Usually this is done by FIR filters because of the interchangeability between sample rate compression and filtering functions [1] A typical and popular application is that of a FIR decimation filter for oversampled A/D converters. In this case, due to the high performance requirements (narrow base-band and transition range, high stop-band attenuation), high filter orders are required. Often, this is still realizable in an efficient way, since the multiplications of the innerproducts are reduced to simple additions/substractions. Fig. 1 shows the basic architecture of this solution [2]. The ROM which contains half of the coefficients of a symmetric FIR filter is scanned with an up-down counter, whereas the convolution sum between coefficient and data is accumulated.

The accumulator has L-coef full adder (FA) slices. On the ROM-side, L+1 bits has to be connected to the logical "0". The NXOR function between the sign-bit of the coefficients and the output of the analog loop determines whether an addition or a substraction is performed.

Time multiplexing on M adders can be used in order to handle oversampling rates of 1/M times the filter order N. In this case, either the ROM should be enlarged with new tables (fig. 1) or a shuffle network should be applied [2]. If a new table is added to the ROM, the new up-down counter should start operation N/M clock cycles later than its predecessor.

Other interesting realizations which are recently published are based on dump- and accumulate filters [3] or using an IIR configuration [4]. As such approaches are only appropriate for decimation down to 4 times the Nyquist rate, a second decimation stage should be used to arrive at the Nyquist rate.

In this paper we will, therefore, only focus on the architecture of fig. 1. First, we will discuss the filter transfer function which is needed. The complexity of the filter is minimized by taking into account the noise shaping. Furthermore, the most important trade-off's between the performance requirements of the analog loop and the complexity of the decimation filter are discussed.

In sections III and IV, a simulated annealing optimization is presented for the minimization of the wordlength of the filter coefficients. Finally, the software environment into which the methodology has been integrated is discussed and a design example is given.

II. THE REQUIRED FILTER COMPLEXITY

The SNR of the A/D converter can be calculated if the spectral density of the quantization noise at the output of the Σ-δ modulator is known. This spectral density can be easily derived by recognizing that the sum of the signal and the noise power is equal to V_{DD}^2. As the power of the input signal is

$$V_{DD}^2\left(\frac{V_{in}}{V_{in,max}}\right)^2,$$

where $V_{in,max}^2$ is the full scale input signal, the power of the quantization noise is given by:

$$P_N = V_{DD}^2\left[1-\left(\frac{V_{in}}{V_{in,max}}\right)^2\right] \qquad (1)$$

Assuming that the quantization noise injected into the comparator is white, it can be deduced that the noise spectral density at the output of the modulator is given by:

$$S_{nn}(f) = V_{DD}^2\left[1-\left(\frac{V_{in}}{V_{in,max}}\right)^2\right]\cdot\frac{|T^2(f)|}{\int_0^{f_H/2}|T^2(f)|\,df} \qquad (2)$$

Reprinted from *IEEE Proc. ISCAS'87*, pp. 479–482, May 1987.

where: T (f) = noise transfer function

f_H = sample frequency.

The decimation filter function can be splitted up into three parts (fig. 2) and the noise contributions of each part can be calculated.

In the base-band, the power of the noise is given by:

$$P1 = V_{DD}^2 \left[1-\left(\frac{V_{in}}{V_{in,max}}\right)^2\right] \cdot \frac{\int_0^{f_s} \left|T^2(f)\right| df}{\int_0^{f_H/2} \left|T^2(f)\right| df} \qquad (3)$$

In the transition band, it is given by:

$$P_2 = V_{DD}^2 \left[1-\left(\frac{V_{in}}{V_{in,max}}\right)^2\right] \cdot \frac{\int_{f_s}^{f_s+\delta f} \left|T^2(f)\right| \cdot \left|1-\frac{(f-f_s)}{\delta f}\right|^2}{\int_0^{f_H/2} \left|T^2(f)\right| df} \qquad (4)$$

Finally, for the stop-band, the noise power is given by:

$$P_3 = V_{DD}^2 \left[1-\left(\frac{V_{in}}{V_{in,max}}\right)^2\right] \cdot \frac{\int_{f_s+\delta f}^{f_H/2} \delta s^2 \, df}{\int_0^{f_H/2} \left|T^2(f)\right| df} \leqslant \qquad (5)$$

$$\leqslant \frac{V_{DD}^2 \left[1-\left(\frac{V_{in}}{V_{in,max}}\right)^2\right] \delta s^2 \cdot \frac{f_H}{2}}{\int_0^{f_H/2} \left|T^2(f)\right| df}$$

By supposing a sinusoïdal input signal with amplitude A, the SNR is given by

$$SNR = 10.\text{Log} \frac{\frac{1}{2}\left(\frac{A}{V_{in,max}}\right)^2 V_{DD}^2}{P_1 + P_2 + P_3} \qquad (6)$$

For an ideal low pass filter, the SNR would be:

$$SNR = 10.\ \text{Log} \frac{\frac{1}{2}\left(\frac{A}{V_{in,max}}\right)^2 V_{DD}^2}{P_1} \qquad (7)$$

Thus, due to the non-ideal filtering of the noise, the converter will loose:

$$SNR = 10.\ \text{Log}\left[1 + \frac{P_2}{P_1} + \frac{P_3}{P_1}\right] \qquad (8)$$

of its signal to noise ratio.

On the other hand, for full scale input signals ($A = V_{in,max}$), a n-bits converter requires a SNR of:

$$SNR > 20.\ \text{Log}\ 2^{n+1} \qquad (9)$$

Thus, in order to satisfy (6) and (9), the following equation should hold:

$$\frac{P_1 + P_2 + P_3}{V_{DD}^2} \leqslant \left(\frac{1}{2}\right)^{2n+3} \qquad (10)$$

Likewise, an ideal decimation filter requires that:

$$\frac{P_1}{V_{DD}^2} < \left(\frac{1}{2}\right)^{2n+3} \qquad (11)$$

where P_1/V_{DD}^2 is proportional to the third or fifth power of f_s/f_H for respectively a first and second order sigma-delta converter.

Equation (10) shows that a trade-off between the performance requirements of the analog part (oversampling rate) and the digital part should be made. Either P_1 is chosen much smaller than (11) and the filter performances are relaxed or it is chosen close to (11) and the filter performances are enhanced.

From (8), it is clear that if a X dB loss of SNR in the decimation filter is accepted, then the following equations should be satisfied:

$$\frac{\int_{f_s}^{f_s+\delta f} \left|T^2(f)\right| \cdot \left|1-\frac{f-f_s}{f}\right|^2 df + \delta_s^2 \cdot f_H/2}{\int_0^{f_s} \left|T^2(f)\right| df} = 10^{X/10}-1 \qquad (12)$$

and

$$\frac{P_1}{V_{DD}^2} = \frac{10^{-X/10}}{2^{2n+3}} \qquad (13)$$

Thus, we should calculate δf and δs in such a way that a minimal filter order Nmin is obtained for a given loss of X dB SNR. As the order of a FIR can be approximated as [5] :

$$N = \frac{-10\ \log(\delta p * \delta s) - 15}{14.\ \delta f}\ fh + 1 \qquad (14)$$

we should minimize N under the constraint of (12). However, as N can only take multiple values of f_h/f_n (f_n = output rate of the digital filter), the minimizing of the filter's complexity can still be a little refined. The margin between M.fh/fn and Nmin can be exploited by taking δf as small as possible. Consequently, for the same SNR loss, the stop-band attenuation requirement δs can be relaxed and can, therefore, reduce the wordlength of the coefficients [8].

III. FINITE WORDLENGTH OF THE COEFFICIENTS

Obviously, while satisfying the specifications, the coefficients of the filter should be calculated with a minimal finite wordlength. Traditionally this problem is tackled by applying the Remez-algorithm [6,7] for the design of an infinite precision solution. A b-bit finite wordlength solution is then found by rounding or truncation of the coefficients to b-bit.

In [8] it has been pointed out that a substantial (5-7 dB) improvement in the stop-band can be obtained if the discrete constraint on the coefficients is included in the optimization process. Mixed Integer Programming can, at the expense of much CPU time and for low filter orders be used for this optimization [8].

As our filters will have a very high order (200--4000 taps), we can rule out this optimization method. Therefore, we applied a simulated annealing optimization algorithm [9].

IV. SIMULATED ANNEALING OPTIMIZATION OF THE WORDLENGTH

The optimization problem can be stated as follows:

> Given a FIR filter of order N, find coefficients which can all be coded with a minimum number of L-coef bits and a maximum number of L-leading zero bits (see fig. 1, definition representation of coefficients) and which respect the imposed transfer function H(jw).

An initial solution can be found by synthesizing the infinite precision filter combined with coefficients rounding. Such a solution can be optimized with simulated annealing. The basic ideas behind this optimization are:

1) Choose at random a coefficient h(k).
2) Choose at random a small coefficient change $\delta h(k)$.
3) Establish a cost (=energy) function. This function should be a function of l(h(k)), L-coef(h(k)) and the difference between the target transfer function and the actual transfer function.
4) The cost-function can easily be updated in an incremental way for small changes of h(k). It is easy to derive that:

$$\frac{\delta H(jw)}{\delta h(k)} = \begin{cases} 1 & k = 0 \\ 2\cos(2\pi k) & k <> 0 \end{cases}$$

5) In order to avoid a prohibitive CPU-time, an adaptive number of iterations per temperature step and an adaptive temperature scheme has been implemented as in [10]. The number of iterations per temperature step is determined by building up a Markov chain. After reaching a steady state behaviour, the annealing is continued at a lower temperature. This new temperature is also chosen as a function of the length of the Markov chain. In this way, the optimization takes a CPU-time which is in the same order of magnitude as the synthesis of the infinite coefficients, i.e. a few minutes on a VAX 8600 computer.

V. SOFTWARE ENVIRONMENT

The synthesis of decimation filters and its optimization for sigma-delta converters have been incorporated in a CAD-tool for the design of high-performance analog circuits [11]. As this tool is, amongst others, also able to synthetize the analog loop of the A/D converter (1st and 2nd order), a trade-off between the required performances of both parts of the converter can easily be made. This performance trade-off depends strongly on the application and the available VLSI technology. Therefore, such a tool is absolutely necessary for a fast "customization" of a state of the art A/D converter.

VI. DESIGN EXAMPLE

We will illustrate the design of the sigma-delta decimation filter and its optimization by means of a very simple example. Consider as example the design of a 8-bit first order sigma-delta A/D converter. Following the methodology proposed above, we find:

SNR loss in filter	= 6 dB	(imposed)
δp	= 11 dB	(")
f_n	= 8 kHz	(")
f_s	= 4 kHz	(")
f_H	= 800 kHz	
M	= 4 adders	
N	= 400	
δ_f	= 5,09 kHz	
δ_s	= 62,7 dB	

An initial synthesis of the coefficients in infinite precision with rounding of the coefficients [7] reveals that we should take L-coef = 18, L = 6 in order to satisfy the requirements. Thus a 12 bits wordlength of the coefficients should be implemented in the ROM-memory. A simulated annealing optimization reduced L-coef to 17 bits and increased l to 7 bits. Thus after optimization only wordlengths of 10 bits are necessary for the ROM-implementation. Fig. 3 shows the transfer function before and after the simulated annealing optimization.

VII. CONCLUSION

The design of decimation filters for sigma-delta converters can efficiently be realized by a high order FIR filter. A simulated annealing optimization has been proposed to reduce the wordlength of the coefficients with several bits.

The synthesis and optimization of the decimation filter has, together with the synthesizing of the analog loop, been integrated in a CAD-tool, enabling us to make rapidly suitable trade-off's between the required performances of both parts. Some further examples of this trade-off are given [12].

REFERENCES

[1] R.E. Crochiere, L.R. Rabiner:"Interpolation and decimation of Digital signals-A tutorial review", Proc. IEEE, Vol. 69, No 3, March 1981.

[2] M.W. Hauser, P.J. Hurst, R.W. Brodersen:"MOS ADC Filter combination that does not require precision analog components, ISSCC 1985.

[3] J.C. Candy:"Decimation for Sigma-Delta Modulation", IEEE Trans. on Comm., Vol. COM-34, No 1, January 1986.

[4] A. Huber et al:"FIR lowpass filter for signal decimation with 15 MHz clock-frequency", Proc. ICASSP 86, Tokyo, pp. 1533-1537.

[5] L.R. Rabiner et al:"Some comparisons between FIR and IIR digital filters",The Bell System Techn. Journal, Vol. 53, pp. 305-331, Feb.74.

[6] J.H. McClellan:"A computer program for designing optimum FIR linear phase digital filters", IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, No 6, Dec. 1973, pp. 506-525.

[7] U. Heute:"A subroutine for finite wordlength FIR filter design", published in IEEE Press, Programs for digital signal processing.

[8] V.B. Lawrence, A.C. Salazar:"Finite precision design of linear phase FIR filters", The Bell System Techn. Journal, Vol. 59, No 9, Nov. 80.

[9] S. Kirkpatrick:"Optimization by simulated annealing, Science 220 (1983), pp. 671-680.

[10] F. Catthoor:"Characterization of Finite wordlength effects and optimization of coefficients by Simulated annealing", Workshop CAD for DSP, Sept. 9-12, IMEC, Leuven, Belgium.

[11] M. Degrauwe et al:"An analog design expert system", ISSCC 1987.

[12] O. Nys, M. Degrauwe, E. Dijkstra:"A CAD tool for oversampled CMOS A/D converters", submitted for publication at 1987 Symp. on VLSI Technology, May 18-21, Nagano, Japan.

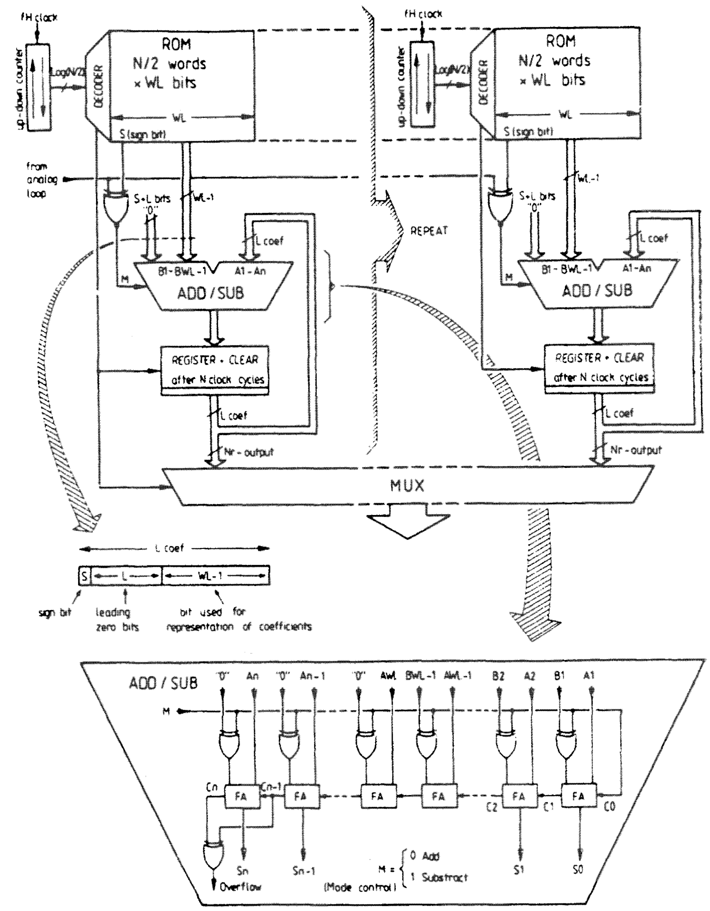


Fig. 1 Architecture of the decimation filter.

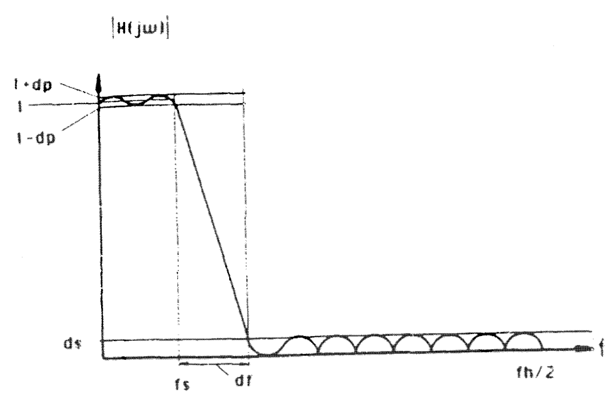


Fig. 2 Filter Characteristic.

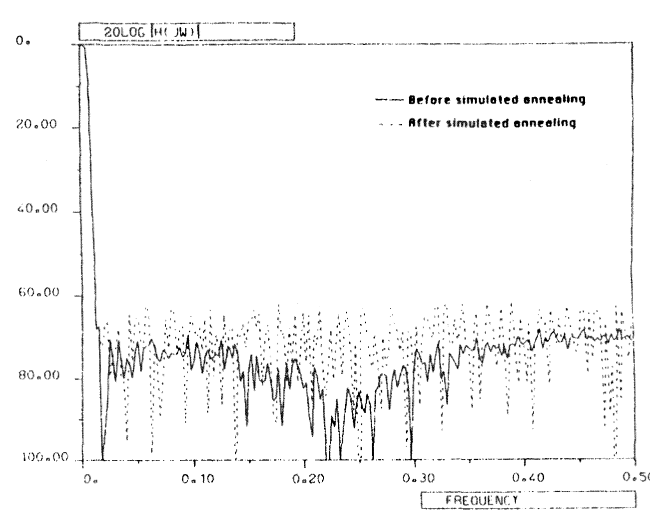


Fig. 3 Transfer functions before and after simulated annealing optimization.

ON THE USE OF MODULO ARITHMETIC COMB FILTERS IN SIGMA DELTA MODULATORS

E. Dijkstra, O. Nys, C. Piguet and M. Degrauwe

CENTRE SUISSE D'ELECTRONIQUE ET DE MICROTECHNIQUE S.A.
Maladiere 71, 2007 Neuchatel, Switzerland

ABSTRACT

A novel architecture of one stage Comb decimation filters for Sigma Delta Modulators is described. It performs the decimation of a 1 bit oversampled modulator output to an arbitrary lower output frequency.
The use of modulo arithmetic throughout the filter together with the proposed algorithmic decomposition allows a low power and area efficient implementation. This also avoids the storing of the coefficients in a ROM or the generation of the coefficients with rather complicated up/down counters.
The architecture is applicable to all comb decimation filters with $\mathrm{sinc}^k(f)$ response. A filter with k=3 and a programmable decimation factor has been integrated in a 3 μm CMOS process.

1. INTRODUCTION

With the advent of VLSI technology, oversampled Sigma-Delta modulators are becoming more and more popular for A/D conversion. The reason is that a large resolution can be obtained without highly matched components[1]. The converter consists of two parts: a relative small analog modulator and a large digital decimation filter. The oversampled analog modulator shapes the quantization noise in such a way that most of it is pushed into the higher frequencies. There this noise can be filtered out by a digital low pass filter. A frequency decimation is simultaneously performed in order to sample down the oversampled frequency to a more convenient one.

In previous work we presented a design methodology for of very long equiripple FIR decimation filters[2]. Furthermore, in [3] the front-end of a silicon compiler for sigma-delta converters has been discussed which is able to determine the system parameters of the converter (clock-frequency, filter characteristics, voltage reference specifications etc.) starting from the system-specifications (nb of bits, gain error, offset error etc).

In this paper, we will focus on a non-convential architecture for comb decimation filters. As Candy[4] showed, such filters are attractive candidates for replacing the very long equiripple FIR decimation filters[1,2].
Their transfer function can be expressed as:

$$\left(\sum_{i=0}^{N-1} z^{-i}\right)^k = \left(\frac{1}{N} \quad \frac{1 - z^{-N}}{1 - z^{-1}}\right)^k \qquad (1)$$

where: N=decimation factor
k=order of the comb filter (usually 2 for a first order modulator and 3 for a second order modulator)

In the time domain the behaviour of the decimation filter is to output after each N input samples a sample which is a weighted average of the last (N-1)*k+1 input samples. The frequency response of the filter is:

$$H(jw) = (\sin(\omega NT)/(N*\sin(\omega T)))^k \qquad (2)$$

Due to this sinc(f) attenuation we can with this filter only sample down to about 4 times the Nyquist rate[4]. Then a second decimation stage should be used to arrive at the Nyquist rate. In [5] we describe Wave Digital Decimation filters which are suitable for this task.

2. ARCHITECTURAL CONSIDERATIONS

Equation (1) can be realized in several ways.
An obvious attempt is to store the N*k filter coefficients in a ROM and to perform directly the convolution equation. However, because of the fact that one output sample should be delivered after N input samples, time multiplexing on k accumulators is necessary. The architectue will strongly resemble the one described in [1,2].
A more subtle variant is described in [6] for the special case k=3. The coefficients are not stored in a ROM but generated by rather complicated up/down counter combinations. Time-multplexing on k=3 accumulators is of course still required.

Reprinted from *IEEE Proc. ICASSP'88*, pp. 2001-2004, April 1988.

Fig(1) shows an architecture presented in [7] for k=3. The transfer function has been splitted up into a FIR part realizing $(1-z^{-N})^k$ and an IIR part realizing $(z^{-1}/(1-z^{-1}))^k$. The disadvantage of this architecture is that one should implement N*k shift registers for the FIR part. Despite the fact that the input is often coded on one bit only, this necessitates a substantial area (N is usually >100). Moreover, as the decimation can only be done after the IIR part, the whole filter operates at the oversampling rate.

The attractive alternative we focussed on is depicted in fig(2). In this case we interchanged the IIR and FIR parts. This allows a decimation in between the IIR and FIR parts and therefore z^{-N} can be replaced by z^{-1}. As a consequence we will use significantly less memory and less power consumption than the previous decomposition.
The recursive stage of the filter structure in fig(2) has a pole at z=1 which is not asymptotically stable and therefore may overflow. It can, however, be proven[8] that overflow can be avoided by using a modulo arithmetic system (or a "wrap around") everywhere in the system.
A sufficient condition for the filter to work correctly is to choose a modulo which is larger than $(N^k+1)*D$, where D is the dynamic range of the input signal. For Sigma-Delta converters the dynamic range D is usually only one bit. Due to this limited dynamic inputrange modulo arithmetic comb filters are very suitable for this kind of VLSI applications. Note that by taking a modulo $2^b \geq (N^k+1)$ all calculations can be performed by ordinary two complement's operators. Carry handling can simply be ignored.

The crucial difference between the decomposition of figs. (1) and (2) is the different need for memory resources in both FIR parts. The FIR part of fig. (1) will need N*k bit locations whereas the structure of fig(2) needs $k^2*\log_2(N)$ bit locations for its FIR part. This means that for k=3 and N>16 the latter will take less memory locations. Besides, in fig(1) all bits should be organized in a large N*k shiftregister, whereas in fig(2) bit locations could be advantageously organized in a small k words of $k*\log_2(N)$ bits RAM.
This difference in memory requirements compensates largely the slightly more complex IIR part of fig(2). (i.e. in fig(2) all integrators use everywhere b-bits, whereas in fig(1) the first and second integrator can eventually be implemented with a few bits less.) The b-bits additions in the FIR part of fig.(2) are performed at the lower frequency and therefore timemultiplexing could be done on serial adders, making the FIR adder complexity comparable to the FIR adder complexity of fig. (1).

3. VLSI IMPLEMENTATION

The structure of fig(2) has been integrated in a 3 µm CMOS technology. For this integration we focussed on a minimum complexity for the control unit and a maximum layout regularity. Moreover, by making the decimation factor programmable, the inherent trade-off between conversion speed and resolution has been exploited. In this way the same chip can be used for several applications.

The integrators of the IIR part are directly "mapped into silicon", i.e. each integrator has his own adder and no multiplexing is performed due to a limited system clock. Because of the significantly lower speed of the FIR part such a direct mapping of this part would spoil much area. Therefore we implemented the calculation of $(1-z^{-1})^3 = 1-3z^{-1}+3z^{-2}-z^{-3}$ with the distributed arithmetic architecture of fig(3). The state variables are written horizontally word by word and read vertically bitslice by bitslice. In order to avoid shifting of state variables a "wrap-around" pointer mechanism provides the new memory location where the data from the IIR part can be stored.
Obviously by using a "wrap-around" pointer mechanism for the state variables we should indicate to the ROM in which order the state variables are classified. This means that 2 extra address lines are needed.
As the ROM output wordlength can be coded on 4 bits, the ROM will contain only $(2^6)*4=256$ bits. With such a small ROM we can even suppress the adder by performing the add functions also in the ROM (fig(4)). The resulting ROM will still be acceptable small (2048 bits) and the layout regularity will be significantly improved.

Fig (5) shows the chip photo of the circuit. The total number of transistors is 7800 on an area of 3.5 mm^2 (3 µm CMOS). The oversampling frequency was 40kHz and the decimation factor N programmable. Power consumption is 5 µA at 1.5 Volts. First silicon was working.

4. CONCLUSIONS

The proposed decomposition together with the use of modulo arithmetic througoutr the filter leads to a very compact, flexible and extensible architecture of Comb decimation filters for Σ-δ converters. It is also very suitable for comb filters in the recently proposed multi-bits "interleaved" modulators[9].
Furthermore, if speed becomes a bottleneck for the structure the throughput can be elegantly enhanced by applying Residue Number System (RNS) arithmetic [10]. In this case, the filter structure (fig(2)) with one large modulo $(m>N^k+1)$ could be splitted up in several units with smaller modulo's $(\Pi m_i >= N^k+1)$. The RNS encoding is for our applications obviously not necessary (the input is coded on 1 bit),

whereas the RNS decoding can be performed after decimation and thus at a significantly lower rate.

References:

[1] M.W. Hauser, P. Hurst, R. Brodersen: "MOS ADC filter combination that does not require precision analog components" Proc. of ISSCC 85, pp 80-81.

[2] E. Dijkstra, M. Degrauwe, J. Rijmenants, O. Nys. "A design methodology for decimation filters in Sigma Delta A/D converters". ISCAS 87 pp.479-482.

[3] O. Nys, M. Degrauwe, E.Dijkstra. "A CAD tool for oversampled CMOS A/D converters" Proc. of 1987 Symp. on VLSI circuits, May 1987, Karuizawa, pp 113-114

[4] J.C. Candy "Decimation for Sigma-Delta modulation" IEEE Trans. on comm. Vol COM-34, No.1, Jan 1986, pp 72-76.

[5] E. Dijkstra et al. "Wave Digital Decimation filters in oversampled A/D converters", subm. for publ. at ISCAS 88

[6] H. Meleis et al. "A novel architecture design for VLSI implementation of an FIR decimation filter" Proc. ICASSP 85, Tampa, pp. 1380-1383

[7] A. Huber et al. "FIR lowpass filter for signal decimation with 15 Mhz clock frequency.", Proc. ICASSP 86, Tokyo, pp 1533-1537.

[8] S. Chu, C. Sidney Burrus "Multirate filter designs using comb filters", IEEE Trans. on CAS, Vol CAS-31, No. 11, Nov 84, pp.913-924.

[9] Y. Matsuya et al. "A 16 bit Oversampling ADC" Proc. ISSCC 87, Feb. 87, New York.

[10] N.S. Szabo, R. Tanaka. "Residue Arithmetic and its applications to computer technology, New-York, McGraw-Hill 1967

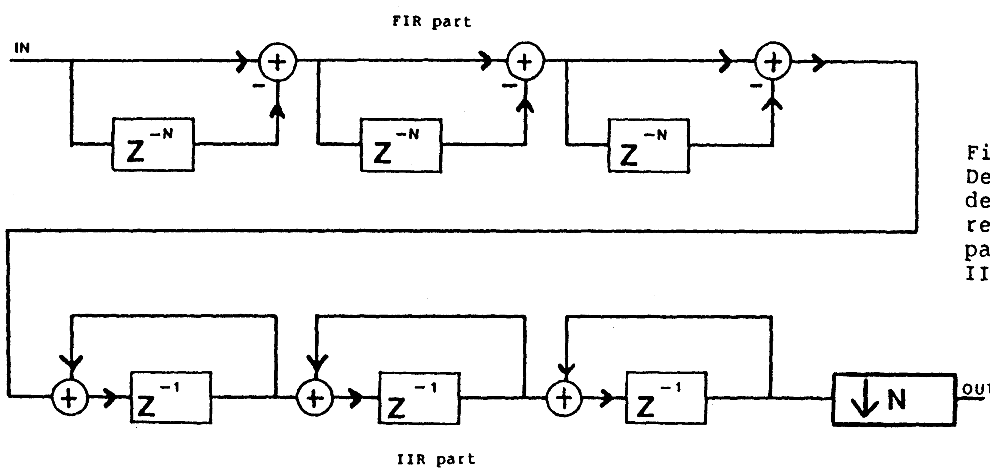


Fig. 1:
Decomposition of comb decimation filters realizing first the FIR part and thereafter the IIR part

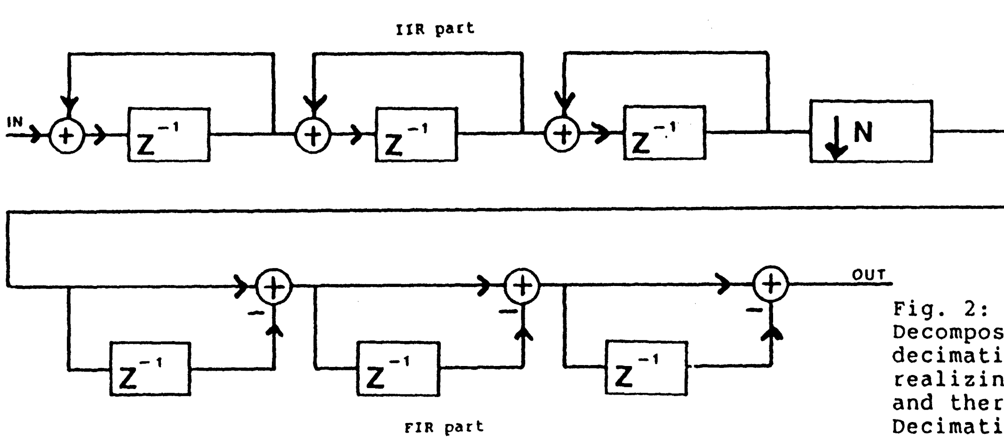


Fig. 2:
Decomposition of comb decimation filter realizing first the IIR part and therafter the FIR part. Decimation is performed in between.

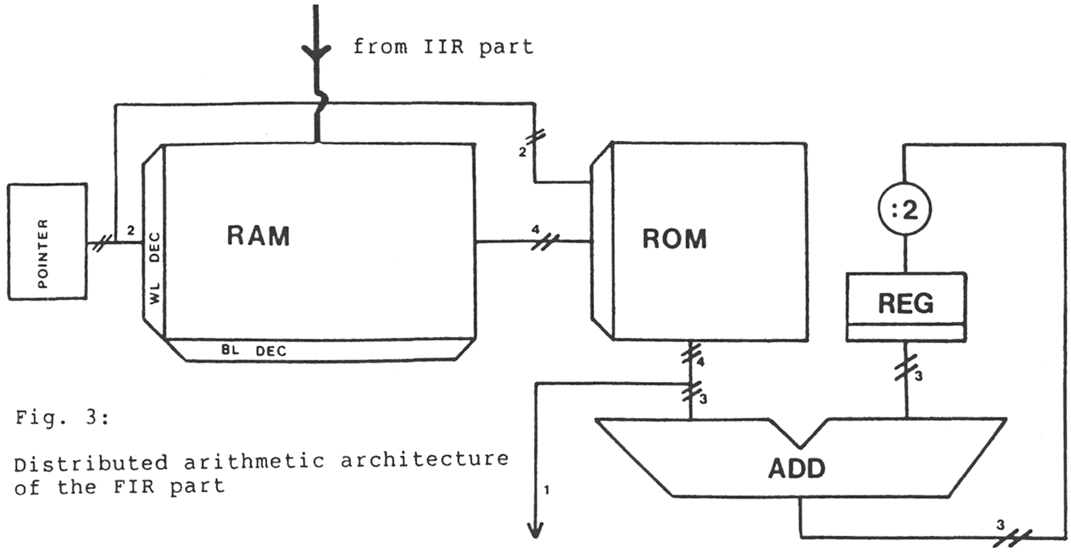


Fig. 3:

Distributed arithmetic architecture of the FIR part

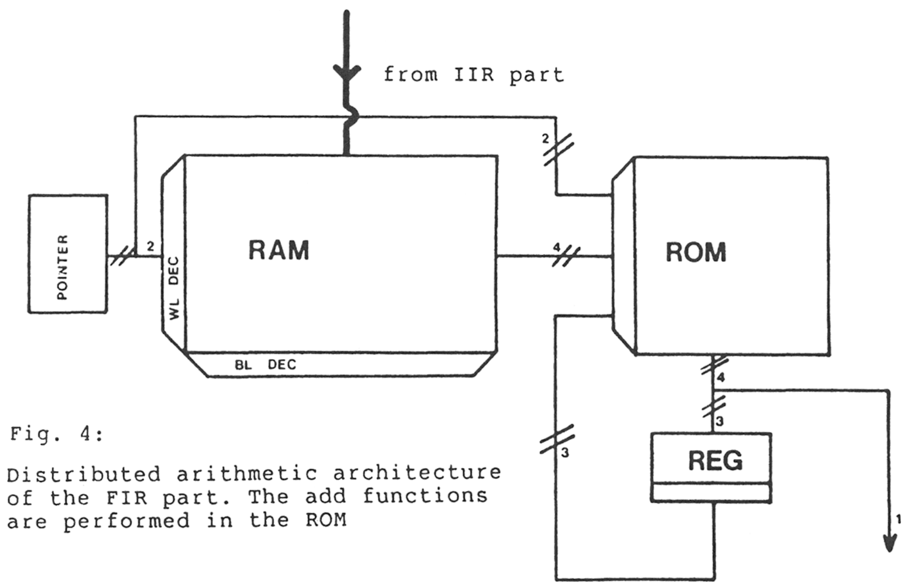


Fig. 4:

Distributed arithmetic architecture of the FIR part. The add functions are performed in the ROM

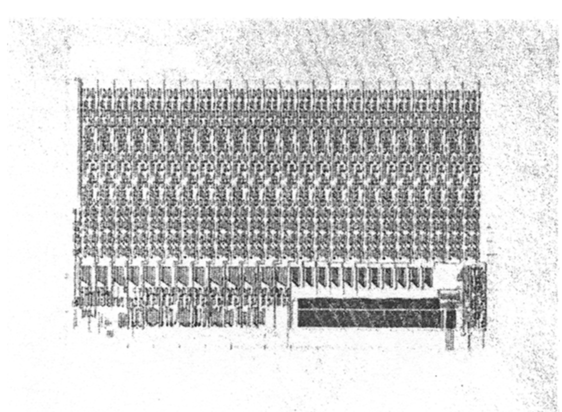


Fig. 5: Chip photograph

Nine Digital Filters for Decimation and Interpolation

DAVID J. GOODMAN, MEMBER, IEEE, AND MICHAEL J. CAREY

Abstract—Filtering is necessary in decimation (decreasing the sampling rate of) or interpolation (increasing the sampling rate of) a digital signal. If the rate change is substantial, the process is more efficient when the decimation or interpolation occurs in stages rather than in one step. Half-band filters are particularly efficient for effecting octave changes in sampling rate and nine digital filters are presented, eight of them half-band filters, to be used as components of multistage interpolators and decimators. Also presented is a procedure for combining the filters to produce multistage designs that meet a very wide range of accuracy requirements (stopband attenuation to 77 dB, passband ripple as low as 0.00014).

The nine filters admit changes between sampling rates above $4W$, where W is the nominal bandwidth of the signal. Established design techniques may be used to obtain efficient filters for conversion between $4W$ Hz sampling and $2W$ Hz, the "baseband sampling rate." With these multistage filters, the possible interpolation and decimation ratios are all integer multiples of powers of two. To overcome this restriction we present a simple resampling technique that extends the range of designs to conversions between any two rates. The interpolation or decimation ratio need not be an integer or even rational. In fact, it can vary slightly as in a practical situation where the input signal and output signal are under the control of autonomous clocks.

We demonstrate the approach by means of several design examples and compare its results with those obtained from the optimization scheme of Crochiere and Rabiner.

I. Introduction

A. Background

SEVERAL RECENT PAPERS [1]-[8] have considered filtering problems that arise in changing the sampling rate of a digital representation of a continuous waveform. It is now recognized that, when the initial rate and final rate are widely separated, it is more efficient to change the sampling rate in stages by means of a sequence of filters than it is to do so all at once with a single filter. The previous papers fall in two categories: some of them offer design methods for choosing decimation (rate reducing) and interpolation (rate increasing) filters from the class of all symmetric finite impulse response (FIR) filters [1], [5]-[8], while others focus on the special class of symmetric FIR filters known as half-band filters [9] in which nearly half of the impulse response coefficients are zero, making them particularly efficient for 2-to-1 decimation and interpolation [2]-[4]. These latter papers describe the efficient use of half-band filters and give examples of practical applications, but none of them provides a design method for finding sequences of half-band filters that meet prespecified fidelity criteria.

Manuscript received December 17, 1975; revised July 12, 1976, and November 10, 1976.

D. J. Goodman is with the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ 07974.

M. J. Carey was with the Post Office Research Center, Martlesham Heath, England. He is now with the University of Keele, Staffordshire, England.

TABLE I
Filter Coefficients

Filter	Order	$h(0)$	$h(1)$	$h(3)$	$h(5)$	$h(7)$	$h(9)$
$F1$	3	1	1				
$F2$	3	2	1				
$F3$	7	16	9	-1			
$F4$	7	32	19	-3			
$F5$	11	256	150	-25	3		
$F6$	11	346	208	-44	9		
$F7$	11	512	302	-53	7		
$F8$	15	802	490	-116	33	-6	
$F9$	19	8192	5042	-1277	429	-116	18

$F1$ can be of any order, with all $h(i) = 1$.
$F2$-$F9$ are half-band filters with $h(-i) = h(i)$ and $h(2) = h(4) = h(6) = 0$.

B. This Paper

Such a design method is the contribution of the present paper, which offers the set of nine filters, denoted $F1$-$F9$, in Table I, and a simple method for selecting cascade combinations of these filters that satisfy a very wide range of accuracy requirements. The filters have been selected with hardware efficiency in mind; some of them can be realized recursively with accumulators alone and in most of the others, the coefficient word lengths are very modest. An important consequence of this hardware-oriented approach is the discovery that even though the required computation rate grows linearly with the ratio of the two sampling rates, the amount of hardware can remain constant beyond a certain ratio.

Although the half-band filters admit 2:1 sampling rate changes, it is possible, by means of a resampling register, to use them to convert between sampling rates that are not related by powers of two. In fact with the resampling method we present, the input and output rates need not even maintain a fixed ratio: the two clocks can be autonomous and slightly variable.

C. Filter Requirements

If a signal, sampled at f_s Hz, has essential information in the band 0, W Hz, we refer to $2W$ as the "baseband sampling rate" and introduce the parameter

$$R = f_s/2W \tag{1}$$

the bandwidth expansion ratio. Fig. 1 shows requirements on filters used in baseband sampling and desampling (conversion between $R = 1$ and $R = \infty$). The passband ripple limit d_p controls linear distortion over the passband, 0 to aW Hz. The stopband attenuation requirement d_s at $f > bW$ controls aliasing in sampling and suppression of spectral images in desampling.

With decimation viewed as an intermediate step in the process of baseband sampling, Fig. 1 is a constraint on the cascade

Reprinted from *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-25, pp. 121-126, April 1977.

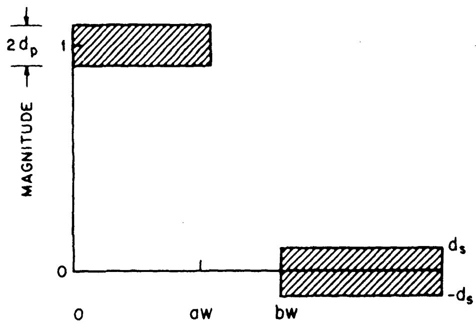


Fig. 1. Filter requirements.

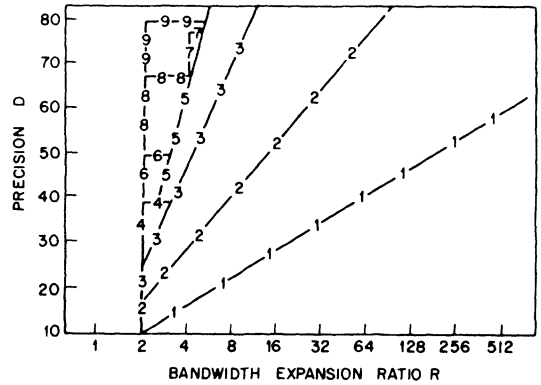


Fig. 2. Design chart.

combination of analog presampling filter and digital decimation filters. Taking the converse view of interpolation, we adopt Fig. 1 as a constraint on the cascade combination of digital interpolating filters and analog desampling filter. Contained in [3] is an example of interpolating and decimating filter sequences that satisfy requirements of telephone transmission systems.

II. Design Method

A. Range of Performance Levels

Our filter set and the method for finding filter sequences that satisfy decimation and interpolation requirements have been derived from an approach that is opposite to that of many filter design techniques. Rather than start with a set of requirements and search for filters that satisfy them, we have selected certain filters that are easy to implement and calculated the sets of requirements satisfied by each filter. These requirements (ranges of speed and accuracy) appear in Fig. 2 as the regions to the right of and below each curve. The ordinate D is either $-20 \log d_s$ or $-20 \log d_p$ and, for a given R, the value of D on a filter locus corresponds to the most stringent accuracy requirement satisfied by that filter. The Appendix provides details of filter characteristics and the derivation of Fig. 2.

The nine filters in our set admit transformations between any pair of bandwidth expansion ratios R_1 and R_2 with $2 \leqslant R_1 < R_2$ with any set of accuracy requirements in the range $-20 \log d_p$ and $-20 \log d_s \leqslant 77$ dB. The filter set does not admit the transformation between baseband sampling rate and $R = 2$. For this transformation, a specially designed filter is necessary and here our work merges with the optimization studies of Crochiere and Rabiner [6], [7] in which the initial or final filter of every efficient sequence is used for conversion between $R = 2$ and $R = 1$.

B. Using the Design Chart

We view multistage decimation as a walk in Fig. 2 from right to left at a certain height D determined by d_p and d_s. Each step in the walk introduces a new filter to the decimation sequence and with the nomenclature $F1$ to $F9$ placing filters in order of increasing complexity, the design strategy is to use at each step in the walk the filter whose locus is immediately to the left of the end of the step. For interpolation, the walk

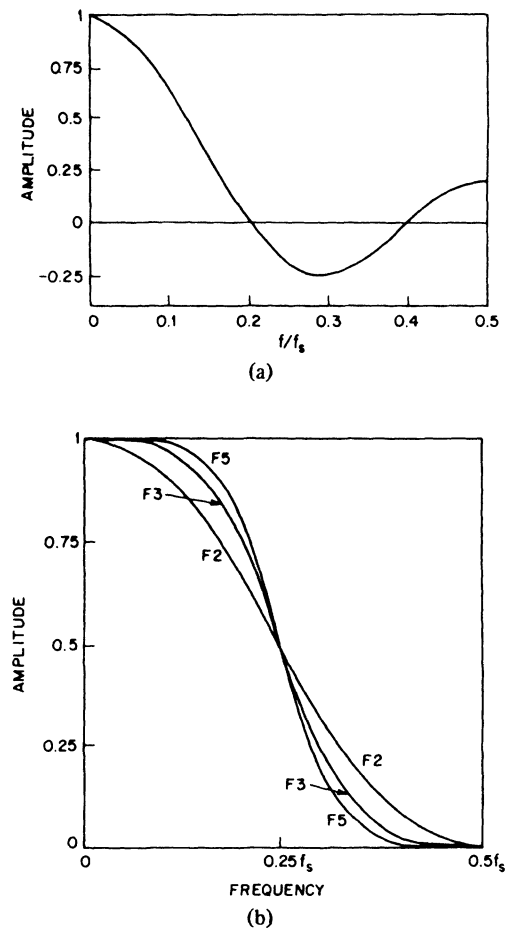


Fig. 3. (a) Amplitude response of $F1$, $N = 5$. (b) Monotonic responses of half-band filters $F2$, $F3$, $F5$.

is from left to right and the strategy is to use the filter with locus immediately to the left of the beginning of each step.

The filter impulse response coefficients are given in Table I. Except for $F1$, an Nth order filter with all coefficients unity, all of the filters are half-band filters. The amplitude responses fall into two categories. Those of $F1$, $F2$, $F3$, and $F5$, shown in Fig. 3, are monotonic over the band 0, W. They have no passband ripple and their rolloff may be equalized by a low-speed filter [3], or the equalization may be built into the passband filter. Consequently, only the stopband re-

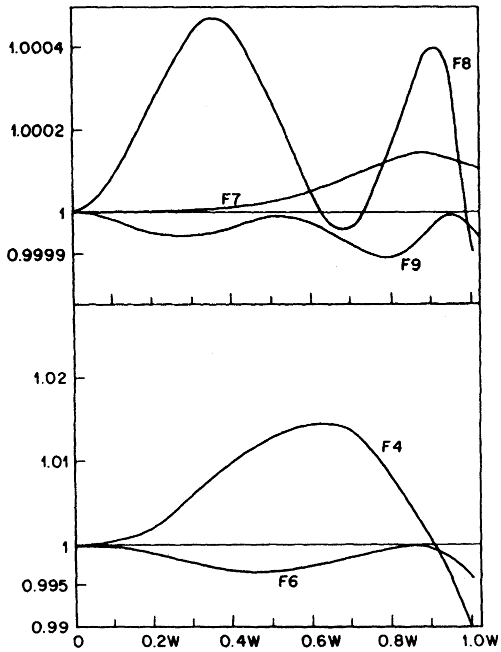


Fig. 4. Passband responses of $F4, F6$-$F9$.

quirement d_s influences their use. The use of the other filters $F4$ and $F6$-$F9$, with passband responses shown in Fig. 4, is constrained by the passband ripple requirement, as well as by d_s.

To obtain a filter sequence, take steps to the left or right in Fig. 2, using at each step the filter with locus just to the left of the entire step. The height D of a step is determined by

$$D = -20 \log d_s, \quad \text{provided this admits the choice of } F1, F2, F3, \text{ or } F5;$$

$$D = -20 \log [\min (d_p, d_s)], \quad \text{otherwise.} \tag{2}$$

Each step length is one octave with $F2$-$F9$. With $F1$, the ratio of the endpoints of the step can be any integer.

C. The Baseband Filter

In the Appendix, it is shown that the filters selected according to the above rules introduce no more than about one-half the allowed passband ripple. It follows that approximate design data for the baseband filter are $d_p/2$, d_s, a, b, with a more precise passband limit obtainable through computation of the overall passband response of the filters chosen from our set. The structure of the baseband filter will depend on the application. Often an infinite impulse response design will be most efficient although there are advantages to the FIR form. In either case many design methods are available [11]. To obtain a half-band design, one may use the optimization program of McClellan *et al.* [12], placing passbands and stopbands with equal error weighting coefficients symmetrically about $f = 0.25$.

D. An Example

Consider a speech signal with nominal bandwidth 4 kHz and design a multistage filter for changing the sampling rate from 1280 kHz ($R = 160$) to 8 kHz ($R = 1$) or vice versa.

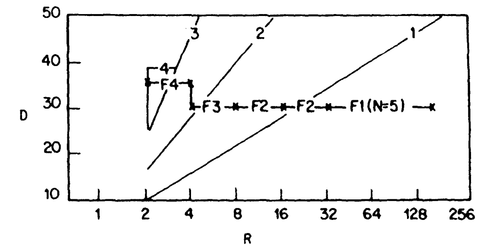


Fig. 5. Use of the design chart to find multistage filters for $R_1 = 1$, $R_2 = 160$.

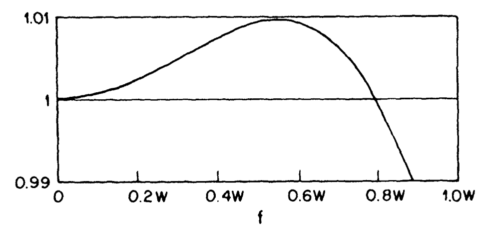


Fig. 6. Passband response of $F1(N = 5), F2, F2, F3, F4$ in cascade.

Assume that the requirements on an 8 kHz presampling filter are: ripple within 0 ± 0.13 dB over 0, 3200 Hz, attenuation at least 30 dB at frequencies above 4800 Hz. In the nomenclature of Fig. 1 these requirements translate to $a = 0.8, b = 1.2$, $d_p = 0.015$, $d_s = 0.031$. For decimation, the walk in Fig. 2 begins at $R = 160$ and with $F1$ admissible, (2) sets the height of the walk at 30 dB. The $F1$ step ends at $R = 32$ ($N = 5$), allowing octave steps for the remaining filters, which are $F2$, $F2$, and $F3$ for conversion from $R = 32$ to $R = 4$. The step from $R = 4$ to $R = 2$ requires one of the filters with passband ripple so that the second line of (2) determines the height of this step, $D = -20 \log d_p = 36.6$ dB, which dictates the selection of $F4$. The steps in the multistage decimation are shown in Fig. 5. (The same steps in reverse order admit interpolation from $R = 2$ to $R = 160$.)

Fig. 6 shows the overall passband response of the five filters used for conversion from $R = 160$ to $R = 2$. It indicates that the rolloff of the first 4 stages and the ripple of $F4$ have partially cancelled one another, producing an overall ripple of magnitude 0.005, which leaves 0.008, at least, for the ripple allowance of the baseband filter. To complete the design, we have derived a half-band filter for conversion between $R = 2$ and $R = 1$. This one is of order 19, with coefficients listed in Table II, which summarizes the characteristic of the 6-stage conversion between $R = 160$ and $R = 1$. Of the 26 multiplications required per baseband sampling interval, 10 are by powers of two, and only 6 have coefficients with more than 4 bits.

III. Resampling

A. General Principle

Although sequences of filters from our set can implement only a limited number of decimation or interpolation ratios, $R_2 : R_1$ (all of them integer multiples of a power of two), an interpretation of the $F1$ locus in Fig. 2 leads to an extension of the design procedure to accommodate any pair of sampling

TABLE II
6-STAGE CONVERSION BETWEEN $f_s = 2W$ AND $f_s = 320W$

Passband: bandedge = $0.8W$ ripple = ± 0.0149 (± 0.13 dB)
Stopband: bandedge = $1.2W$ attenuation = 0.0169 (35.4 db)

	A	B	C	D	E	F	Total
Filter	$F1$	$F2$	$F2$	$F3$	$F4$	b	
Order	5	3	3	7	7	19	
Storage (interpolation)	1	1	1	3	3	9	18
Storage (decimation)	1	1	1	4	4	13	24
Multiplies[a)]	0	0	0	12	8	6	26
Adds[a)] (interpolation)	0	64	32	12	6	9	123
Adds[a)] (decimation)	160	64	32	16	8	10	290

[a)] Computations per $1/2W$ seconds, assumes $F2$ realized with only an accumulator [3].
[b)] Baseband filter coefficients:

i:	0	1	3	5	7	9
$h(i)$:	238	149	-46	22	-12	6.

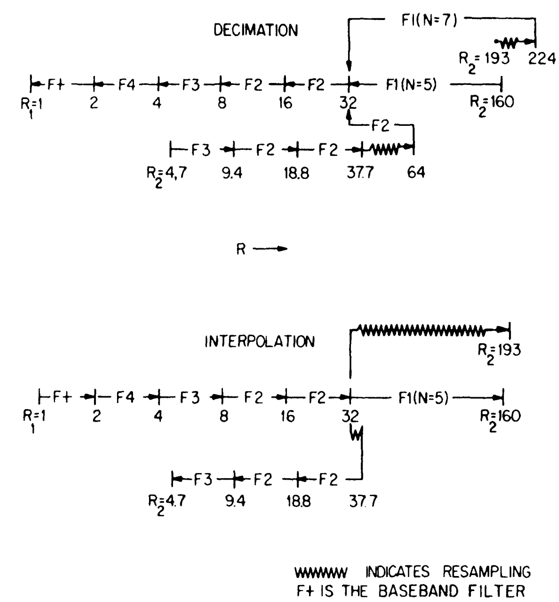


Fig. 7. Examples of interpolation and decimation filters for three values of R_2.

rates. In this interpretation, we imagine a zero-order desampling (holding the value of each sample until the arrival of the next sample) of a signal with $R = R_2$. This desampling, equivalent to filtering with $F1, N = \infty$, is followed by sampling at $R = R' > R_2$. The energy aliased to baseband by the resampling originates at frequencies above $(R' - 1)W$ and it follows that if the point $(R_2, -20 \log d_s)$ is in the admissible region of $F1$, then

$$H_\infty(f) = \frac{\sin(\pi f/2RW)}{(\pi f/2RW)} < d_s, \quad f > (R' - 1)W \tag{3}$$

which implies that the hold and resampling together meet the stopband constraint. Thus any signal to the right of the $F1$ curve (with $D = -20 \log d_s$) may be resampled at a higher rate without incurring excessive aliasing.

Augmented by a resampling register, the filter set can transform (with accuracy up to 77 dB) between any pair of sampling rates greater than $4W$. The two rates need not be related by rational numbers; as in practice, the clocks can be autonomous and each slightly variable. To generalize the design rules to incorporate resampling, we introduce a sampling rate R_2' which is an appropriate multiple (integer times a power of 2) of R_1 and obtain a filter sequence that transforms between R_1 and R_2'. We then use a resampling register (and perhaps additional filters) to convert between R_2' and R_2.

B. Examples

In the example of Section II-D, let the initial sampling rate be 1544 kHz ($R_2 = 193$), the bit rate of a $T1$ transmission line. Because $(R,D) = (193,30)$ is within the admissible region of $F1$, it is possible to resample the signal at $R_2' = 224 = 7 \times 2^5$. Decimation to $R = 1$ can then proceed with $F1$, $N = 7$ for decimation to $R = 32$ and filters $B - F$ in Table II for decimation to $R = 1$. For interpolation from $R = 1$ to $R = 193$, it suffices to use filters $B - F$ to increase the bandwidth expansion ratio to $R = 32$ and the resampling register to increase the rate to $R = 193$.

A more complicated task is to convert between $f_s = 8$ kHz ($R = 1$) and the 37.7 kHz rate ($R = 193/41$) of the delta modulators in a subscriber loop carrier system [10]. The decimation task requires, initially, interpolation to a bandwidth expansion ratio greater than 30 (the value of R on the $F1$ locus at which $D = 30$). To this end one can interpolate (by means of the three filters $F3$, $F2$, $F2$) to $R = 8 \times 193/41 = 37.7$ and then resample at $R_2' = 64$. Now the 64:1 decimation can be accomplished with an $F2$ decimator followed by filters $B - F$. For interpolation from $R_1 = 1$ to $R_2 = 193/41$, use filters B-F to produce $R_2' = 32$. Then resample at $R = 37.7 = 8R_2$ and use $F2$, $F2$, $F3$ for 8:1 decimation to R_2. The filtering and resampling operations of the examples are summarized in Fig. 7.

IV. IMPLEMENTATION

The filters in our set have been chosen with hardware efficiency in mind: careful attention has been given to coefficient word lengths. When half-band filters are used as interpolators and decimators, their complexity increases relatively slowly [2] with filter order N, as indicated in Table III, which shows the number of multiplications increasing as $N/4$, additions as $N/2$ and storage registers as $N/2$ (interpolation) or $3N/4$ (decimation).

$F1$ and $F2$ can be realized without multipliers. $F1$ as an N:1 interpolator is a "hold-for-N" circuit, a single register from which N identical output words are read after each input word is written. With $F1$ an N:1 decimator, an output is the sum of N inputs, which can be obtained from a single accumulator that is reset after each output is generated. It follows that from any point, R, to the right of the $F1$ curve in Fig. 2 it is possible to interpolate to a bandwidth expansion ratio NR with only a single memory element; conversely it is possible to decimate from NR to R with only a single accumulator, regardless of the size of N. The amount of hardware is constant to the right of the $F1$ locus; filter complexity does not grow indefinitely with the highest sampling rate to be accommodated.

TABLE III
COMPLEXITY OF HALF-BAND FILTERS

		Decimation		Interpolation	
Order	Multipliers	Storage	Adders	Storage	Adders
3	2	1	2	1	1
7	3	4	4	3	3
11	4	7	6	5	5
15	5	10	8	7	7
19	6	13	10	9	9
N	$(N+5)/4$	$(3N-5)/4$	$(N+1)/2$	$(N-1)/2$	$(N-1)/2$

Reference [3] shows how $F2$ may be realized with a single accumulator, operating at a rate $2f_s$. As an interpolator, the accumulator output is, alternately, a) the sum of two successive inputs and b) a single input added to itself. As a decimator, the $F2$ accumulator accepts one input sample, adds the next sample twice and a third sample once before generating an output and resetting to zero.

V. Comparison with an Optimization Method

The optimization procedure of Crochiere and Rabiner [6], [7] considers the class of all symmetric FIR filters and produces sequences that minimize either the number of multiplications or the number of storage registers. Their optimum sequences usually contain three or four stages in which the first (interpolation) or last (decimation) stage converts between $R = 2$ and $R = 1$. Half-band filters do not normally appear in their solutions because their procedure does not distinguish between multiplication by zero (the value of the even numbered half-band filter coefficients) and multiplication by nonzero coefficients.

To compare their procedure with our approach we have studied with Crochiere and Rabiner three decimator design problems. The first is the example in Section III-D of this paper which imposes relatively loose accuracy and transition band constraints. The second design appears in [6] and is quite stringent, while the third example, taken from [2], is intermediate between the other two.

The results are summarized in Table IV. In Example A, our method has a significant overall advantage because the baseband filter accounts for only a fraction of the overall complexity, making the efficiencies of $F1$ and half-band filters in the earlier stages quite salient. In Example B, the 100:1 decimation ratio puts our method at a severe disadvantage. With $d_s = 0.0001$, our resampling principle demands interpolation to an intermediate bandwidth expansion ratio of 6400 followed by resampling at $R = 8192$ and decimation to $R = 128$, a procedure that involves hundreds of additions per input sample. Table IV shows the results of only the subsequent 128:1 decimation in Example B. There, as in Example C, the baseband filter accounts for a substantial fraction of the computational complexity and the overall complexity of the two schemes is similar. The added overheads associated with a large number of stages are a disadvantage of our approach which may be only partially offset by the economies of short coefficient words.

TABLE IV
COMPARISONS OF DESIGN METHODS

Example A

$R_2:R_1 = 160$, $d_p = 0.015$, $d_s = 0.03$, $a = 0.8$, $b = 1.2$

Crochiere-Rabiner 3-stage design:

Stage	D[a]	N	Multiplies[b]	Adds[b]
1	20	42.4	1.05	2.10
2	4	13	0.0875	0.175
3	2	19	0.0375	0.063
			1.1750	2.338

Goodman-Carey 6-stage design:

Stage	D	N	Multiplies	Adds
1	5	5	–	1.00
2	2	3	–	0.4
3	2	3	–	0.2
4	2	7	0.075	0.1
5	2	7	0.0375	0.05
6	2	19	0.0375	0.063
			0.150	1.813

Example B

$R_2:R_1 = 100$, $d_p = 0.001$, $d_s = 0.0001$, $a = 0.95$, $b = 1$

Crochiere-Rabiner 3-stage design:

Stage	D	N	Multiplies	Adds
1	10	38	1.9	3.8
2	5	38	0.38	0.76
3	2	356	1.78	3.56
			4.06	8.12

Goodman-Carey 7-stage design[c]:

Stage	D	N	Multiplies	Adds
1	2	3	–	2.56
2	2	7	0.96	1.28
3	2	7	0.48	0.64
4	2	11	0.32	0.48
5	2	19	0.24	0.4
6	2	19	0.12	0.2
7	2	356	1.78	3.56
			3.90	9.12

Example C

$R_2:R_1 = 32$, $d_p = 0.00316 = d_s$, $a = 0.8$, $b = 1$

Crochiere-Rabiner 3-stage design:

Stage	D	N	Multiplies	Adds
1	8	27	1.75	3.5
2	2	8	0.25	0.5
3	2	52	0.813	1.62
			2.813	5.62

Goodman-Carey 5-stage design:

Stage	D	N	Multiplies	Adds
1	2	3	–	2.0
2	2	7	0.75	1.0
3	2	11	0.5	0.75
4	2	15	0.313	0.5
5	2	52	0.813	1.62
			2.376	5.87

[a] Ratio of input rate to output rate.
[b] Per input sample.
[c] Assumes initial $R = 128$ (after resampling). Multiplications and additions per input samples are referred to $R = 100$.

VI. Conclusions

We have shown how a set of nine specially designed filters can be used to cover a wide range of decimation and interpolation accuracy requirements. The filters are efficient in terms of number of multiplications and coefficient word length. Filter sequences (excluding a baseband filter) can be designed quickly, without a computer. The results often compare favorably with those of a more elaborate optimization procedure.

Although a decimator or interpolator with many half-band filters is efficient computationally, it requires more elaborate timing and control circuitry than a scheme with fewer stages. Consideration of these overheads may in practice lead to a single stage design or one produced by the Crochiere and Rabiner method or perhaps a combination of their method and ours using $F1$ or $F2$ at high speeds and filters with interpolation or decimation ratios greater than 2 at lower speeds.

APPENDIX

Filter Characteristics

Except for $F1$, all of the filters are half-band filters: approximations to the ideal low-pass filter with cutoff frequency $f_c = f_s/4$ at the center of the filter operating band. The impulse response coefficients h_i have the property

$$h_{\pm 2} = h_{\pm 4} = \cdots = h_{\pm (N-3)/2} = 0. \tag{A1}$$

They also satisfy the constraint

$$h_0 = 2 \sum_{i=1}^{N-1} h_i. \tag{A2}$$

which implies that each amplitude response function has odd symmetry about f_c

$$H(f_c + \psi) + H(f_c - \psi) = H(0). \tag{A3}$$

Consequently, $H(2f_c) = H(\frac{1}{2}f_s) = 0$ and each filter has at least a double zero at the bandedge $f = \frac{1}{2}f_s$. The zeros strongly suppress spectral images of narrow-band, low-frequency signals such as power line hum and certain speech sounds.

The curves for $F2$-$F9$ in Fig. 2 show the minimum R for which

$$-20 \log \left| \frac{H(f)}{H(0)} \right| \geqslant D, \quad (2R-1)W \leqslant f \leqslant 2RW. \tag{A4}$$

This frequency range contains the spectral images to be suppressed in interpolation from $f_s = 2RW$ to a higher rate. It is the set of frequencies aliased into the baseband when decimation reduces the rate to $2RW$. It follows that if $D_s = -20 \log d_s$, the intersection of a filter curve and the horizontal line $D = D_s$, indicates the lowest R for which that filter meets the stopband specification. The passband ripple requirement d_p influences the selection of $F4$, $F6$-$F9$ the filters with nonmonotonic responses over 0, W Hz. The symmetry of the response of each filter implies that with the sampling rate in the admissible region shown in Fig. 2,

$$1 - d \leqslant H(f)/H(0) \leqslant 1 + d, \quad f \epsilon (0, W) \tag{A5}$$

where $D = -20 \log d$ is the height of the horizontal line for the filter. Moreover, Fig. 4 shows that for each filter one of the two bounds in (A5) is quite loose, i.e., that the peak-to-peak ripple of each filter is nearer d than $2d$. It follows that a filter with $D = -20 \log d_p$ uses about half of the peak-to-peak ripple allowance of the entire sequence.

$F1$ is the class of filters in which all impulse response coefficients are unity. The frequency response of the Nth order $F1$ filter is

$$H_N(f) = \frac{\sin(\pi f/2RW)}{N \sin(\pi f/2NRW)} \tag{A6}$$

in which $2NRW$ is the filter sampling rate. In the band $(0, 2NRW)$ there are $N-1$ zeros located at $2RW, 4RW, \cdots, (N-1)2RW$. For decimation, these are the centers of the frequency bands that are aliased into the signal baseband. In the case of interpolation they are the center frequencies of the spectral images to be suppressed. Over all of these bands, each of width $2W$, the most critical frequency is $2RW - W$. If

$$|H_N(2RW - W)| \leqslant d_s \tag{A7}$$

the filter is sufficiently accurate over its entire range. The $F1$ curve in Fig. 2 corresponds to the equation $H_3(2RW - W) = d_s$. As N increases, precision improves, so that curves for $N > 3$ can be drawn above the $F1$ curve in Fig. 4. This curve does, however, provide a reasonably accurate lower bound on precision. It is within 1.65 dB of the $N = \infty$ curve for all values of R.

REFERENCES

[1] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692-702, June 1973.

[2] M. G. Bellanger, J. L. Daquet, and G. P. Lepagnol, "Interpolation, extrapolation and reduction of computation speed in digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 231-235, Aug. 1974.

[3] D. J. Goodman, "Digital filters for code format conversion," *Electron. Lett.*, vol. 11, pp. 89-90, Feb. 20, 1975.

[4] D. W. Rorabacher, "Efficient FIR filter design for sample-rate reduction and interpolation," in *Proc. 1975 IEEE Int. Symp. Circuits and Systems*, Apr. 21-23, 1975, pp. 396-399.

[5] R. R. Shively, "On multistage finite impulse response (FIR) filters with decimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 353-357, Aug. 1975.

[6] R. E. Crochiere and L. R. Rabiner, "Optimum FIR digital filter implementation for decimation, interpolation and narrow band filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 444-456, Oct. 1975.

[7] —, "Further considerations on the design of decimators and interpolators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 296-311 Aug. 1976.

[8] G. Oetken, T. W. Parks, and H. W. Schussler, "New results in the design of digital interpolators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 301-309, June 1975.

[9] D. W. Tufts, D. W. Rorabacher, and W. E. Mosier, "Designing simple, effective digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 142-158, June 1970.

[10] R. J. Canniff, "Signal processing in SLC-40, a 40 channel rural subscriber carrier," in *Conf. Rec., 1975 IEEE Int. Conf. Communications*, vol. III, pp. 40.7-40.11, June 1975.

[11] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, chs. 3 and 4.

[12] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 506-526, Dec. 1973.

A NOVEL ARCHITECTURE DESIGN FOR VLSI IMPLEMENTATION OF AN FIR DECIMATION FILTER

Hanafy Meleis
Pierre Le Fur

AT&T Bell Laboratories
Murray Hill, NJ 07974

ABSTRACT

A novel architecture design of a one stage FIR filter for decimation is described. It performs the decimation of a 1-bit code at 1024KHz of double integration Sigma Delta modulation output to PCM at 16KHz. This architecture is designed in such a way that it needs only a simple control structure suitable for VLSI implementation. We devised an algorithm for generating the coefficients of the filter with a minimum of required hardware. It does not require storing the coefficients in a ROM and continuously reading it to calculate the convolution. The accumulators needed to perform the direct convolution are arranged in a way that simplifies and minimizes the hardware required for the filter implementation.

The filter response is Sinc^3(f) which provides sufficient attenuation for modulation generated by means of double integration. The implementation of this filter requires the generation of the coefficients and the performance of the convolution. Three coefficients are needed with every input to obtain the output sequence. The major feature of this architecture is the use of an efficient algorithm to obtain the coefficients thereby reducing the area and power consumption. It is very suitable for VLSI implementation in CMOS technology.

1. INTRODUCTION

This paper describes the design of an FIR filter which is used to perform the decimation of a 1-bit code at 1024KHz of a double integration Sigma Delta modulation[1] output to PCM sampled at 16KHz. The filter response is Sinc^3(f) and the duration of the filter impulse response is three resampling periods. Its frequency response has a third order zero at 16KHz resampling rate and all its harmonics.

The problem of implementing the Sinc^3(f) in hardware lies in generating the coefficients of the filter. Previously a Sinc^2(f) filter[2] was used for decimation and conversion of Sigma Delta modulation output to PCM. This filter has a triangularly shaped impulse response and its coefficients can be generated using only a six bit counter. A Sinc^2(f) filter response does not provide sufficient attenuation for modulations generated by means of double integration. A filter with a frequency response equal to Sinc^3(f) is needed. This filter would have incremently non linear coefficients. In this paper we describe a novel method of generating the coefficients of this filter with a minimum of required hardware. Also, two alternative architectures for performing the convolution will be described.

2. FIR FILTER REALIZATION FOR DECIMATION

The process of digitally converting the sampling rate of a signal from a given rate $f_i = 1/T_i$ to a lower rate $f_o = 1/T_o$ is called decimation[3]. Assume x[n] is an input signal with a sampling rate f_i and with full band, i.e., its spectrum is nonzero for all frequency f in the range $-f_i/2 \leqslant f \leqslant f_i/2$. In order to lower the sampling rate and avoid aliasing at the new rate, it is necessary to filter the signal with a digital low-pass filter. An FIR filter can be used to obtain a band-limited input signal[4]. It can be realized by direct application of the convolution equation:

$$y[n] = \sum_{k=0}^{N-1} h[k]\, x[n-k] \tag{1}$$

where h[n] is the impulse response of the filter and $N = f_i/f_o$ is the decimation factor. x[n] is the input signal and y[n] is the filter output signal. The sample rate reduction is achieved by extracting y[n] every Nth sample and forming a new sequence as an output of the decimator.

In applications wherein the analog signal is encoded by means of delta modulation, the requirement is that the low pass filter supply an adequate attenuation of the modulation noise[2]. In the case of double integration Sigma Delta modulation, a filter having Sinc^3(f) response is required. The impulse response of this filter is:

$$h[n] = \frac{n(n+1)}{2} \qquad \text{for } 1 \leqslant n < N \tag{2}$$

$$h[n] = \frac{N(N+1)}{2} + (n-N)(2N-1-n) \qquad \text{for } N \leqslant n < 2N \tag{3}$$

$$h[n] = \frac{(3N-n-1)}{2} \qquad \text{for } 2N \leqslant n < 3N \tag{4}$$

where N is the decimation factor.

The filter we designed decimates the 1-bit code at 1024KHz to 16KHz PCM. In this case the filter has 190 coefficients and N=64. The action of the filter as a decimator in the time domain is an averaging of 190 samples with 1μsec being the period of the input sequence and 64 the period of the output sequence. In the frequency domain the filter response is:

$$H(\omega) = \left(\frac{\sin\omega NT}{\sin\omega T}\right)^3 \tag{5}$$

This filter has a third order zero at 16KHz resampling rate and all its harmonics. The realization of this filter in the time domain using the following convolution equation:

$$y[n] = \sum_{k=0}^{189} h[k]\, x[n-k] \tag{6}$$

requires the generation of the coefficients and performance of the convolution. The input sequence is a 1-bit sequence, this reduces the complexity of the filter by eliminating the required multi-bit multiplier for calculating the convolution. Only accumulators are needed. Also, the decimation of the input sequence with period T =1μsec to a new sequence with period T =64μsec requires generating three different coefficients every period of 1μsec and performs three accumulations in this period. Figure 1 shows the three simultaneous accumulations needed to filter the data and lower the sampling rate by a factor of 64. The envelope in this figure represents the values of the coefficient. For every 64 samples of the input sequence, a calculation of an output starts by accumulating the corresponding coefficient if the input sample equals 1 for a period of 190 input samples. This output will be extracted after 192 samples of the input signals to form the new sequence.

Reprinted from *IEEE Proc. ICASSP'85*, pp. 1380–1383, March 1985.

An efficient design for VLSI implementation of this decimator required minimum storage to minimize the area required on the chip and a simple control to minimize the power consumption. We designed this decimator with a minimum of required hardware. In the next section we discuss the architecture for this decimator.

3. ARCHITECTURE DESIGN

We designed the $Sinc^3(f)$ filter for decimating the delta modulation output sequence as a one stage decimator. The architecture design consists of two modules, one for generating the coefficients and the other for performing the direct convolution.

3.1 Generating the coefficients

Equations 2, 3 and 4 represent the coefficients of the filter. One method of implementing this filter can be accomplished by storing the coefficients in a ROM and reading it three times every period of the input sequence to form the output sequence as described in section 2. This implementation does require more hardware than in our method and the rate of changing the control signals in the ROM implementation will lead to a higher power consumption in CMOS implementation.

The coefficients of this filter are incremently nonlinear as shown in equations 2, 3 and 4. We generated these coefficients with a minimum of required hardware by calculating the incremental values in every section of the impulse response. From equation 2, the incremental value between two successive coefficients is:

$$\Delta_1[n] = n+1 = n_c \qquad 1 \leqslant n_c \leqslant 64 \ . \tag{7}$$

Where n_c is the output of a seven bit counter which counts from 1 to 64. From equation 3 the incremental value will be :

$$\Delta_2[n] = N - 2(n - N - 1) \qquad N \leqslant n < 2N \ .$$

The value n-N-1 can be replaced by n_c and Δ_2 can be written as follows:

$$\Delta_2[n] = N - 2n_c \qquad 1 \leqslant n_c \leqslant 64 \ . \tag{8}$$

In the same manner equation 4 can be used to calculate the incremental values in the third portion of the impulse response :

$$\Delta_3[n] = -N + (n - 2N + 1) \qquad 2N \leqslant n < 3N$$

It can be written in relation to n_c as follows·

$$\Delta_3[n] = -N + n_c \quad 1 \leqslant n_c \leqslant 64 \tag{9}$$

The hardware used to generate the coefficients are one seven bit counter, three twelve bit adders, three twelve bit registers, three half adders and control logic. The counter is driven by a 1024KHz clock and changes its output with the negative edge of this clock. The three registers latch with the positive edge of the 1024KHz clock. The adders used are carry ripple adders. Figure 2 shows the architecture of the first module of the design. CR1, CR2, and CR3 are the coefficient registers. MC is the master counter which counts from 1 to 64. The control logic generates a reset pulse every 64th cycle of 1024KHz. It also generates two signals, DSC0 and DSC1, which count three frames of 64 cycles each. These signals will be used in the second module to generate the output sequence.

The first section of the impulse response contains the coefficients from n=1 to n=64. The value of the first coefficient equals 1 and the 64th coefficient equals 2080 (equation 2). The incremental value between any two successive coefficients equals the counter output nc(equation 7). Figure 3 shows the time diagram for the counter and the coefficient registers. With every negative edge of the 1024KHz clock the counter generates its output. A twelve bit adder is used to accumulate the previous coefficient to the output of the counter. The new coefficient is stored in the first coefficient's register CR1 at the positive edge of the 1024KHz clock . At the end of the 64th cycle the reset pulse (figure 3) initializes CR1 to a value zero and the coefficient starts again from the first value.

The second section of the impulse response contains the coefficients from n=65 to n=128. The 65th coefficient equals 2142 and the 128th equals 2016 (equation 3). Equation 8 shows the incremental value in this portion of the impulse response. To explain the method of generating the value N-2nc from the counter output we can rewrite this equation as follows:

$$\begin{aligned} \Delta_2[n] &= 64 + 2(-n_c) \\ &= 64 + \overline{2n_c} + 1 \ . \end{aligned} \tag{10}$$

where $\overline{2n_c}$ is the complement of nc after it shifted to the left by 1 bit. The value 1 in equation 10 is added in order to obtain the 2's complement of 2nc. Figure 4 shows the process of obtaining the value of equation 10. Generating $\Delta_2[n]$ will require only three half adders. Adding 1 in the first bit is accomplished by holding the carry input of the adder to a logic level one. Adding this incremental value to the previous coefficient in the CR2 register will generate the new coefficient. After 64 cycles, the reset pulse will initialize the CR2 register to the value 2080. The first coefficient in this section is generated by adding the first incremental value (62) and so on.

The third section of the impulse response represents the 129th coefficient to the 190th. The coefficient register CR3 is initialized with the value of the 128th coefficient and the incremental value is added to generate the corresponding coefficient in this section. From equation 9, the incremental value -N+nc can be constructed from the first six bits of the counter. The value -N (where N=64) in 2's complement is 7700 in octal. Thus the last six bits of the adder are connected to logic level one and the first six bits to the first six bits of the counter MC.

From the above discussion, we can summarize the algorithm of generating the coefficients in two steps. The first step is initializing CR1, CR2 and CR3 every 64 cycles of 1024KHz. CR1 is initialized to zero, CR2 to the 64th coefficient and CR3 to the 128th coefficient. Secondly, the incremental value in every section of the impulse is generated every clock cycle from the output of the counter MC. These values are accumulated to the values in CR1, CR2 and CR3 to generate the new values of the coefficients.

3.2 Performing the direct convolution

The second module of the architecture performs the convolution to obtain the output sequence as shown in equation 6.

As discussed in section 2, the input data is a 1-bit sequence at 1024KHz and three accumulations are needed for every cycle (see figure 1). A calculation of an output starts every 64 samples of the input sequence. The corresponding coefficient accumulates if the input sample equals 1 for a period of 190 input samples. The output is extracted every 64th sample of the input sequence. This will achieve the decimation of the data by an integer factor of 64.

There are two ways to implement this module depending on the frequency of the master clock. If the available master clock frequency is 1024KHz, the three accumulations needed during every sample are obtained in a parallel fashion. Figure 5 shows the hardware required in this case. It consists of three 12 bit multiplexers, three 19 bit adders, three 19 bit registers, a 19 bit multiplexer and a control logic. The 19 bits are needed to accommodate the maximum number that can be obtained from the accumulations of all the coefficients. The control logic uses DSC0 and DSC1 signals which generate from the first module. Then it generates the control signals for the multiplexer and a reset signal for the registers DR1, DR2 and DR3 . The control signals CM1, CM2 and CM3 of the multiplexer are generated in such a way that when MUX1 is using the first section of the impulse response, MUX2 is using the third section and MUX3 is using the second section. The output of these multiplexers equals zero if the input sequence is zero. C1, C2 and C3 represents the coefficient values from the registers CR1, CR2 and CR3. In order to understand the function of the second module examine figure 1 and figure 5 simultaneously. Every series combination of multiplexer, adder and register in figure 5 accumulates one output. Figure 1 shows that three outputs are constructed simultaneously. And 190 samples of the input are needed to form an output. The output sequence is extracted every 64th input sequence. The control signals of the multiplexer MUXO are designed in such a way to pass the data from the register which is ready to give an output. A reset signal will clear this register after reading its content and a new cycle of forming the output will start.

The second method of implementing the accumulators needed to perform the convolution required a 4096KHz master clock. Figure 6 shows the hardware used. It consists of a 12 bit multiplexer, a 19 bit adder, three 19 bit registers and a 19 bit output register. The adder and the three 12 bit registers are connected in a series array. The inputs of the adder are a coefficient and the output of the last register DR1 in the array. The value in DR1 is added to a coefficient and stored in DR3, the value of DR3 is shifted to DR2 and DR2 is shifted to DR1. This process takes place three times in a 1μsec period. This requires the generation of a 3072KHz clock to latch the registers DR1, DR2 and DR3. Figure 7 shows the 3072KHz master-slave clock generated from the 4096KHz clock. In a period of 1 μsec there are three slave signals and three master signals. With every master-slave the MUXI supplies the adder with the necessary coefficient to be added to DR1. The output register reads the output of DR1 every 64 cycles of 1024 KHz during a slave clock. It reads the first frame during SL1, the second during SL2 and the third during SL3.

In the previous section we demonstrated two alternative methods of implementing the hardware required to perform the convolution. We demonstrated that less hardware is required if 4096KHz is available. As shown in figures 5 and 6 use of the higher master clock frequency eliminates two 19 bit adders and two 19 bit multiplexers.

4. CONCLUSION

A novel architecture design of a one stage FIR filter for decimation with a Sinc^3(f) response has been described. It performs the decimation of a 1-bit code at 1024KHz to PCM at 16KHz. The algorithm used here for generating the coefficients of the filter minimized the hardware required for implementation.

The architecture and logic design presented in this paper has been tested using the standard TTL board level implementation.

REFERENCES

[1] J. C. Candy, "A Use of Double Integration in Sigma Delta Modulation," IEEE Transactions on Communications Vol. COM-33, March 1985.

[2] J. C. Candy, Bruce A. Wooley and O'Connell J. Benjamin, "A Voiceband Codec with Digital Filtering," IEEE Transactions on Communications, vol. COM-29, June 1981.

[3] R. E. Crochiere and Lawrence R. Rabiner, "Interpolation and Decimation of Digital Signals - A Tutorial Review," Proc. IEEE, vol. 69, pp. 300-331, March 1981.

[4] R. R. Shively, "On Multistage Finite Impulse Response (FIR) Filters with Decimation," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, August 1975.

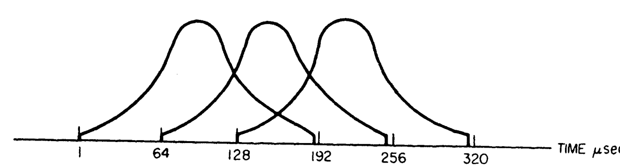


FIGURE 1 EVERY 1μsec THREE ACCUMULATIONS ARE NEEDED. THE ENVELOPE REPRESENTS THE VALUE OF THE COEFFICIENTS

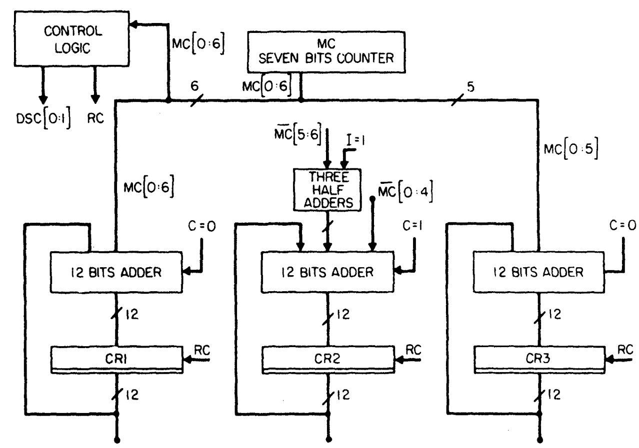


FIGURE 2 THE HARDWARE REQUIRED TO GENERATE THE COEFFICIENTS OF SINC^3(f) FILTER WITH CUTOFF FREQUENCY AT 16 kHz

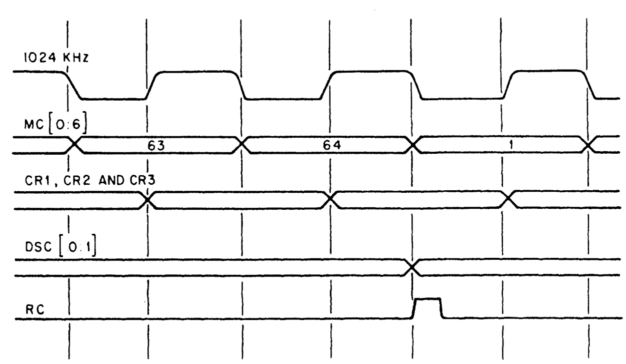


FIGURE 3 TIMING DIAGRAM OF THE ARCHITECTURE SHOWN IN FIGURE 2

BIT NUMBER	8	7	6	5	4	3	2	1	0
$\overline{n_c}$	1	1	X	X	X	X	X	X	X
$\overline{2n_c}+1$	1	X	X	X	X	X	X	X	1
64	0	0	0	1	0	0	0	0	0
	B_8	B_7	B_6	X	X	X	X	X	1

FIGURE 4 GENERATING $\Delta_2[n] = 64 + \overline{2n_c} + 1$
B_6, B_7, B_8 ARE GENERATED USING THREE HALF ADDERS

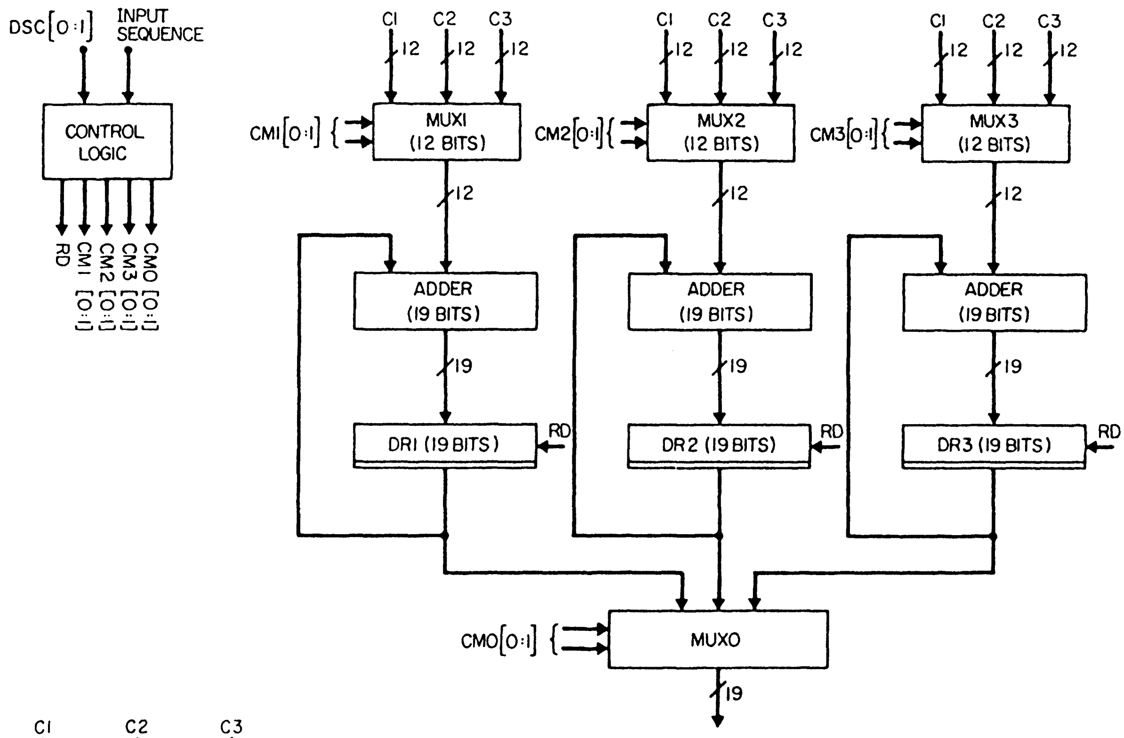


FIGURE 5 THE MODULE OF THE DECIMATOR WHICH PERFORMS THE CONVOLUTION AND THE DECIMATION

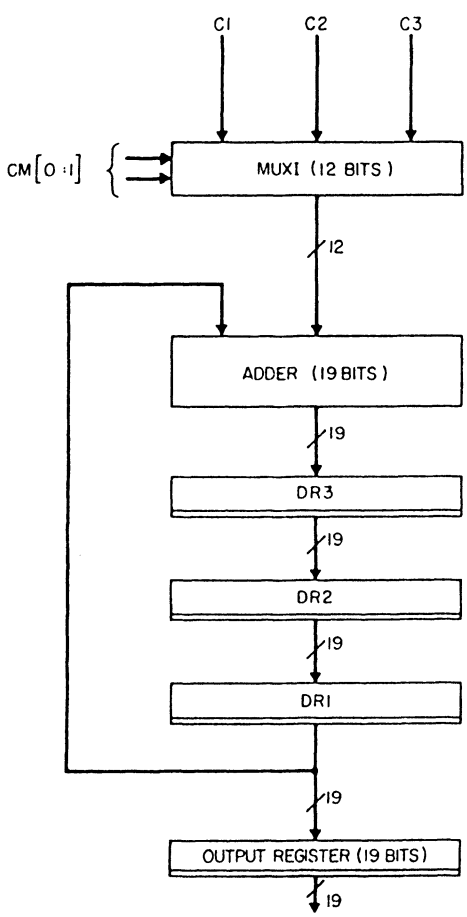


FIGURE 6 ALTERNATIVE DESIGN FOR THE MODULE SHOWN IN FIGURE 5

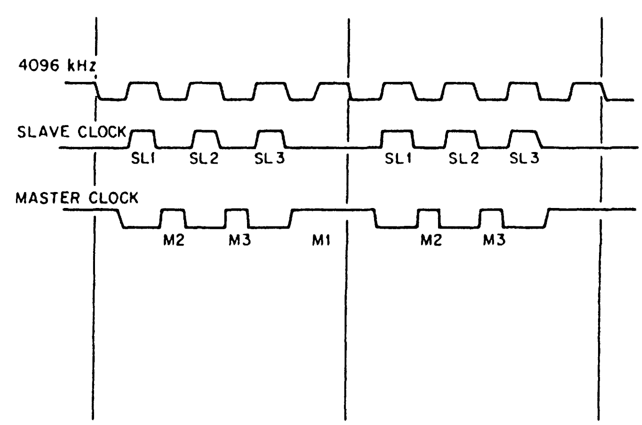


FIGURE 7 CLOCKS NEEDED TO LATCH THE DATA IN TO DR1, DR2 AND DR3 OF THE ARCHITECTURE SHOWN IN FIGURE 6.

Efficient VLSI-Realizable Decimators for Sigma-Delta Analog-to-Digital Converters

Tapio Saramäki and Hannu Tenhunen

Department of Electrical Engineering
Tampere University of Technology
P. O. Box 527, SF–33101 Tampere, Finland

Abstract **– This paper introduces a class of efficient linear-phase FIR decimators for attenuating the out-of-band noise generated by a sigma-delta analog-to-digital modulator. These decimators contain no general multipliers and very few data memory locations, thereby making them easily VLSI-realizable. This is achieved by using several decimation stages with each stage containing a small number of delays and arithmetic operations. The output sampling rate of these decimators is the minimum possible one, unlike for most other existing designs and the proposed decimators can be used, with very slight changes, for many ovesampling ratios. Futhermore, these decimators attenuate highly the undesired out-of-band signal components of the input signal, thus significantly relaxing the anti-aliasing prefilter requirements.**

I. Introduction

Efficient high resolution analog-to-digital conversion is obtained by using oversampled sigma-delta modulation with one-bit quantization [1], [2]. Modulation together with oversampling moves most of the quantization noise out of the baseband [3], [4]. The noise lying out of the baseband can then be reduced by using a decimator [5].

One of the major obstacles in the sigma-delta converter design is the efficient small area integration of the decimator. In this paper, we report a novel high performance linear-phase FIR filter structure which can be easily implemented in small area. In order to optimize both the decimator performance (noise, baseband frequency) and the VLSI realizability (circuit area, power, speed), the proposed decimators are designed to consist of several stages with each stage requiring a small number of arithmetic operations. The optimization is performed in such a way that no general multipliers are required. The overall filter is constructed using a fixed part and an adjustable part. With slight programmable changes in the adjustable filter part, the overall filter can be used for many oversampling ratios. Moreover, the output sampling rate is the minimum possible one, unlike for most other existing designs, and the proposed decimators attenuate highly the undesired input signal components lying out of the baseband, thereby relaxing the anti-aliasing prefilter requirements.

II. Statement of the Problem

The block diagram for the overall system is depicted in Fig 1. The output sampling rate of the sigma-delta modulator is M times the final sampling rate f_s. We assume that the sigma-delta modulator is of second order so that the spectral density of the noise at its output is [5]

$$E(f) = \frac{\sigma^2}{12}\frac{2}{Mf_s}\left[2\left(1-\cos\left(\frac{2\pi f}{Mf_s}\right)\right)\right]^2, \qquad (1)$$

where σ is the spacing of the quantization levels. We state the following amplitude requirements for the decimator

$$1+\delta_p \le |H(e^{j2\pi f/(Mf_s)})| \le 1-\delta_p \quad \text{for} \quad 0 \le f \le \frac{3}{4}\frac{f_s}{2} \qquad (2a)$$

$$|H(e^{j2\pi f/(Mf_s)})| \le \delta_s \quad \text{for} \quad \frac{5}{4}\frac{f_s}{2} \le f \le M\frac{f_s}{2}, \qquad (2b)$$

where $\delta_p = 0.01$ and $\delta_s = 0.001$ (60-dB stopband attenuation). In addition, it is required that that the noise spectral density at the output of the decimator filter before sampling rate reduction

$$\widehat{E}(f) = |H(e^{j2\pi f/(Mf_s)})|^2 E(f) \qquad (3)$$

is on the region $[f_s/2, Mf_s/2]$ well below the level on the baseband $[0, f_s/2]$. This guarantees that the contribution of the aliased components to the overall baseband noise becomes very small.

III. Proposed Class of Decimators

To reduce the arithmetic complexity of the decimator, it is preferred to construct it using several low-order stages, instead of one high-order stage. A multistage implementation of the proposed decimator is given in Fig. 2. It consists of an adjustable filter part and a fixed filter part. The adjustable filter part enables us to use the same structure for various values of M. The values of M in use are M = 16, 32, 64, 128, 256, 512. The transfer function of the equivalent single-stage design can be written in the form

$$H(z) = H_3(z^M)H_2(z^{M/2})H_1(z^{M/4})F_2(z^{M/16})F_1(z), \qquad (4a)$$

where

$$F_2(z) = 2^{-8}\left[\frac{1-z^{-4}}{1-z^{-1}}\right]^4 \qquad (4b)$$

$$F_1(z) = 2^{-P}\left[\frac{1-z^{-K}}{1-z^{-1}}\right]^3 \qquad (4c)$$

with

$$K = \begin{cases} 32 & \text{for } M = 512 \\ 16 & \text{for } M = 256 \\ 8 & \text{for } M = 128 \\ 4 & \text{for } M = 64 \\ 2 & \text{for } M = 32 \\ 1 & \text{for } M = 16 \end{cases} \qquad (4d)$$

and

$$P = \begin{cases} 15 & \text{for } M = 512 \\ 12 & \text{for } M = 256 \\ 9 & \text{for } M = 128 \\ 6 & \text{for } M = 64 \\ 3 & \text{for } M = 32 \\ 0 & \text{for } M = 16. \end{cases} \qquad (4e)$$

Here, $F_1(z)$ is the transfer function of the adjustable filter part in the case where the sampling rate reduction is not performed and $F_2(z)$ is the transfer function from the input of the fixed filter

Reprinted from *IEEE Proc. ISCAS'88*, pp. 1525–1528, June 1988.

part to the input of $H_1(z)$. The term in parentheses in Eqn. (4c) can be rewritten in the form

$$\frac{1-z^{-K}}{1-z^{-1}} = \sum_{r=0}^{K-1} z^{-r}.$$

Similarly, the term in the parentheses of Eqn. (4b) can be expressed in the above form. Thus these terms correspond to linear-phase FIR filters. Linear-phase filters with transfer functions consisting of the above recursive terms have been been used for sampling rate alteration in [6] and [7]. Using the techniques proposed in these papers, we can implement $F_1(z)$ and $F_2(z)$ using the substructures shown in Fig. 2. We note that when the feedforward term $1 - z^{-K}$ is transfered after the sampling rate reduction by a factor of K, it becomes $1 - z^{-1}$. It should be noted also that if 1's or 2's complement arithmetic (or modulo arithmetic in general) and the worst-case scaling are used, the output of the filters $F_1(z)$ and $F_2(z)$ implemeted as shown in Fig. 2 is correct even though there may occur overflows in the feedback loops realizing the term $1/(1-z^{-1})$. The proofs of this fact can be found in [7] and [8]. Also, under the above conditions, the effect of temporary miscalculations vanishes from the output in finite time and initial resetting is not necessary needed. The scaling constant 2^{-P} and 2^{-8} have been selected according the worst-case scaling. Later on, we return to the selection of the number of terms in $F_1(z)$ and $F_2(z)$.

We have designed the remaining transfer functions $H_1(z)$, $H_2(z)$, and $H_3(z)$ using two different approaches to be described later on. Figure 3 gives the amplitude response of the overall design in these two cases for $M = 64$. For the first design, $H_3(z)$ is absent and $H_1(z)$ and $H_2(z)$ are linear-phase FIR filters of orders 5 and 22, respectively. By exploiting the symmetry in the filter coefficients, these filters require 3 and 12 multipliers, respectively. These two filters have been optimized in such a way that the overall filter meets the specifications of Eqn. (2) in the frequency region $[0, 2f_s]$. The optimization has been accomplished using the methods proposed in [9] and [10]. The transfer function $H_1(z)$ has all the zeros on the unit circle and the locations of these zeros have been determined such that the overall filter response exhibits an equiripple behavior on the stopband region $[11f_s/8, 2f_s]$ which alias to the region $[0, 5f_s/8]$ after decimating by a factor of two at the output of $H_1(z)$. $H_2(z)$, in turn, has been designed to provide for the overall amplitude response the desired equiripple nature on the passband $[0, 3f_s/8]$ and on the stopband region $[5f_s/8, f_s]$. In addition to providing the desired attenuation, this filter equalizes the passband distortion caused by the earlier filter stages. We note that when changing M, the change in the response of $F_1(z)$ is negligible on $[0, 2f_s]$ and the response of $F_2(z)$ remains the same on this region. This enables us to use the same filters $H_1(z)$ and $H_2(z)$ for all the values of M. In the actual implementations of $H_1(z)$ and $H_2(z)$ we have exploited the fact that only every second output needs to be computed. For the multirate FIR filter structures exploiting the coefficient symmetry, see, e.g., [11].

The disadvantage of the first design is that it is not possible to use powers-of-two or sums or differences of two powers-of-two to represent the filter coefficients. If rounding is used, the filter coefficients need approximately 10-bit representations to meet the given amplitude criteria. To overcome this limitation, we have designed, as a second alternative, special tailored subfilters which have been optimized such that the resulting overall filter does not require general multipliers. The transfer functions are

$$\begin{aligned} H_1(z) =&(2^{-5} - 2^{-10})(1 + z^{-5}) + (2^{-3} + 2^{-9})(z^{-1} + z^{-4}) \\ &+ (2^{-2} - 2^{-6})(z^{-2} + z^{-3}), \end{aligned} \tag{5}$$

$$H_2(z) = 2^{-1}z^{-15} + \widehat{H}_2(z^2), \tag{6a}$$

where

$$\widehat{H}_2(z) = F(z)[(2^0 - 2^{-2})z^{-5} + (-2^{-2} + 2^{-9})[F(z)]^2], \tag{6b}$$

with

$$\begin{aligned} F(z) =&(2^{-4} + 2^{-6})(1 + z^{-5}) + (-2^{-3} - 2^{-5} - 2^{-6})(z^{-1} + z^{-4}) \\ &+ (2^{-1} + 2^{-3})(z^{-2} + z^{-3}), \end{aligned} \tag{6c}$$

and

$$\begin{aligned} H_3(z) =&(2^0 + 2^{-1} - 2^{-6})[(2^{-6} + 2^{-9})(1 + z^{-4}) \\ &+ (-2^{-4} - 2^{-5})(z^{-1} + z^{-3}) + (2^0 + 2^{-6})z^{-2}]. \end{aligned} \tag{7}$$

As for the first design, $H_1(z)$ provides the desired attenuation on the region $[11f_s/8, 2f_s]$. $H_2(z)$ is a special half-band filter which can be implemented effectively using a polyphase structure based on the commutative model [11]. The resulting structure is shown in Fig. 4. One of the branches is a pure delay term. The other branch is a tapped cascaded interconnection of three identical fifth-order filters. This filter has been designed to provide the desired attenuation on $[5f_s/8, f_s]$. The actual design of this filter has been accomplished by properly modifying the methods proposed in [12] for optimally designing FIR filters as a tapped cascaded interconnection of identical subfilters. Since $H_2(z)$ is a half-band filter, it cannot be used for compensating the passband distortion caused by the earlier filter stages. For this purpose we use $H_3(z)$.

IV. FILTER PERFORMANCE

The performance of the overall A/D converter of Fig. 1 is limited by the modulator limitations [13] and by the performance of the decimator due to the aliasing of the noise into the baseband. The decimator filter performance has been examined by assuming that the sigma-delta modulator is ideal with the output noise spectral density as given by (1). For the first decimator design, the overall output noise powers for the used values of M have been evaluated. An illustrative way is to express these values in terms of the number of additional bits defined by

$$\text{number of additional bits} = \log_4\left(\frac{\sigma_0^2}{\sigma_{\text{out}}^2}\right),$$

where σ_{out}^2 is the output noise power and $\sigma_0^2 = \sigma^2/12$ is the noise power generated by using direct rounding of data at the final sampling rate of f_s. Note the accuracy increases by one bit if the noise power is reduced to be one fourth. The evaluated number of additional bits for various values of M is

$$\begin{aligned} 8.2 \quad &\text{for} \quad M = 16 \\ 10.7 \quad &\text{for} \quad M = 32 \\ 13.2 \quad &\text{for} \quad M = 64 \\ 15.7 \quad &\text{for} \quad M = 128 \\ 18.2 \quad &\text{for} \quad M = 256 \\ 20.7 \quad &\text{for} \quad M = 512. \end{aligned}$$

In calculating the above figures, the overall output noise power obtained after decimation has been taken into consideration (not only in the passband). The dashed and solid lines in Fig. 5 give before decimation the noise spectra before and after filtering, respectively, whereas the dashed and solid lines in Fig. 6 show the spectra on the region $[0, f_s/2]$ at the ouputs of the modulator and the overall system, respectively. The spectrum given by the solid line of Fig. 6 contains thus also the contribution of the aliased components. The spectra have been scaled in such a way that 0 dB is the level of the noise spectrum which is obtained by using direct rounding of data at the final sampling rate of f_s. For the second tailored design, the number of additional bits is the same. It is interesting to observe from Fig. 6 that in the passband region

$[0, (3/4)f_s/2]$ the contribution of the aliased noise to the overall output noise is negligible. The only exception is the beginning of the passband where the output of the sigma-delta modulator contains very little amount of noise. Another interesting feature is that the noise level of the overall system is lower than that of the sigma-delta modulator in the region $[(3/4)f_s/2, f_s/2]$. This shows that the proposed decimator attenuates very effectively the out-of-band noise. If only the noise in the passband region $[0, (3/4)f_s/2]$ before decimation is taken into consideration, the above values increase only by 0.7 bits.

From Fig. 3, it is seen that except for the very beginning of the stopband, the stopband attenuation of both designs is much higher than 60 dB. This is a desired property since filters of this kind attenuate the stopband noise generated by the sigma-delta modulation well below the noise level in the baseband (see Fig. 5). Therefore, the contribution of the stopband noise to the overall noise obtained after decimation is very small. The fixed filter part $F_1(z)$ controls the stopband peaks around the frequencies $32f_s/2$ and $64f_s/2$. Selecting three terms in $F_1(z)$, as given in Eqn. (4c), guarantees that these peaks in the filtered noise spectrum are well below the level of the first peak around the zero frequency (see Fig. 5). If only two terms had been selected, then these peak would be at the same level. The other peaks in the stopband are controlled by $F_2(z)$ and as seen from Figs. 3 and 5, four terms in $F_2(z)$ guarantee that these peaks are well below the noise level in the baseband and the overall filter satisfies the given amplitude criteria.

To show the efficiency of the proposed structure, we have estimated the minimum lengths of optimal direct-form FIR filters to meet the specifications of Eqn. (2) for the various values of M. The minimum lengths are 163, 326, 651, 1302, 2603, and 5206 for $M = 16$, $M = 32$, $M = 64$, $M = 128$, $M = 256$, and $M = 512$, respectively. The number of multipliers required by these designs for large values of M becomes too high for practical implementation. Another disadvantage of these equiripple designs is that the level of the noise lying in the stopband is larger than that of the noise lying in the baseband. In this case, the contribution of the stopband noise to the overall noise obtained after decimation is dominating.

V. FILTER ARCHITECTURE

The above filter structures will lead directly to a very efficient VLSI implementation. The filter parts $F_1(z)$ and $F_2(z)$, given by Eqns. (4c) and (4b), respectively, have been implemented for programmable decimation ratios 32–128 using combined bit-parallel and bit-serial architectures for maximum layout compactness and speed. The first sections, feeded directly by a one-bit stream from the modulator, have been implemented parallelly because of speed limitations. After decimating by a factor of K, the bit-rate is reduced and bit-serial structures can be utilized. Bit-serial and bit-parallel sections have been synchronized such that K parallel sums take as long as 16 serial cycles. The total area for implementing $F_1(z)$ and $F_2(z)$ is 2.2 mm^2 using 2.5 micron CMOS technology. The maximum bit-rate can be as high as 20 MHz for the current design.

The remaining fixed filter sections can be implemented either using the first design with some general multipliers or the second multiplier-free design. For the first design, a multiplier-based architecture is used. It is bit serial and follows the structure proposed in [14]. The estimated area is 4.7 mm^2 for 12-bit coefficients. An optimal implementation for the multiplier-free solution is a dedicated minimal core processor because of the modularity of the design. This will result in a similar silicon area. The overall sigma-delta A/D converter with 16-bit dynamic range for modem applications can be integrated with 2.5 micron CMOS technology in area less than 8 mm^2.

VI. CONCLUSION

An efficient linear-phase FIR filter structure has been proposed for eliminating the out-of-band noise generated by a sigma-delta analog-to-digital converter. The main advantages of the proposed filter structure are:

1. It can be easily implemented in CMOS VLSI.
2. The quantization noise generated by the sigma-delta modulator is effectively attenuated.
3. The output sampling rate is the minimum possible one, unlike for most other proposed designs.
4. Input signal components, such as possible sinusoidal components, lying in the frequency band $[(5/4)(f_s/2), M(f_s/2)]$ are highly attenuated, thus relaxing the anti-aliasing prefilter requirements.
5. The same structure can be used for many oversampling ratios.
6. The area for implementing the overall A/D converter is less than 8 mm^2 with 2.5 micron CMOS technology.

ACKNOWLEDGEMENT

This work has been supported by the National Microelectronics Program of Finland. The authors also thank the Median-Free Group International for excellent working atmosphere and fruitful discussions during the course of this work.

REFERENCES

[1] M. W. Hauser, P. J. Hurst, and R. W. Brodersen, "MOS ADC filter combination that does not require precision analog components," in *Proc. IEEE Int. Solid-State Circuit Conf.*, pp. 80–82, Feb. 1985.

[2] M. W. Hauser and R. W. Brodersen, "Circuit and technology considerations for MOS delta-sigma A/D converters, in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 1310–1315, May 1986.

[3] R. M. Gray, "Oversampled sigma-delta modulation," *IEEE Trans. Comm.*, vol. COM-35, pp. 481–489, May 1987.

[4] S. H. Ardalan, J. J. Paulos, "An analysis of nonlinear behavior in sigma-delta modulators," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 593–603, June 1987.

[5] J. C. Candy, "Decimation for sigma delta modulation," *IEEE Trans. Comm.*, vol. COM-34, pp. 72–76, Jan. 1986.

[6] T. Saramäki, "Efficient recursive digital filters for sampling rate conversion," in *Proc. IEEE Int. Symp. Circuits Syst.* (Newport Beach, CA), pp. 1322-1326, May 1983.

[7] S. Chu and C. S. Burrus, "Multirate filter design using comb filters," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 913–924, Nov. 1984.

[8] T. Saramäki, Y. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst.*, pp. 70–88, vol CAS-35, Jan. 1988.

[9] T. Saramäki, "A class of linear-phase FIR filters for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1023–1036, Oct. 1984.

[10] T. Saramäki, "Design of optimal multistage IIR and FIR filters for sampling rate alteration", in *Proc. IEEE Int. Symp. Circuits Syst.* (San Jose, CA), pp. 227–230, May 1986.

[11] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[12] T. Saramäki, "Design of FIR filters as a tapped cascaded interconnection of identical subfilters," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 1011–1029, Sept. 1987.

[13] T. Ritoniemi, T. Karema, H. Tenhunen, and M. Lindell, "Fully differential CMOS sigma-delta modulator for high performance analog-to-digital conversion with 5 V operating voltage," in *IEEE Int. Conf. Circuits Systs.*, 1988, this conference.

[14] Y. Matsuya, K. Uchimura, A. Iwata, T. Kobayaski, M. Ishikawa, and T. Yoshitome, "A 16-bit oversampling A-to-D conversion technology using triple-integration noise shaping," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 921-929, Dec. 1987.

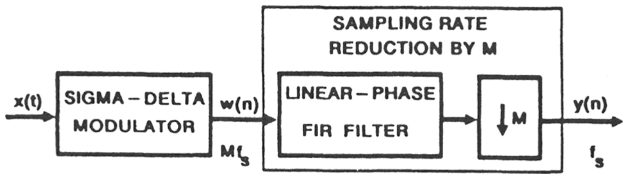
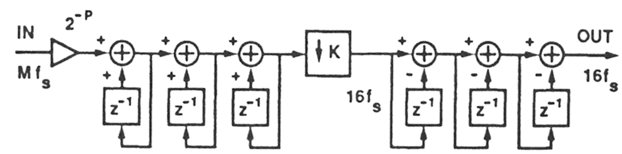
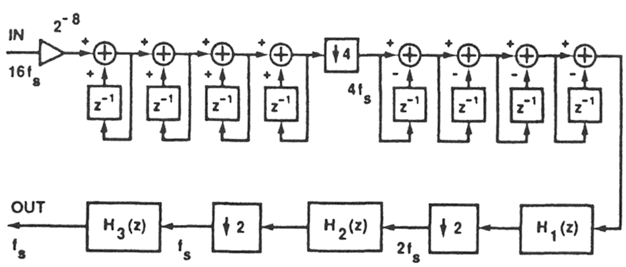


Fig. 1. Block diagram for the A/D converter consisting of an oversampled sigma-delta modulator and a decimator filter.

(a)

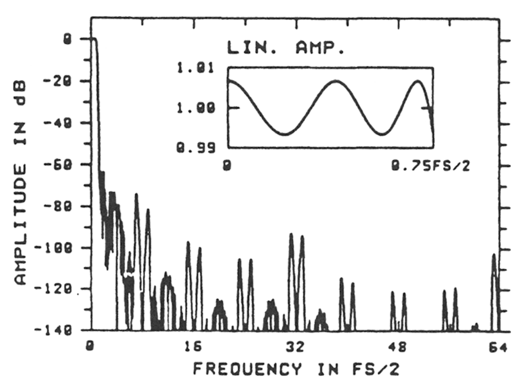


(b)



(c)

Fig. 2. Implementation of the proposed decimator. (a) Overall filter structure. (b) Adjustable filter part. (c) Fixed filter part.



(a)

Fig. 3. Amplitude responses for the proposed decimators for $M = 64$. (a) Multiplier-based design.

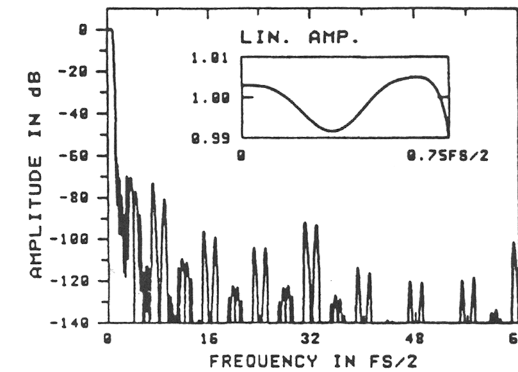


Fig. 3. (Continued.) (b) Multiplier-free design.

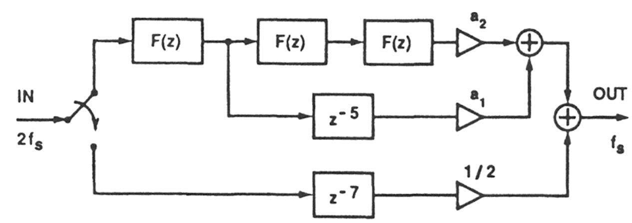


Fig. 4. Efficient implementation of the half-band filter $H_2(z)$. $a_1 = 2^0 - 2^{-2}$ and $a_2 = -2^{-2} + 2^{-9}$.

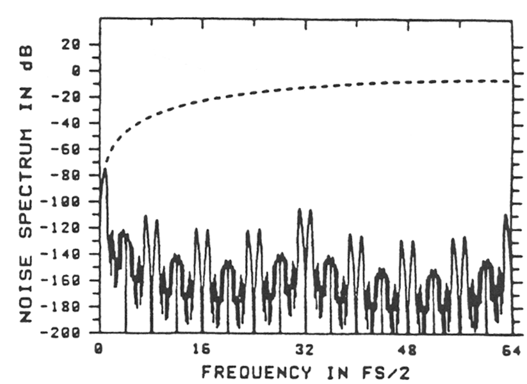


Fig. 5. Noise spectra before (dashed line) and after (solid line) filtering. These are the spectra before decimation. 0 dB corresponds to the noise level obtained when using direct rounding of data at the final output sampling rate of f_s.

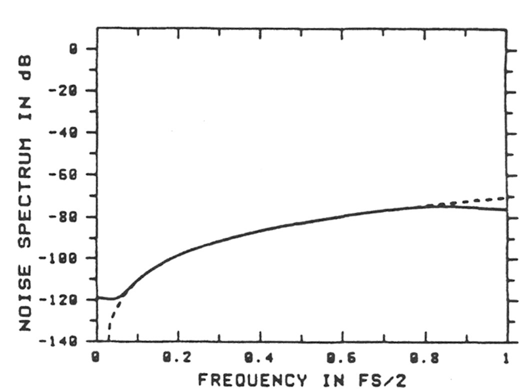


Fig. 6. Noise spectra in the baseband $[0, f_s/2]$. The solid line gives the spectrum at the output of the overall system after filtering and decimation and the dashed line gives the spectrum at the output of the modulator.

Part 5
Theory and Implementations of Oversampling D/A Converters

Double Interpolation for Digital-to-Analog Conversion

JAMES C. CANDY, FELLOW, IEEE, AND AN-NI HUYNH, STUDENT MEMBER, IEEE

Abstract—**Interpolative digital-to-analog converters generate an output that has only a few analog levels. They provide fine resolution by oscillating rapidly between these levels in such a manner that the average output represents the value of the applied code. Here we describe an improved method of interpolating that results in reduced noise in the signal band. A theory of the interpolation, confirmed by experiments, demonstrates that switching between only two levels at 1.3 mHz could provide 16 bit resolution for telephone signals.**

I. Introduction

ORDINARY digital-to-analog converters (D/A's) provide a discrete output level for every value of the digital word that is applied to their input. There is difficulty in implementing these converters for long digital words because of the need to generate a large number of distinct output levels. A method [1] for circumventing the difficulty calls for spanning the signal range with a few widely spaced levels and interpolating values between them. The interpolating mechanism causes the output to oscillate rapidly between the levels, in such a manner that the average output represents the value of the input code. This technique provides a useful tradeoff between the complexity of the analog circuits and the speed at which they operate.

Essential to the technique is an interpolating circuit for truncating the input words to shorter output words. These shorter words change their value at high speed in such a manner that the truncation noise that lies in the bandwidth of the signal is satisfactorily small. The present work describes an improved interpolating circuit that permits significant reduction in the rate at which outputs oscillate, or reduction in the number of levels needed for the output. The technique is suitable for use with pulse code modulated signals that are sampled regularly and quantized uniformly.

The following sections describe and analyze methods of interpolation, while Section V describes the results of measurements on a circuit model.

II. First-Order Interpolation

Fig. 1 shows a circuit of a basic interpolating D/A. The input word, held in register R_0, feeds one port of a binary adder, the output of which separates into two paths. The more significant component y_1 feeds to the output, while the less significant component e_1 feeds back to the second port of the adder via register R_1. The least significant bits accumulate until they overflow into the most significant bits and, thus, contribute to the output. The input register loads at the incoming word rate $2f_0$, while register R_1 and the output D/A operate k times faster at $2kf_0$. We represent the period of the faster clock by τ where

$$2kf_0\tau = 1. \tag{1}$$

Paper approved by the Editor for Signal Processing and Communication Electronics of the IEEE Communications Society. Manuscript received April 3, 1985.

The authors are with AT&T Bell Laboratories, Holmdel, NJ 07733.

IEEE Log Number 8406424.

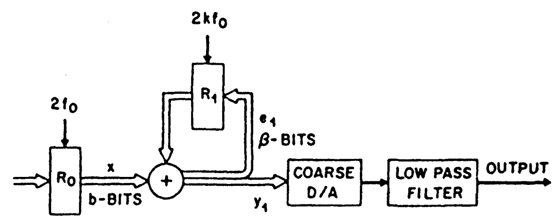


Fig. 1. An outline of a single interpolative D/A.

The output from this circuit, expressed as its z-transform, is

$$Y_1(z) = X(z) - (1 - z^{-1})E_1(z) \tag{2}$$

where

$$z = \exp\ (j\omega\tau).$$

Y_1 represents the input contaminated by a truncation noise E_1 which is filtered by the high-pass function $(1 - z^{-1})$.

The interpolating converter is implemented most easily, and its action is easily explained when the digital signals are expressed in displaced binary notation rather than in two's-complement or sign-magnitude. We use this notation in all the following discussion. Let the input word comprise b bits and let the error e_1 comprise the β least significant bits of the sum. Then y_1 comprises the $(b - \beta + 1)$ most significant bits, the extra bit being the carry from the top of the adder. Input codes can assume integer values from 0 through $(2^b - 1)$, the error integer values from 0 through $(2^\beta - 1)$, while the output assumes integer multiples of 2^β in the range 0 through 2^b. The number of levels needed to represent the output is only

$$l_1 = (2^{b-\beta} + 1) \tag{3}$$

but the switching between values must occur fast enough to suppress the truncation noise that enters the signal band. We will now calculate the frequency ratio k that is needed by this circuit in order to obtain resolutions comparable with b-bit PCM.

The truncation error e comprises a constant term $0.5\ (2^\beta - 1)$ and a noise that can fluctuate with uniform probability in the range $\pm 0.5\ (2^\beta - 1)$, its rms value being $(2^\beta - 1)/\sqrt{12}$. If we now assume that the signals applied to the converter are sufficiently busy to make this noise random with white rms spectral density

$$E_1(f) = \frac{(2^\beta - 1)}{\sqrt{12kf_0}}, \tag{4}$$

then the spectral density of the noise in the output is given by

$$N_1(f) = |E_1(z)(1 - z^{-1})| = \frac{(2^\beta - 1)}{\sqrt{3kf_0}} \left|\sin\left(\frac{\pi f}{2kf_0}\right)\right|. \tag{5}$$

Notice that the dc offset filters away. The net noise in the

Reprinted from *IEEE Trans. Commun.*, vol. COM-34, pp. 77–81, January 1986.

signal band $0 < f < f_0$ can be expressed as

$$N_{10} = (2^\beta - 1)\left(\frac{1 - \text{sinc}\left(\frac{1}{k}\right)}{6k}\right)^{1/2}$$

and

$$N_{10} \approx \frac{\pi(2^\beta - 1)}{6k\sqrt{k}} \qquad \text{when } k^2 \gg 0.5. \tag{6}$$

We now compare this noise to the quantization noise that is inherent in the input, its rms value being $1/\sqrt{12}$. In order for the interpolation noise (6) to be smaller, it is required that

$$k^3 > \frac{1}{3}\pi^2(2^\beta - 1)^2. \tag{7}$$

For example, when $b = 16$ and $\beta = 12$, k should exceed 381. This requires an interpolation rate in excess of 3 MHz and 17 levels of output signal for 4 kHz voiceband signals [2]. The case where the output has only two levels is particularly important for practical implementation. For this converter to have 16 bit resolution requires that $\beta = 16$ and k exceed 2418, which corresponds with a 19 MHz interpolation rate for voiceband signals. Such high rates are a handicap that we can avoid by improving the filtering of the truncation noise. One method replaces register R_1 with more complex digital processing [3], but a better method uses the multiple interpolation described in the next section.

III. Second-Order Interpolation

Fig. 2 shows a converter that uses two accumulations to reduce the amount of truncation noise that enters the signal band. Its output may be expressed in the form

$$Y(z) = Y_1(z) + (1 - z^{-1})Y_2(z) \tag{8}$$

$$= X(z) - (1 - z^{-1})^2 E_2(z). \tag{9}$$

When the error e_2 is random, the spectral density of the noise present in the output is given by

$$N_2(f) = |(1 - z^{-1})^2 E_2(f)| = \frac{(2^\beta - 1)}{3kf_0}\left(1 - \cos\left(\frac{\pi f}{kf_0}\right)\right) \tag{10}$$

and the net noise in the signal band is

$$N_{20} \approx \frac{\pi^2(2^\beta - 1)}{2k^2\sqrt{15k}}; \qquad k^2 > 1. \tag{11}$$

The number of levels needed in the output is

$$l_2 = (2^{b-\beta} + 3). \tag{12}$$

In order for the noise (11) to be less than the noise in b-bit PCM, it is required that

$$k^5 > \frac{\pi^4(2^\beta - 1)^2}{5}. \tag{13}$$

For example with $b = 16$ and $\beta = 12$, k should exceed 51; this corresponds to an interpolation rate of only 404 kHz and 19 level outputs for voiceband signals. When $\beta = b$, four-level output interpolating in excess of 1.25 MHz would provide the resolution of 16 bit PCM. The next section presents a simplified implementation of this converter.

IV. A Double Interpolating D/A Converter

The output (8) comprises two components; y_1 carries the signal contaminated with noise (2) while y_2 provides a first-order compensation for the noise. Converting these two signals into analog form by separate means significantly improves the tolerance of the circuit to inaccuracies. Two-level conversion of y_1 avoids signal distortion caused by misplaced levels. Likewise, two-level conversion for y_2 is desirable, but this requires analog differentiation to replace the digital differentiation in register R_3 and subtractor S_3.

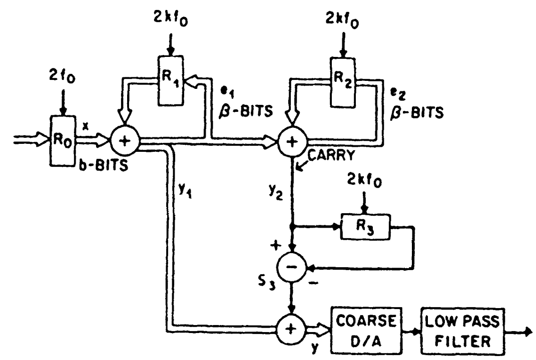


Fig. 2. An outline of a double interpolative D/A.

The approximation of digital differentiation by analog differentiation is satisfactory in circuits such as this, where the word rate far exceeds the signal frequency, because

$$(1 - z^{-1}) \equiv 2j \exp\left(-\frac{j\omega\tau}{2}\right)\sin\left(\omega\frac{\tau}{2}\right)$$

and

$$(1 - z^{-1}) \approx j\omega\tau \exp\left(\frac{-j\omega\tau}{2}\right) \qquad \text{when } (\omega\tau)^2 \ll 24 \tag{14}$$

or

$$(1 - z^{-1}) \approx j\omega\tau \qquad \text{when } \omega\tau \ll 2. \tag{15}$$

The circuit in Fig. 3 uses approximation (14): register R_3 provides a half period delay and C differentiates y_2. The net output of this circuit may be expressed as

$$Y = G\left(Y_1 + j\omega RC\exp\left(-\frac{j\omega\tau}{2}\right)Y_2\right) \tag{16}$$

and this can be equivalent to (8) in the signal band, provided that (14) is valid and that

$$RC \approx \tau = \frac{1}{2kf_0}. \tag{17}$$

The net gain of the analog circuit to the signal is

$$G = \frac{r_1}{(R + r)}\left(1 + \frac{j\omega rRC}{r + R}\right). \tag{18}$$

It cuts off at a frequency that is $k(R + r)/\pi r$ higher than signal frequencies. The purpose of this low-pass filtering, introduced by the presence of resistor r, is to stop high-frequency components of the binary signal y_2 from hitting the amplifier.

Analysis in the Appendix shows that approximation (14) is good for this application, and that relationship (17) must be satisfied to one part in k. It also demonstrates that the least significant 3/5 $(\beta - 1)$ bits of the signal that feeds from the first to the second accumulator may be truncated. Measurements on a circuit model described in the next section confirm these results. They also show a close resemblance between properties of these interpolating converters and those of sigma delta modulators.

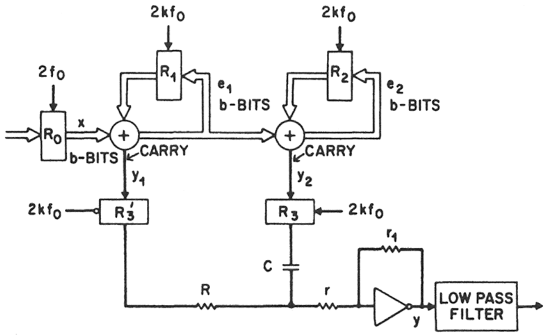


Fig. 3. A double interpolative D/A using two-level conversion to analog.

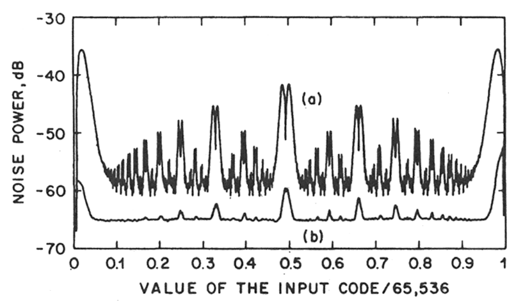


Fig. 4. The dependence of interpolation noise on the value of the applied digital word. Noise is expressed in decibels with respect to $(2^{\beta} - 1)$. The interpolation rate $2kf_0$ = 128 kHz. (a) Single interpolation, calculated mean -45 dB. (b) Double interpolation, calculated mean -64 dB.

V. Measurements on a Circuit Model

The circuits to be described here use relatively low switching rates in order that the interpolation noise be easily distinguishable from spurious circuit imperfection and from the quantization of the input signal. The input comprised 16 bit words generated by a computer at 8 kHz. It represented dc levels and 870 Hz sinewaves of various amplitudes. The circuits employed two-level D/A's and the low-pass filter at their output approximated C-message weighting; its cutoff frequency was about 3 kHz.

Fig. 4 shows graphs of the noise at the output of the converter, plotted in decibels against the value of the binary code as it swept slowly through the entire range 0–65 535. Curve (a) is for single interpolation, y_2 disconnected, and curve (b) is for double interpolation. We see that double interpolation lowers and decorrelates the noise in much the same way as it does in sigma delta modulation [4]. The theory developed in [5] applies to curve (a).

Fig. 5 is a graph of signal-to-noise ratio plotted against the amplitude of the input sinewave. Curve (a) is for single accumulation with the sinewave biased at code 32 768, the center of the range. Curve (b) is for the sinewave biased at 31 744, 1/64 of the range from center. Curve (c) is for double integration with the bias at center. Dashed lines show the result derived from (6) and (11).

Fig. 6 shows the signal-to-noise ratio plotted against the amplitude of the input sinewave for double integration at various interpolation rates.

Fig. 7 shows the signal-to-noise ratio plotted against the deviation of the time constant RC from the ideal value τ. Result (26) of the Appendix calls for ± 8 percent precision. Finally, Fig. 8 shows how the signal-to-noise ratio depends on the number of bits in the second accumulation; result (22) requires that at least 6 bits be processed.

The results of these measurements agree very well with the calculated noise values, especially at the lower cycling rates. At rates above 256 kHz, additional noise is introduced by the limited switching speeds of the commercial TTL components used in our implementation. We anticipate no difficulty in realizing resolutions corresponding to 16 bit PCM in a single chip implementation of the converter using CMOS technology.

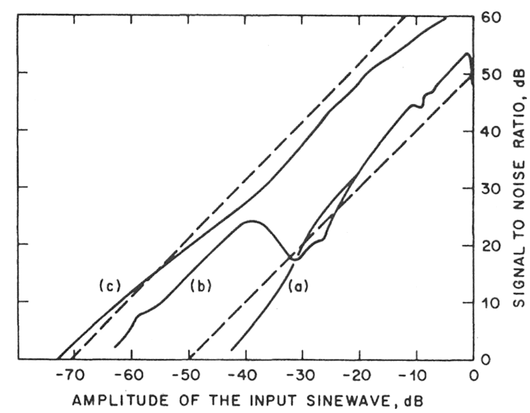


Fig. 5. Signal-to-noise ratio plotted against the amplitude of the applied sinewave. The interpolation rate is 256 kHz. (a) Single interpolation biased to center. (b) Single interpolation biased 1/64 from center. (c) Double interpolation biased to center. Dashed lines show responses calculated for uncorrelated noise.

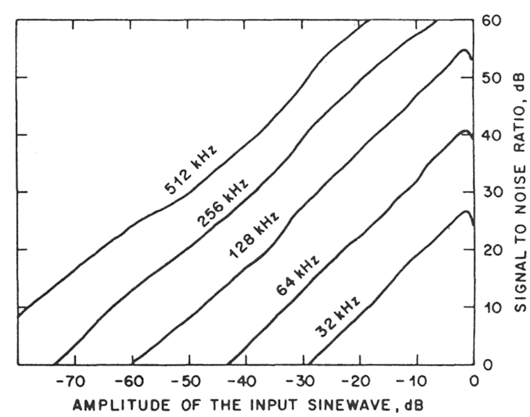


Fig. 6. Signal-to-noise ratio plotted against the amplitude of the applied sinewave for double interpolation at various rates.

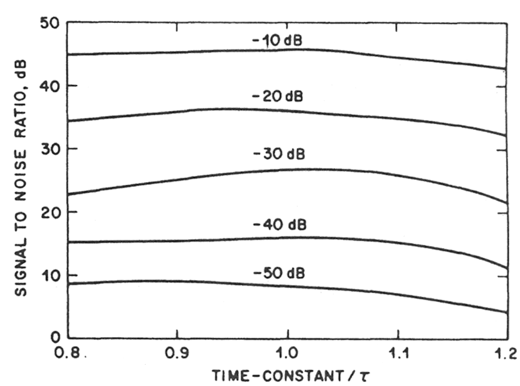


Fig. 7. The variation of signal-to-noise ratio with the time constant RC for various amplitudes of the applied sinewave.

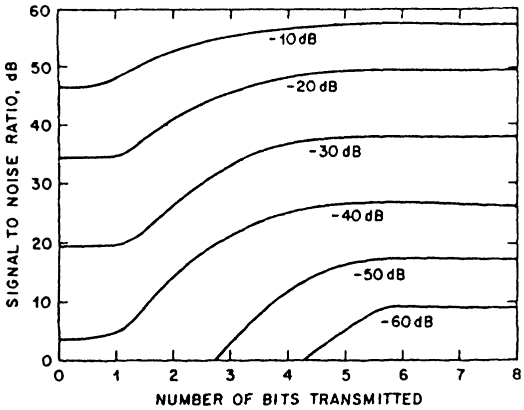


Fig. 8. The dependence of signal-to-noise ratio on the number of bits that are transmitted from the first to the second accumulator for various amplitudes of the applied sinewave.

VI. Concluding Remarks

Use of double accumulation greatly enhances the performance of these interpolating converters, in much the same manner as it improves sigma delta modulation. The technique can clearly be extended to more than two accumulations. Limit cycle oscillations [4] that spoil the performance of high-order sigma delta modulations cannot occur in these circuits because their accumulators are not included in feedback loops. Notice that the accumulations used in these circuits are not cleared at the start of each new word as they are in [1]. Such resetting significantly increases the magnitude of the interpolation noise and its correlation with input code values.

The specification of filters needed to smooth the output signal will depend on the application. Relatively simple filters will usually suffice, because the interpolation noise increases no more than 12 dB per octave and the restriction on out-of-band noise in most applications is usually much less severe than the restriction on in-band noise.

Appendix

Precision of the Circuit Parameters

Fig. 9 shows the main functions of the double interpolating D/A. X represents the input. E_1 and E_2 are the truncation errors at the output of the first and second accumulations, respectively. δ represents a d-bit truncation that may be introduced in the connection between the accumulators. g is a gain factor that may differ from unity because of circuit imperfections.

The output from this circuit may be expressed as

$$Y=X-(1-z^{-1})(1-g)E_1-(1-z^{-1})g\delta-(1-z^{-1})^2gE_2. \tag{19}$$

If the noise from the three sources E_1, E_2, and δ are uncorrelated, then the net noise power in baseband may be derived using results (6) and (11). It is

$$N_0^2=\frac{\pi^2(2^\beta-1)(1-g)^2}{9k^3}+\frac{\pi^2(2^d-1)^2g^2}{9k^3}+\frac{\pi^4(2^\beta-1)^2g^2}{60k^5}. \tag{20}$$

Our design should make this noise less than the quantization noise in the input signal; it provides tradeoffs between required values of k, β, δ, and g. In order to illustrate a possible design, we will assume that k has been chosen and we determine the conditions that cause the noise introduced by g and δ, to be small compared with the inherent interpolation noise (11).

In order that the noise introduced by the truncation δ be

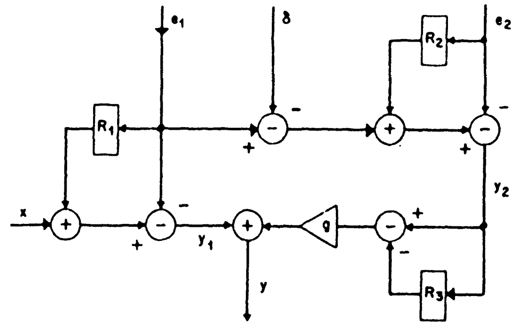


Fig. 9. A model of the double interpolative D/A. e_1, e_2, and δ represent truncation noises. g is a gain factor that ideally is unity.

small, it is required that

$$\frac{\pi^2g^2(2^d-1)^2}{9k^3}<\frac{\pi^4(2^\beta-1)^2}{60k^5} \tag{21}$$

which reduces to

$$\left(\frac{2^d-1}{2^\beta-1}\right)<\frac{\pi}{k}\sqrt{\frac{3}{20}}; \qquad g\approx 1 \tag{22}$$

when the value of k just satisfies (13).

The number of bits that may be truncated is given approximately by

$$d<\frac{3}{5}(\beta-1). \tag{23}$$

A significant reduction in the size of the second accumulation is usually permissible.

The condition that the noise introduced by imprecise gain g be small requires that

$$\frac{\pi^2(2^\beta-1)^2(1-g)^2}{9k^3}<\frac{\pi^4(2^\beta-1)^2}{60k^5} \tag{24}$$

which gives

$$\Delta g=(1-g)<\frac{1.2}{k}. \tag{25}$$

A major reason for the gain g not equaling unity is error in the value of capacitor C in Fig. 4. Result (25) requires that the time constant RC should satisfy

$$\left|\frac{RC}{\tau}-1\right|<\frac{1.2}{k}. \tag{26}$$

Another reason for inaccurate gain is the approximation (14). This approximation introduces a change in gain that is dependent on frequency. It may be expressed as a per-unit error

$$\Delta g=\frac{(\omega\tau)^2}{24}=\frac{\pi^2}{24k^2}; \qquad f=f_0. \tag{27}$$

This satisfies condition (25) especially when k is large. In comparison, approximation (15) entails a per-unit change of gain

$$\Delta g=\frac{\omega\tau}{2}=\frac{\pi}{2k}; \qquad f=f_0 \tag{28}$$

which does not satisfy (23). Approximation (15) can be acceptable, however, if sufficient margin is allowed in satisfying (13).

REFERENCES

[1] G. R. Ritchie, J. C. Candy, and W. H. Ninke, "Interpolative digital-to-analog converters," *IEEE Trans. Commun.*, vol. COM-22, pp. 1797–1806, Nov. 1974.

[2] J. C. Candy, B. A. Wooley, and O. J. Benjamin, "A voiceband codec with digital filtering," *IEEE Trans. Commun.*, vol. COM-29, pp. 815–830, June 1981.

[3] H. G. Musmann and W. W. Korte, "Generalized interpolative method for digital/analog conversion of PCM signals," US Patent 4 467 316, Aug. 1984.

[4] J. C. Candy, "A use of double integration in sigma delta modulation," *IEEE Trans. Commun.*, vol. COM-33, pp. 249–258, Mar. 1985.

[5] J. C. Candy and O. J. Benjamin, "The structure of quantization noise from sigma-delta modulation," *IEEE Trans. Commun.*, vol. COM-29, pp. 1316–1323, Sept. 1981.

A 16-BIT 4'TH ORDER NOISE-SHAPING D/A CONVERTER

L. Richard Carley and John Kenney

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh PA 15213

A 16-bit oversampling D/A converter has been designed using a 4'th order all-digital noise-shaping loop followed by a 3-bit D/A converter. The 3-bit D/A converter, which employs a novel form of dynamic element matching, achieves high accuracy and long-term stability without requiring precision matching of components. The harmonic distortion of the untrimmed monolithic CMOS prototype D/A converter is less than -90dB. This multi-bit noise-shaping D/A converter achieves performance comparable to that of a 1-bit noise-shaping D/A that operates at nearly 4 times its' clock rate.

I. Introduction

Delta-Sigma Modulation (DSM) converters have recently achieved popularity for use in integrated circuit data converters, both A/D[1] and D/A[2] converters. Their attractiveness for IC systems, in part, is due to the fact that they employ a 1-bit D/A converter which does not require precision component matching. However, experience has revealed that high integral linearity is very difficult to achieve, perhaps because the 1-bit loop quantizer is frequently overloaded. The all-digital noise-shaping loop presented in this paper avoids quantizer overload by employing a quantizer which truncates the signal to 3 bits rather than 1 bit as in DSM-type converters. Eliminating quantizer overload, in addition to improving integral linearity, also allows higher order (> 2) noise-shaping loops to be employed, since the low frequency oscillations observed in higher order delta-sigma modulation loops[1] are a result of quantizer overload.[7] For these reasons, a multi-bit noise-shaping D/A converter system can achieve a signal-to-noise (SNR) ratio comparable to that of a second order 1-bit noise-shaping (DSM) D/A converter operating at a much higher sampling rate. For example, the prototype system, which operates at 3.2MHz, achieves performance comparable to that of the DSM-type converter which operates at 11.3MHz described by Naus, et. al..[2]

As stated above, many advantages result from using a multi-bit noise-shaping converter instead of a DSM converter. However, there is one major disadvantage; a multi-bit D/A converter typically *requires* precision component matching.[3,4] This paper presents a D/A converter architecture that employs a novel form of "dynamic element matching"[5,6] particularly suited to oversampled data systems, to achieve excellent integral and differential linearity, while requiring only modest component matching. For example, the prototype system has a peak component mismatch of approximately 0.3%, and a measured peak integral linearity error of 0.0022% of full scale. This new topology allows multi-bit noise-shaping coders to employ D/A converters which achieve high integral linearity without requiring component trimming. Therefore, this new D/A converter topology is easily implemented in IC technology and does not require on-chip component trimming.

System Topology

Figure 1 shows the topology of the 16-bit D/A converter system. Interpolation, the process of increasing the sampling rate, is performed in two stages. The input signal is interpolated by a factor of 64 (L in figure 1) in two stages. First, the input is interpolated by a factor of two and filtered by a standard 64-tap FIR filter topology. Then, the FIR filter's output is interpolated by a factor of 32 and filtered using a comb filter.[8] The combined filter attenuates frequencies above 30KHz by more than 80dB. Extensive simulations indicate that the SNR at the output of the noise-shaping loop with this interpolator architecture is less than 3dB below the SNR with an ideal interpolating filter.

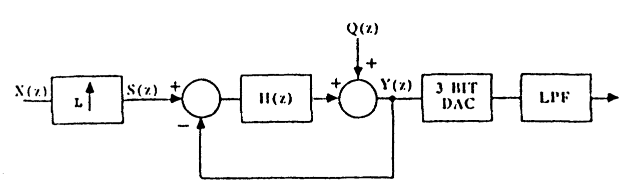


Figure 1 - Topology of the 16-Bit D/A converter system.

Reprinted from *IEEE Proc. CICC*, pp. 21.7.1-21.7.4, 1988.

The noise-shaping loop filter (H(z) in figure 1) employs a 4'th order loop filter with 4 poles at $z=1$ and 3 zeros positioned to maintain system stability and to minimize the "noise gain" of the loop.[9,10] The worst-case noise gain is determined by assuming that the quantization error will take on a sequence of worst case values, +1/2 LSB or −1/2 LSB.[12] The noise gain can be computed by summing the magnitude of the impulse response of the transfer function from the quantizer's output back to its' input. Noise gain is a constraint used in choosing the location of the poles of the loop transmission of the noise-shaping loop. A maximum noise gain of 4 was chosen as the constraint when placing the closed-loop poles for the noise-shaping loop of the prototype system. Therefore, the quantization error at the noise-shaping loop's output, which is in the range +1/2 LSB to −1/2 LSB, is amplified in the worst case by a factor of 4. Hence, the noise signal at the quantizer's input will be in the range +2 LSB to −2 LSB. In order to prevent the noise-shaping quantizer from being overloaded, the 16-bit input signal is scaled so that its' full-scale range is between +2 LSB and −2 LSB.

3-Bit D/A Converter Topology

The 3-bit D/A converter topology achieves high integral linearity without requiring precisely matched components through the application of a modified form of dynamic element matching. The D/A converter's n'th output level is generated by charging all capacitors to a +5 volt reference level and then switching n of them into the summing junction of an operational amplifier during each clock cycle. Note, the capacitors which are not switched into the summing junction are switched into ground in order to maintain a constant current in the +5 volt reference line. A resistor version of the 3-bit D/A converter was also fabricated. As with the capacitor version, the n'th output level was generated by switching n resistors into the summing junction of an operation amplifier. The other end of the resistors was tied to a reference voltage.

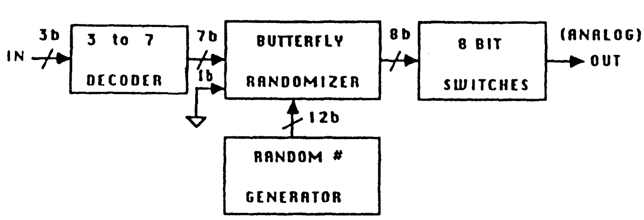


Figure 2 – Topology of the 3-Bit D/A converter.

Dynamic element matching is implemented by choosing different elements to represent the n'th level as a function of time. The "randomizer" selects each element on n out of 8 clock cycles (see figure 2). The goal of this approach is to decorrelate the error on successive samples. Idealy, a mismatch between the capacitors is converted into a white noise signal; and, since the D/A converter is being used in an oversampling application, a large portion of the error power is filtered out. One simple approach to decorrelating the error on successive samples would be to have a barrel shifter which rotated one increment after each clock. This approach would completely decorrelate successive output errors only if the mismatch between capacitors were independent of the capacitor's position on the die. Unfortunately, just the opposite is true. Adjacent capacitors are much more likely to match than distant capacitors due to gradients in oxide thickness across the wafer. An ideal randomizer, one which connects each of the 8 inputs to all 8 possible outputs in a random fashion, would have to include 40,320 possible permutations. In order to conserve die area, a selected set of random combinations was allowed. The randomizer circuit consists of a series of 3 "butterfly" networks coupling the inputs to the output (see figure 3). This randomizer implements 4096 different combinations. A pseudo-random sequence generator is used to generate the random 12-bit control sequences for the butterfly switches. Simulations verified that this partial randomizer resulted in good decorrelation of successive converter errors in the face of both linear and quadratic gradients in oxide thickness across the wafer. Figure 4 shows a photomicrograph of the 3-bit D/A converter, the digital randomizer circuitry, and the pseudo-random sequence generator.

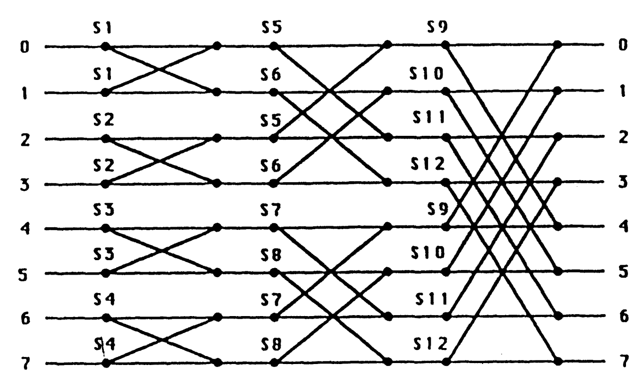


Figure 3 – Topology of the "Butterfly" type randomizer.

For a DC input code of n, each capacitor is discharged into the output on average n out of every 8 clock cycles. Therefore, the D/A converter acts as a duty-cycle modulator and the integral linearity is limited only by the product of the fractional element mismatch ($\Delta E/E$) and the fractional clock jitter ($\Delta T/T$).[5,6] Extremely high DC integral linearity can be achieved, even if the elements match very poorly, as long as a

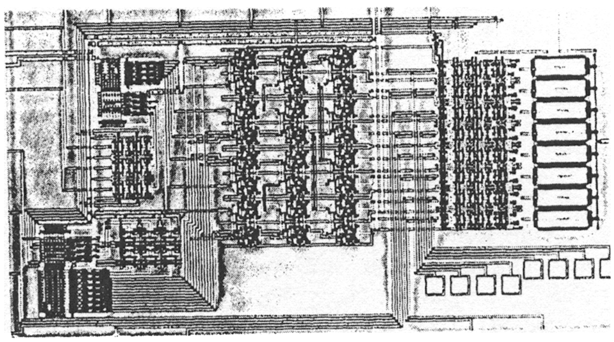


Figure 4 – Photomicrograph of randomizer and 3–Bit D/A converter.

precise clock signal is used. However, the element mismatch will appear as an AC noise signal added to the D/A converter's output.

The maximum noise signal, as a fraction of the 3–bit D/A converter's full scale (FS), is $\frac{RMS[\Delta E/E]}{\sqrt{2M}}$. In an oversampling converter, only noise power in the signal passband is important. Assuming that the randomizer's pattern is not correlated with the element variations, the noise signal will be white and the RMS fraction of FS for the noise signal in the passband will be $\frac{RMS[\Delta E/E]}{\sqrt{2M \times OV}}$ where OV is the oversampling ratio. For example, if a 3–bit D/A converter was constructed using capacitors with an RMS variation of 0.1% and the system oversamples by a factor of 64, the in–band RMS noise signal is approximately 0.003% of FS.

The analog output from the 3–bit D/A converter (see figure 1) was filtered by a fifth–order Butterworth filter. The real axis pole was provided by the first op amp and the two complex pole pairs were implemented as a cascade of two Sallen–and–Key second order sections. The SNR and the open–loop gain of the operational amplifiers limit the system's SNR. The analog filter was implemented off–chip using ultra–low noise operational amplifiers. Although the prototype CMOS IC did not include the analog filter, low–noise monolithic CMOS operational amplifiers with performance suitable for this task have been designed and fabricated.[2,11]

Results

Figure 5 shows the SNR at the analog filter's output as a function of the input amplitude for a 1 KHz sinusoidal input. The system achieves a total dynamic range greater than 92dB.

Figure 6 shows the power spectral density (PSD) of the 3–bit D/A converter's output without the analog filter for a 0dB 1KHz sinusoidal input. The flattening of the PSD of the noise at

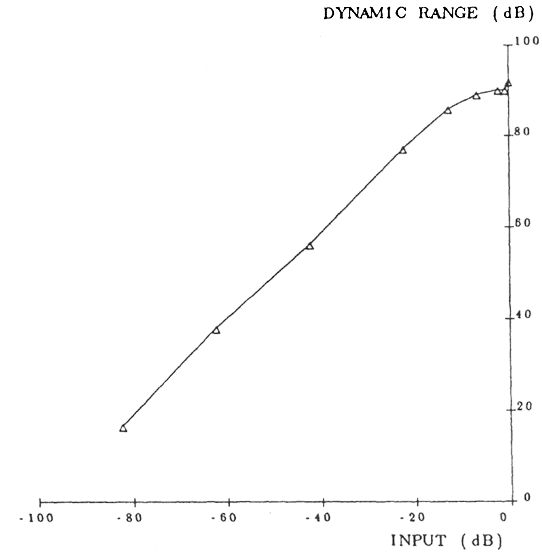


Figure 5 – Plot of SNR as a function of input amplitude. In this case distortion is *included* with the noise. Input signal is a 1KHz sinewave.

approximately 1 MHz is a result of the poles incorporated into the noise–shaping loop's transfer function to control the noise gain. Note, this signal contains noise components from two sources: quantization noise from the noise–shaping loop and the approximately "white" noise resulting from the randomized capacitor mismatch.

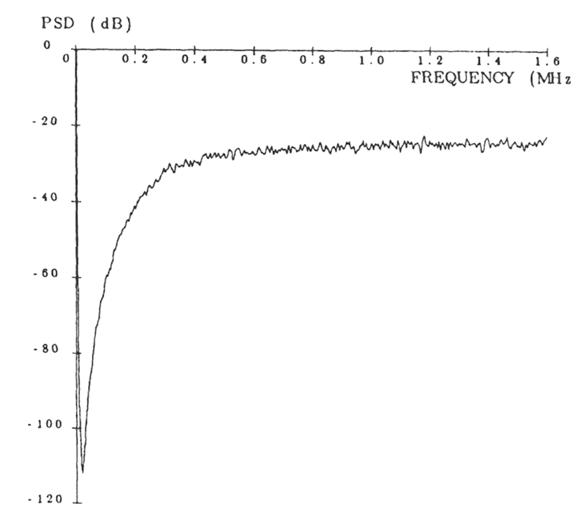


Figure 6 – Power Spectral Density of the signal at D/A converter output with a 0dB 1KHz input signal.

Figure 7 shows the power spectral density at the output of the analog filter. Simulations with an ideal 3–bit D/A converter indicate that the PSD of the in–band noise is dominated, in the audio frequency range, by the component mismatch noise rather than quantization noise from the noise–shaping loop, The harmonic distortion components are each more than 96dB below the maximum input level.

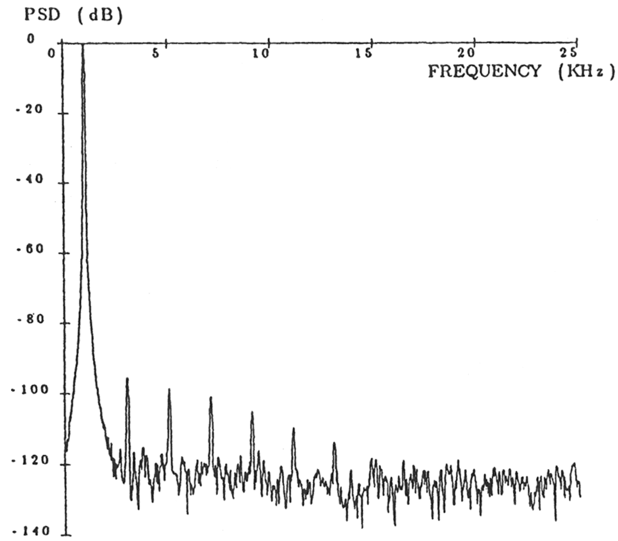


Figure 7 – Power Spectral Density of the signal at output of the analog filter with a 0dB 1KHZ input signal. Note, the amplitude of the input signal has been artificially reduced by passing it though a 1 KHz notch filter in order to decrease the dynamic range to allow accurate measurement of the PSD.

Conclusions

A 16-bit oversampling D/A converter has been designed using an all-digital noise-shaping loop followed by a 3-bit D/A converter that employs a modified form of dynamic element matching to achieve high accuracy without the need for precision matching of components. Using a version of the 3-bit D/A converter which was fabricated in 3μm CMOS, the 16-bit D/A conversion system achieved a dynamic range of 90dB and a THD < −94dB at a sampling rate of only 3.2MHZ. It is the topology of the 3-bit D/A converter which enables us to design a noise-shaping loop which can attain performance equivalent to that of DSM converters operating at much higher clock rates, without incurring the penalty of high integral nonlinearity.

Although all of the system was not fabricated in IC form, other researchers have fabricated digital interpolating filters, digital noise-shaping loops, and analog output filters equivalent in performance to those required in this system. For example, a higher order (1:256) interpolating filter and a 2'nd order noise-shaping loop were implemented by Naus et. al.[2] in approximately 18mm^2. Therefore, we conclude that by testing the novel portion of this system in IC form we have proved that integration of the entire system is viable.

Acknowledgements

This work was supported in part by a grant from the Analog Devices Corporation and by the NSF under grant ENG-8451496.

References

[1] J. C. Candy, "A Use of Double Integration in Sigma-Delta Modulation", *IEEE Trans. Commun.*, vol. COM-33, pp. 249-258, March 1985.

[2] P. J. A. Naus, E. C. Dijkmans, E. F. Stikvoort, A. J. McKnight, D. J. Holland, and W. Brandinal, "A CMOS Stereo 16-bit D/A Converter for Digital Audio", *JSSC* Vol. SC-22, No. 3, pp. 390-395, June 1987.

[3] J. W. Scott. W. Lee, C. Giancarlo, and C. G. Sodini, "A CMOS Slope Adaptive Delta Modulator", Proceedings of the 1986 ISSCC, pp. 130-131, Feb. 1986.

[4] R. W. Adams, "Design and Implementation of an Audio 18-Bit A/D Converter using Oversampling Techniques." *J. Audio Eng. Soc.* Vol. 34, No. 3, pp. 153-166, March 1986.

[5] R. J. Van De Plassche, "A monolithic 14-bit D/A converter", *JSSC* Vol. SC-14, No. 3, pp. 552-556, June 1979.

[6] K. B. Klaassen, "Digitally controlled absolute voltage division", *IEEE Trans. on Instrumentation and Measurement* Vol. 24, No. 3, pp. 106-112, June 1975.

[7] C. Wolff and L. R. Carley, "Modeling the Quantizer in Higher-Order Delta-Sigma Modulators", *International Symposium on Circuits and Systems*, Helsinki Finland, June 1988.

[8] S. Chu and C. S. Burrus, "Multirate Filter Designs using Comb Filters", *IEEE Trans. on Circuits and Systems*, vol. CAS-31, pp. 913-924, Nov. 1984.

[9] J. Kenney, "Design Methodology for N-th Order Noise-Shaping Coders", Masters Project Report, Department of ECE, CMU, Pittsburgh PA, Jan. 1988.

[10] B. P. Agrawal and K. Shenoi, "Design Methodology for ΣΔM", *IEEE Trans. Commun.* COM-31:360-369, March 1983.

[11] Y. Matsuya, K. Uchimura, A. Iwata, T. Kobayashi, M. Ishikawa, and T. Yoshitome, "A 16-bit Oversampling A-to-D Conversion Technology Using Triple-Integration Noise Shaping", *JSSC* Vol. SC-22, No. 6, pp. 921-929, December 1987.

[12] L. R. Carley, "An Oversampling Analog-to-Digital Converter Topology for High Resolution Signal Acquisition Systems", *IEEE Trans. Circuits and Systems* CAS-34, #1, 83-91, January 1987.

A CMOS Stereo 16-bit D/A Converter for Digital Audio

PETER J. A. NAUS, EISE CAREL DIJKMANS, EDUARD F. STIKVOORT, ANDREW J. McKNIGHT, DAVID J. HOLLAND, AND WERNER BRADINAL

Abstract —**A complete monolithic stereo 16-bit D/A converter primarily intended for use in compact-disc players and digital audio tape recorders is described in this paper. The D/A converter achieves 16-bit resolution by using a code-conversion technique based upon oversampling and noise shaping. The band-limiting filters required for waveform smoothing and out-of-band noise reduction are included. Owing to the oversampling principle most applications will require only a few components for an analog post-filter. The converter has a linear characteristic and linear phase response. The chip is processed in a 2-μm CMOS process and the die size is 44 mm². Only a single 5-V supply is needed.**

I. INTRODUCTION

IN CONVENTIONAL linear PCM the quantizing noise is assumed to be white noise having a power $q^2/12$, where q is the quantizing step [1], [2]. If the signal bandwidth is less than half the sample frequency f_s, the noise spectrum can be reshaped in order to decrease the in-band noise [3]. For equal in-band noise, this noise shaping results in a lower number of bits per sample, together with a higher sample rate than is used in conventional PCM. In the system presented here (see Fig. 1), a performance, equivalent to 16-bit D/A conversion at a sample rate of 44.1 kHz, has been obtained by using a 1-bit D/A converter at a sample rate of 11.2896 MHz ($256 f_s$).

In the system, the stereo 16-bit PCM 44.1-kHz input signal is filtered and upsampled to $4 f_s$ in the first oversampling section. After the first oversampling section the signal is demultiplexed and from this point on both channels (left and right) are treated separately. The sample rate is increased to $256 f_s$ in the second oversampling section. The word length is reduced to 1 bit, by a noise-shaping code conversion, and the 1-bit code passes through a 1-bit D/A converter. A small analog post-filter completes the D/A conversion of the original 16-bit PCM signal.

Manuscript received October 6, 1986; revised December 19, 1986.
P. J. A. Naus, E. C. Dijkmans, and E. F. Stikvoort are with Philips Research Laboratory, Eindhoven 5600 JA, The Netherlands.
A. J. McKnight and D. J. Holland are with Mullard, Southampton, England.
W. Bradinal is with Valvo, Hamburg, Germany.
IEEE Log Number 8714201.

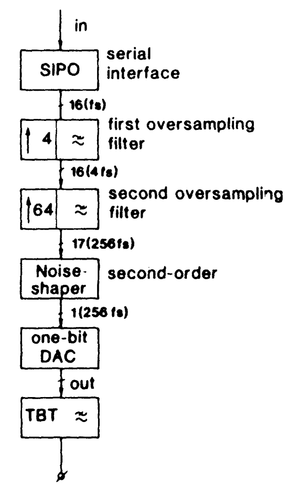


Fig. 1. Block diagram of one channel of the D/A converter.

II. OVERSAMPLING AND FILTERING

The 44.1-kHz sample rate of the PCM input signal is increased by a factor of 4 in the first oversampling section. It contains a finite-impulse-response (FIR) low-pass filter, which has a 20-kHz bandwidth, a passband ripple of ± 0.02 dB, and a stopband rejection of 60 dB for frequencies above 24 kHz. Its hardware contains a ROM, a RAM, and an array multiplier. Although the upsampling filter structure is multiplexed between the two channels, there is no phase shift between the two analog outputs.

The second oversampling section is an interpolating filter based on an adder structure. The sample rate is increased by a factor of 64, by using a linear interpolator ($\uparrow 128 f_s$) and a sample-and-hold circuit ($\uparrow 256 f_s$). An internally generated out-of-band dither signal is used to prevent audible idling patterns of the noise shaper at low-input signal levels. The amplitude increases because of the dither signal and therefore an extra bit is needed. The word length at the output of the linear interpolator has to be 17 bits.

All signal processing steps involved in generating the 1-bit code use FIR filters, ensuring a linear phase characteristic. Owing to the digital oversampling filters only a simple analog post-filter is required, which is realized by a

Reprinted from *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 390–395, June 1987.

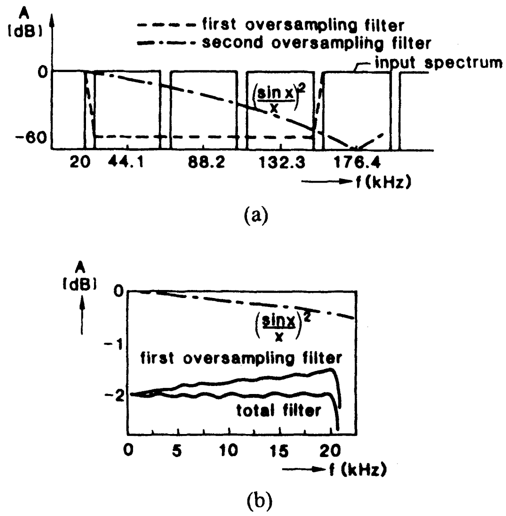


Fig. 2. Characteristics of the filters: (a) stopband, and (b) passband.

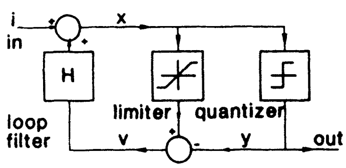


Fig. 3. Block diagram of the noise shaper.

third-order Butterworth filter with a -3-dB cutoff frequency at 60 kHz. The frequency responses of the digital filter and the analog output filter are shown in Fig. 2. Frequencies at multiples of the sample frequency f_s are attenuated by the first oversampling filter. The linear interpolator has a transfer function of $(\sin(x)/x)^2$, in which $x = \pi f/4f_s$, and attenuates the frequencies at multiples of $4f_s$. The first oversampling filter contains compensation for the $(\sin(x)/x)^2$ roll-off due to the interpolator plus the roll-off of the Butterworth filter. The combined filter curve will be flat over the audio band with a ripple of 0.02 dB.

III. Noise Shaper

The 17-bit oversampled PCM signal is supplied to a second-order noise shaper. Fig. 3 shows the block diagram of the noise shaper. The noise shaper is a feedback loop consisting of a quantizer which reduces the word length to 1 bit, a limiter which prevents overflow, and a loop filter which shapes the spectral distribution of the quantization noise generated by the quantizer.

Fig. 4(a) and (b) shows the characteristics of the quantizer and the limiter as a function of the input x, respectively. The operation of the noise shaper can easily be explained, if one crudely models the quantizer by

$$y = cx + r \tag{1}$$

where r is the quantizing error of the quantizer and c is the gain attributed to the quantizer in the loop. Together with

$$X = I + HV$$
$$V = X - Y$$

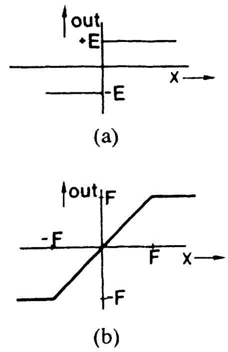


Fig. 4. (a) Output of the quantizer as a function of the input. (b) Output of the limiter as a function of the input.

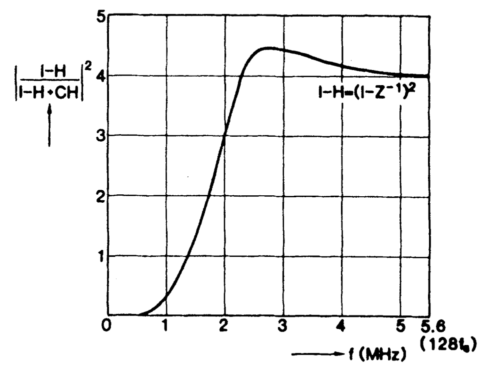


Fig. 5. Noise density at the output of the noise shaper.

it follows that

$$Y = \frac{cI}{1 - H + cH} + \frac{R(1-H)}{1 - H + cH}. \tag{2}$$

I, X, V, and Y are the frequency-domain representations of i, x, v, and y. R is the spectral density of the quantizing error. The spectral components of the quantizing error in the signal band $(0, \Theta_b)$ must be minimized, hence $|1 - H(e^{j\Theta})|$ must be minimized for $\Theta \in [0, \Theta_b]$. If $|1 - H(e^{j\Theta})|$ is small relative to one, (2) may be approximated by

$$Y = I + R(1 - H)/c. \tag{3}$$

The coder output Y contains the undistorted spectrum of the input signal I, added to the spectrally shaped quantizing error.

In our design we have chosen for a rather uncomplicated type of loop filter $H(z) = z^{-1}(2 - z^{-1})$. The resulting noise transfer function, in the signal band, of the 1-bit quantization noise has the shape

$$1 - H(z) = (1 - z^{-1})^2. \tag{4}$$

This noise shaping changes the quantizing noise spectrum (see Fig. 5). In the signal band the noise density increases 12 dB/octave with frequency. The power in the quantizer output in the loop can be computed from the noise density for zero input signal to the coder:

$$\bar{y}^2 = \int_{-\pi}^{\pi} \frac{E^2}{6\Pi} \left| \frac{1 - H(\theta)}{1 - H(\theta) + cH(\theta)} \right|^2 d\theta. \tag{5}$$

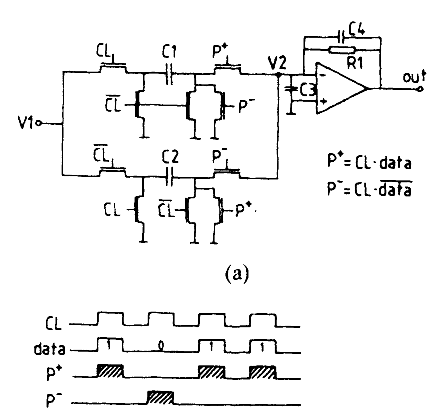
The signal Y only takes the values $\pm E$, hence c follows from

$$\int_{-\pi}^{\pi} \left| \frac{1-H(\theta)}{1+(c-1)H(\theta)} \right|^2 d\theta = 6\Pi. \tag{6}$$

The idle channel in-band noise power N_i is approximated for $\Theta_b \ll 1$ by

$$N_i = \frac{E^2 \Theta_b^5}{15\pi c^2} \tag{7}$$

where E is the amplitude of the output of the quantizer. In our design the value of c is 0.667. For an oversampling factor of 256 and an audio bandwidth of 20 kHz, the in-band noise power $N_i = -107.8$ dB$_n$, where 0 dB$_n$ corresponds to the maximum signal power $E^2/2$. It should be noted that the maximum signal power which the coder can handle is less than $E^2/2$. For increasing input signal level, the quantizer input level increases sharply in the proximity of overload of the coder, the signal-to-noise ratio in the output decreases, and the coder tends towards instability. Hence, in a practical device, a limiter in the loop is necessary for reasons of stability and overflow protection.



(a)

(b)

Fig. 6. One-bit D/A converter: (a) circuit diagram, and (b) timing diagram.

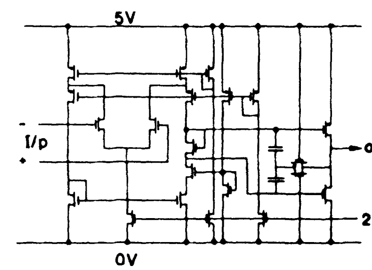


Fig. 7. Circuit diagram of the op amp.

IV. The 1-bit D/A Converter

The D/A converter circuit, as shown in Fig. 6(a), has a very simple structure. Its function is to modulate a dc voltage with the 1-bit data stream. However, as the data contain a wide-band frequency spectrum, the modulator needs a high linearity, otherwise distortion will cause intermodulation with the out-of-band quantization noise, folding it back into the audio band.

The circuit is implemented as a switched-capacitor circuit. The timing diagram is shown in Fig. 6(b). During the first half of the sample period, either C_1 is charged by drawing a unity charge out of the summing node of the op amp or capacitor C_2 is discharged by pushing a unity charge into the op amp. During the second half, C_1 and C_2 are discharged and charged, respectively. The current flowing through the feedback network of the op amp, disregarding the quantization noise, can be approximated as follows:

$$I^+ = \frac{C_1 f_s}{2}(V_1 - V_2)(1 + m\sin(pt) + \cdots)$$

$$I^- = \frac{C_2 f_s}{2}(-V_1 - V_2)(1 - m\sin(pt) + \cdots)$$

$$\begin{aligned} I^+ + I^- = \frac{f_s}{2}\{ & V_1(C_1 - C_2) - V_2(C_1 + C_2) \\ & + mV_1(C_1 + C_2)(\sin(pt) + \cdots) \\ & - mV_2(C_1 - C_2)(\sin(pt) + \cdots)\}. \end{aligned} \tag{8}$$

Clearly, the resulting current contains three components—a dc term, the wanted signal, and a term which contains the signal V_2—that can be described as

$$V_2 = V_o - \frac{V_{\text{out}}}{A} \tag{9}$$

where V_o is the dc offset voltage of the op amp and A is the open-loop gain of the amplifier. From (8) and (9) it follows that distortion products will be generated where the second harmonic distortion will dominate. From

$$\frac{V_{\text{dist}}}{V_{\text{signal}}} = \frac{mR_1 f_s}{4}\left(\frac{C_1 - C_2}{A}\right) \tag{10}$$

it follows that this second-harmonic distortion is very dependent on the gain of the op amp and the matching between C_1 and C_2.

As the input signal is the sum of a low-frequency audio signal and high-frequency noise, any nonlinearity will result not only in harmonic distortion of the audio signal, but also in the folding back of intermodulation products of the high-frequency noise components into the audio band.

The voltage step on the input of the summing node of the op amp generates slew-rate distortion of the op amp. The amplitude of this voltage step is dependent on the ratio of the capacitors C_4 and C_1, C_2 and the ratio of the ON resistance of the switches and the high-frequency output impedance of the op amp. The high-frequency output impedance of the op amp is roughly determined by the $1/g_m$ of the output stage. Capacitor C_3 is added to the

TABLE I
OP-AMP SPECIFICATIONS

DC-Gain	90 dB
Phase margin (0 dB) $C_{load} = 20\,pF$	45 deg. at 45 MHz
Phase margin (20 dB) $C_{load} = 20\,pF$	85 deg. at 4.5 MHz
Output impedance	80 ohm
Power consumption	18 mW
Input offset voltage	2 mV
Slew-rate	30 V/μsec
Size	610 μm x 350 μm

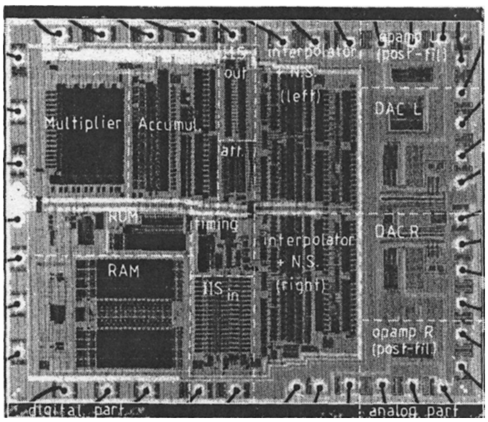


Fig. 8. Photograph of the stereo 16-bit D/A converter.

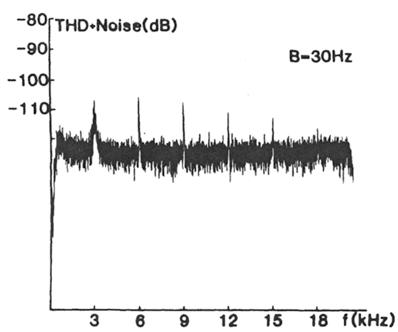


Fig. 9. Measured total harmonic distortion plus noise in the audio band.

summing node of the op amp to decrease the high-frequency components in V_2 sufficiently. The op amp with the feedback network R_1 and C_4 serves as a first-order low-pass filter reducing the high-frequency noise. Fig. 7 shows the circuit diagram of the op amp. It is a classical design [4] with a low-gain differential input stage driving a very high-gain cascode stage from both ends. This drives a Miller gain stage to give good gain, bandwidth, and output impedance with medium power consumption. The transistors are all large to give a low $1/f$ noise figure. The specifications of the op amp are shown in Table I.

V. PRACTICAL D/A CONVERTER CHIP

Fig. 8 shows a photograph of the stereo 16-bit converter. The circuit is processed in a 2-μm CMOS technology. The stereo D/A converter needs a chip area of 44 mm^2 and is

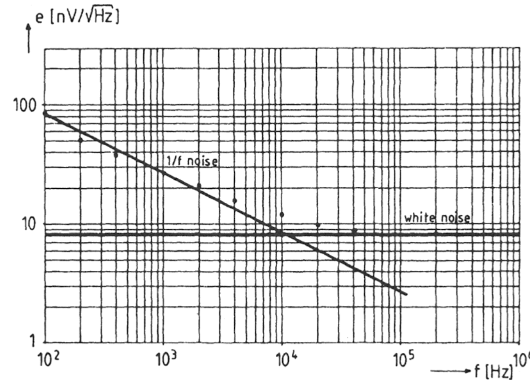


Fig. 10. Measured $1/f$ plus thermal noise of the op amp.

TABLE II
NOISE CALCULATIONS

Input quantization noise (16 bit)	- 98.4 dB
Quantization noise first filter section	-104.4 dB
Noise-shaper	-105.8 dB
Opamp 1/f noise relative to 0.7Vrms	-104.1 dB
Opamp thermal noise relative to 0.7 Vrms (folded noise included)	-103.0 dB
Theoretical total noise	- 95.6 dB

mounted in a 40-pin dual-in-line plastic package or in a 44-pin quad flat pack.

Two inter IC signal (I^2S) [5] ports have been incorporated, one to receive data from the compact-disc decoder IC (or any 16-bit 44.1-kHz I^2S source), and the other to transmit the four-times oversampled data to an external device such as a dual 16-bit D/A converter [6]. The I^2S standard, in which the two channels are multiplexed, provides easy interfacing between digital signal processing devices operating at various word lengths. For compact-disc applications an input has been provided which will allow an attenuation of the analog output by -12 dB during track search. A de-emphasis input can control the roll-off of the integrator by switching in an extra (external) feedback network.

VI. MEASUREMENTS

Fig. 9 shows the noise and the total harmonic distortion as a function of the frequency measured with an HP 339A distortion-measurement set and a spectrum analyzer with a resolution bandwidth of 30 Hz. The input signal has a frequency of 3 kHz. At full signal level (-2 dB$_n$) the harmonic distortion is less than -100 dB$_n$.

Fig. 10 shows the thermal noise and the $1/f$ noise measured at the op amps. It can be seen that the $1/f$ noise plays a dominant role.

The minimum noise level generated by a D/A converter system is given by the resolution of the input signal. Extra noise will be added owing to imperfections of the converter and noise generated by the low-pass filter. In the case of a partially digital and partially analog filter, round-

TABLE III
MEASURED DATA OF THE STEREO 16-BIT D/A CONVERTER

Dynamic range	> 94 dB
Distortion	< -90 dB
Dissipation	< 250 mW
Pass-band ripple	< 0.02 dB
Stop-band attenuation	> 60 dB
Power supply	single 5V ±10%
Package	44 pin QFP or 40 pin DIL
De-emphasis switch	internal
Output low-pass filter	internal
Serial input interface	I^2S
Serial output interface	I^2S
X-tal oscillator	internal
Die size	44 mm^2
Process	2 μm CMOS

ing noise is generated in the digital filter and $1/f$ and thermal noise in the analog filter. The noise contributions, over a 20-kHz band, of the several components used in our system are given in Table II (0 dB corresponds to a full-level sine wave). Measurements show a noise level of -94 dB which is fairly close to the calculated noise level. The main noise contribution is given by the noise of the op amp.

Table III gives some additional data for the stereo 16-bit D/A converter.

VII. CONCLUSIONS

A stereo 16-bit D/A converter for use in digital audio equipment has been realized in a 2-μm CMOS process by using a code-conversion technique based upon oversampling and noise shaping. The oversampling filters, the noise shaper, the 1-bit D/A converter, and the analog post-filter are integrated on one chip. For most applications only a few external components are required. The power consumption is <300 mW and only a single 5-V supply is needed.

ACKNOWLEDGMENT

The authors wish to acknowledge the valuable contributions made by D. Goedhart of the CD-lab, Eindhoven, The Netherlands, and A. Durham and D. Braithwaite of Mullard, Southampton, United Kingdom.

REFERENCES

[1] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[2] A. Oppenheim and R. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[3] H. A. Spang and P. M. Schultheiss, "Reduction of quantizing noise by use of feedback," *IRE Trans. Commun. Syst.*, pp. 373–380, Dec. 1962.
[4] P. R. Gray and R. G. Meyer, "MOS operational amplifier design—A tutorial overview," *IEEE J. Solid-State Circuits*, vol. SC-17, no. 6, pp. 969–982, Dec. 1982.
[5] Philips publication, "I^2S bus specification," Feb. 1986.
[6] H. J. Schouwenaars, E. C. Dijkmans, B. M. J. Kup, and E. J. M. van Tuijl, "A monolithic dual 16-bit D/A converter," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 424–429, June 1986.

Author Index

A

Adams, R. W., 206, 279, 340
Agrawal, B. P., 153
Alexander, D. S., 377
Amano, K., 340
Ardalan, S. H., 33, 163, 245

B

Benjamin, O. J., 52, 385
Bishop, R. J., 163
Boser, B. E., 168, 293
Bradinal, W., 486
Brauns, G. T., 245
Brinthaupt, D. M., 311
Brodersen, R. W., 213, 317, 333
Burrus, C. S., 405
Buzo, A., 106

C

Caldwell, J., 359
Candy, J. C., 1, 44, 52, 174, 377, 385, 400, 477
Cardoletti, L., 449
Carey, M. J., 461
Carley, L. R., 184, 304, 482
Cataltepe, T., 192, 227, 266
Chao, K. C.-H., 196
Chen, D.-P., 311
Ching, Y. C., 377
Chou, W., 60
Chu, S., 405
Copeland, M., 233
Crochiere, R. E., 417

D

Degrauwe, M., 449, 453, 457
Del Signore, B. P., 365
Deppa, T. W., 311
Deval, P., 249
Dijkmans, E. C., 486
Dijkstra, E., 449, 453, 457

E

Eckbauer, F., 326
Elward, Jr., J. P., 311
Engelhardt, E., 326

F

Ferguson Jr., P. F., 206
Fetterman, H. S., 342
Fields, E. M., 311
Fisher, J. A., 326
Friedman, V., 311

G

Ganesan, A., 206
Giancario, C., 363
Goodman, D. J., 461
Gosslau, A., 209
Gottwald, A., 209
Gray, P. R., 333
Gray, R. M., 60, 73, 81
Greene, R., 359

H

Hallock, R. W., 139
Hamashita, K., 365
Hara, S., 365
Haug, J., 359
Hauser, M. W., 213, 219, 317
Hayashi, T., 259, 320
Heise, B., 326
He, N., 106
Holland, D., 486
Hurst, P. J., 223, 317
Huynh, A.-N., 477

I

Inabe, Y., 320
Inose, H., 115
Ishii, E., 340
Ishikawa, M., 237
Iwata, A., 237, 259

K

Karema, T., 127, 322, 353
Karmann, K. P., 168
Kenney, J., 482
Kimura, T., 259
Kobayashi, T., 237
Koch, R., 326
Kramer, A. R., 192
Kuhlmann, F., 106

L

Larson, L. E., 192, 227
Le Fur, P., 467
Lee, W. L., 196, 363
Leslie, T. C., 229
Leung, B. H., 333
Levinson, R. A., 223
Lindell, M., 353
Longo, L., 233

M

Martin, H., 168
Matsumoto, K., 340
Matsuya, Y., 237
McKnight, A. J., 486
Meleis, H., 467

N

Nadeem, S., 196
Naus, P. J. A., 486
Neff, R., 333
Norsworthy, S. R., 342
Nys, O., 60, 449, 453, 457

P

Parzefall, F., 326
Paulos, J. J., 33, 163, 245
Piguet, C., 449, 457
Post, I. G., 342

R

Rabiner, L. R., 417
Rakers, P., 359
Rebeschini, M., 359
Rijmenants, J., 453
Ritoniemi, T., 127, 322, 353
Robert, J., 249

S

Saramäki, T., 471
Schultheiss, P. M., 131
Scott, J. W., 311, 363
Shenoi, K., 153
Shoji, Y., 255
Singh, B., 229
Sodini, C. G., 196
Spang III, H. A., 131
Steer, M. B., 163, 245
Stikvourt, E. F., 486
Suzuki, T., 255
Swanson, E. J., 365

T

Takasuka, K., 365
Tanaka, T., 365
Temes, G. C., 1, 192, 227, 266
Tenhunen, H., 127, 322, 353, 471
Tewksbury, S. K., 139

U

Uchimura, K., 237, 259, 320

V

van Bavel, N., 359
Viswanathan, T. R., 311

W

Walden, R. H., 192, 266
Welland, D. R., 365
Wong, P.-W., 60
Wooley, B. A., 168, 293, 385

Y

Yasuda, Y., 115
Yoshitate, K., 340
Yoshitome, T., 237
Yukawa, A., 270

Subject Index

Abstract dynamical system, 76
Analog section, dual-channel voice-band PCM codec, 312-13
Arbitrary loop gain, analysis of system with, 49-51
Audio A/D converter:
  D/A conversion, 290-91
  decimator design, 286-90
    second stage of decimator, 288-89
    truncation, effect on output spectrum, 289-90
  design/implementation, 279-92
  extension to higher resolution, 291
  front-end A/D design, 280-82
  front-end topology:
    linearity considerations, 283
    selection of, 282-83
    waveform symmetry, 283-86
    integrator considerations, 291
    performance of prototype, 291

Baseband filter, 463
Baseband noise, 55-57
Binary quantizer error, moments of, 65-68
Block diagram, dual-channel voice-band PCM codec, 311-12

Cascaded demodulators, 23-24
Cascaded modulators, 11-12
Circuit parameters:
  effect on signal, 46-48
  loop gain/circuit stability, 46-48
  threshold settings, 48
CMOS sigma-delta A/D converter:
  with 5 V operating voltage, 353-58
    design, 353-54
  amplifier gain, 354-55
    amplifiers, 355-56
    area, 356
    capacitor nonidealities, 355
    comparator, 356
    comparator hysteresis and speed, 355
    differential circuit topology, 355
    integrators, 355
    switches, 356
  modulator noise, 354
  with 350-kHz output rate, 359-62
    capacitor mismatch, 361
    experimental results, 361-62
    folded cascade structure, 361
    modulator circuit implementation, 360-61
    theoretical analysis, 360
    thermal noise, 361
    voltage-variable switched-charge-injection, 361
CMOS slope adaptive delta modulator, 363-64
CMOS stereo 16-bit D/A converter for digital audio, 486-90
  measurements, 489-90
  noise shaper, 487-88
  one-bit D/A converter, 488-89
  oversampling and filtering, 486-87
  practical D/A converter chip, 489

D/A quantization levels, positioning, 6
DC idle-channel noise, interpolative modulators, 202-3
Dead zones, 5
Decimating the modulated signal, 18-21
  first-stage decimator design, 19-20
  low-pass filter, 21
  multistage decimation, 18-19
  sinc decimators, implementing, 20-21
Decimation, 400-404
  decimating with sinc^K, 401-2
  defined, 400
  of digital signals, 417-48
  digital words, length of, 402
  first-stage decimator design, 19-20
  graphical presentations, 403-4
  low-pass filter, 21
  modulator, 400-401
  multistage, 18-19
  overloading characteristics, 402-3
  sinc decimators, implementing, 20-21
  of spurious noises, 403
Decimation filter design, 453-56
  design example, 455
  finite wordlength of coefficients, 455
  required filter complexity, 453-54
  simulated annealing optimization of wordlength, 455
  software environment, 455
Delta modulation, 12, 115
  noise from, 58-59
Delta-sigma A/D converters:
  analog constraints, 219-21
  area and power, 221-22
  basic building blocks, 330-32
    bidirectional current source, 330-31
    chip architecture, 332
    comparator, 331-32
    switched current source, 331
  circuit implementation, 345-48
    operational details, 346-47
    performance limitations, 346
    subcircuit characteristics, 347-48
  core circuitry, 219
  decimation filter design, 453-56
  with 15-MHz clock rate, 326-32
  functional principle, 326-27
  integrator concept, 328-30
  measured results, 327-28
  mixed A/D constraints, 221
  modeling/simulation, 345
  MOS converters, 213-18
  performance evaluation, 348-50
    alternative measures, 348-49
    custom data acquisition board, 349-50

Delta-sigma converters (*cont.*)
performance evaluation system, 349
with reduced sensitivity to op-amp noise and gain, 223-26
sigma-delta conversion, 342-45
architectures for practical realization, 343-44
digital filtering/decimation, 344-45
higher-order noise-shaping modulators, 345
nonlinearities in, 344
quantization noise, 343
VLSI-realizable decimators for, 471-74
Delta-sigma demodulation, circuit design for, 24
Delta-sigma interpolative modulator, 196-97
Delta-sigma modulation, 3-7, 34-35, 60-72, 175-76
alternative modulator structures, 10-12
cascaded modulators, 11-12
delta modulation, 12
error feedback, 10-11
circuit parameters, influence of, 5-7
with DC inputs, pattern noise from, 4-5
dead zones in, 5
decimation for, 400-404
design methodology, 153-62
circuit-level design, 157-58
D/A or digital design, 158-59
discrete-time model, 154-55
parameter definitions, 155-56
systems-level design, 156-57
uniform quantizer, analogy with, 156
double integration in, 174-83, 255-57
simulation results, 257
first-order feedback circuit, 3
gain compensation, 224-25
loop stability, 259
low-frequency noise compensation, 224
model, 53-54
analysis of, 54-57
modulation noise in busy signals, 3-4
multistage, 60-72
multilevel quantization, 58
noise from, 52-53
noise in, 7-8
nonlinear behavior of, 159-61
performance limits due to circuit non-idealities, 223-24
quantization noise, 75-76
random noise, analysis based on, 58
stability, 159-61
tri-level, signal-to-noise ratio using, 245-48
See also Multistage sigma-delta modulation
Delta-sigma modulators:
basic theory of operation, 229-30
coder topologies, 230
DC input, 35-37
improved architecture, 229-31
simulation program, 231
simulation results, 231-32
modulo arithmetic comb filters in, 457-60
MOS ADC-filter combination, 317-19
noise spectra and signal-to-noise ratio, 38-41
nonlinear behavior in, 33-43
nonlinear quantizer, modeling of, 35-41
optimization by use of a slow ADC, 209-12
sinusoid input and nonlinearity modeling, 37-38
stability analysis, 41-42
table-based simulation of, 163-67
thresholding in, 379-81
triangularly weighted interpolation, 377-84
using multibit quantizers, 16-18
See also Oversampled A/D converters; Oversampled D/A converters
Digital filters, 461-66
baseband filter, 463
characteristics, 466
design chart, using, 462-63
design method, 462-63
performance levels, range of, 462
example, 463
implementation, 464-65
optimization method, comparison with, 465
requirements, 461-62
resampling, 463-64
examples, 464
general principle, 463-64
Digitally corrected multi-bit sigma-delta data converters, 192-95
A/D converter, 193
D/A converter, 192-93
experimental results, 193-94
self-calibration of, 193
Digital modulation, 2-12
delta-sigma modulation, 3-7
high-order modulators, 7-10
quantization, 2-3
Digital section, dual-channel voice-band PCM codec, 313-15
Digital signals:
FIR filter design, 429-38
interpolation/decimation of, 417-48
quantizing, 23
sample rate conversion, multistage implementations of, 438-46
sampling rate conversion, basic concepts of, 418-23
signal processing structures for decimators/interpolators, 423-29
comparisons of structures, 429
direct form FIR structures for integer changes in sampling rates, 424-25
FIR structures with time-varying coefficients, 428-29
polyphase FIR structures for integer decimators and interpolators, 425-28
signal-flowgraphs, 423-24
Dither, 91-93, 185-87
A/D converter system, modeling, 186
dither distribution vs. quantizer error, 185-86
and output truncation, 187
quantization error:
modeling, 186
PSD of, 186-87
Double integration in sigma-delta modulation, 174-83
decimator, design of, 181-82
digital processor, design of, 179-80
feedback, penalties for using, 176-77
limit cycles that overload the quantizer, 178-79
method, 175-76
modulation noise, 180-81
quantization with feedback, 174-75
toll network use requirement, 175
two integrators, use of, 179
two-level quantization, 177-78
Double integrator, 118
Double interpolation for D/A conversion, 477-81
circuit parameters, precision of, 480-81
first-order interpolation, 477-78
measurements on a circuit model, 479
second-order interpolation, 478-79
Double-loop sigma-delta modulation, 106-14
discrete-time model, 107-8
difference equation solution, 108
difference equations, 107
stability problems, 107-8
long-term time average properties, 108-11
fundamental result, 109-10

preliminaries, 108-9
properties of various processes, 110-11
system performance, 111-12
comparisons, 112
optimum FIR filter, 111
sigma-delta quantization noise, 111
sinc filters, 111-12
Dual-channel voice-band PCM codec, 311-16
analog section, 312-13
block diagram, 311-12
digital section, 313-15
digital sigma-delta modulator, 315
IIR filters, 314-15
sinc cube decimator, 313-14
experimental results, 315-16
Dynamical system, 76-77

Elliptical filter design techniques, 204
Equiripple (optimal) FIR designs, 433-35
Error feedback, 10-11
quantization with, 23
Esaki diode, 122-24

Feedback:
penalties for using, 176-77
quantization with, 174-75
reducing quantization noise by use of, 131-38
Feedback loop, net gain in, 5-6
Finite op-amp gain, interpolative modulators, 201
FIR filter design, 429-38
architectural design, 468-69
generating the coefficients, 468
performing the direct convolution, 468-69
decimation, realization for, 467-68
equiripple (optimal) FIR designs, 433-35
filters based on window designs, 432-33
half-band FIR filters, 435
ideal frequency domain characteristics, 431-32
minimum error in frequency domain, 437-38
minimum mean-square-error design, 435-36
optimum FIR filter, 111
procedures, 432
prototype filter/polyphase representation, relationship between, 430-31
time-domain properties of ideal interpolation and decimation filters, 432
First-order delta-sigma modulators, 13-14, 249
First-order feedback circuit, 3
First-order incremental A/D converter, 250-51
First-stage decimator, design, 19-20

Half-band FIR filters, 435, 461
Higher-order cascade modulators, 16
Higher-order single-stage modulators, 15-16
High-order modulators, 7-10, 196-205
architectures for, 265-69
multistage sigma-delta modulators, 266-67
dynamic range of, 8-9
in-band values of quantization error, predicting in, 7
noise in, 7-8
noise shaping, 10
second-order modulators, influence of circuit parameters on, 9
third-order modulators, limit cycles in, 9-10
High-order one-bit sigma-delta A/D converters, 206-8
experimental results, 208
theory, 206-8
High-order one-bit sigma-delta modulators:
design of, 127-30
examples, 128-29
modulator analysis, 127-28
modulator structures, 128
optimized NTF, 128

Integration:
oversampled A/D converters, 191
range of, 9
Integrators, leakage in, 6-7, 9
Interpolation, 377-79
analysis of, 378-79
triangularly weighted, 377-84
Interpolative modulators, 196-205
elliptical filter implementation, 204
experimental results, 203-4
*N*th-order topology, 198-200
loop coefficients, design of, 199-200
stability, 199
system function, 198-99
quantization noise, 200
simulation results, 200-203
coefficient errors, 202
DC idle-channel noise, 202-3
finite op-amp gain, 201
integrator settling time, 207-8
limited op-amp swing, 202

Limit cycle oscillations, 44-51
arbitrary loop gain, analysis of system with, 49-51
circuit parameters, effect on signal, 46-48
operating modes, 45
pulse-code modulation (PCM), circuit for, 44-45
resolution, 45-46
time-shared operation, 48-49
timing circuit operations, 48
Limited op-amp swing, interpolative modulators, 202
Low-pass filter, 388-92
and decimation of modulated signal, 21
ideal low-pass filter decoder, 69

MASH modulator circuit configuration, 14
for oversampled A/D converters, 262-63
for oversampled D/A converters, 263-64
Modulation noise, 3, 7-8, 180-81
Modulator architectures, 13-18
criteria for choosing the architecture, 13
oversampling modulator architectures, 13-18
scaling signal amplitudes in modulators, 18

Modulo arithmetic comb filters, 457-60
architectural considerations, 457-58
VLSI implementation, 458
MOS A/D converter filter combination, 317-19
MOS delta-sigma A/D converters, 213-18
capacitor linearity, 216
op amp requirements, 216-17
oversampling converters in MOS, 213-14
performance limits due to noise, 215-16
ratio- and parasitic-insensitivity, 214-15
simulation, 216
*M*th-order incremental A/D converter, 252
Multi-bit quantizers, delta-sigma modulators using, 16-18
Multi-bit sigma-delta data converters, digitally corrected, 192, 227-28
Multichannel oversampled PCM voice-band coder, 333-39
coder implementation, choice of, 333-35
experimental results, 337-38
prototype, 335-37
Multilevel quantization, 58
Multirate filter designs, 405-16
multistage, 405-7
examples/comparisons, 412-16
FIR design, 412
IIR design, 411-12
using comb filters:
stability/arithmetic of, 409-11
structures, 407-9
Multistage decimation, 18-19
Multistage delta-sigma modulator, without double integration loop, 320-21
Multistage sigma-delta modulation, 60-72, 266-67
average quantization noise power, 71-72
binary quantizer error, 64-69
moments of, 65-69
decoding filters, 69-70
ideal low-pass filter decoder, 69
sinc^K filter decoder, 69-70
difference equation and linear network, 61-64
digital correction, 267
multiple sinusoids, inputs with, 71
simulation results, 70

Noise shaping, 174
high-order modulators, 10, 23
Noise-shaping coders of order $N > 1$, 139-49
design, 148-49
optimum oversampled higher order linear noise-shaping coder structures, 43-44
optimum oversampled higher order linear predictive coder structures, 141-43
oversampled linear feedback coder structures, 139-41
predictive coder design, 146-48
third-order predictive coder, 144-46
Noise-shaping coder topology, 304-5
design example, 307-10
results, 309-10
system topology, 307
three-bit internal D/A converter, 307-9
for 15+ bit converters, 304-10
internal D/A converter topology, 305-7
dynamic element matching approach, 306-7
randomizer, 307
Nonlinear behavior, in delta-sigma modulators, 159-61, 297-98, 344
Nonlinear quantizer, modeling of, 35-41
*N*th-order topology, 198-200
loop coefficients, design of, 199-200
quantization noise, 200
stability, 199
system function, 198-99

Output truncation, and dither, 187
Oversampled A/D converters:
advantage of, 196
constraints analysis, 270-75
integrator settling and quantizer uncertainty effects, 274-75
with lossy integrator, 273-74
design, 293-303
electronic noise, 295
implementation, 299-301
integrator and comparator design, 296-99
sampling jitter, 295-96
second-order sigma-delta modulator, 294
signal range, 294-95
for digital audio, 340-41
with digital error correction, 227-28
dither, 185-87
DPCM loop, 187-89
random errors, 188
start-up problem, 188-89
tracking error, 188
experimental results, 189-90, 301-2
implementation, 299-301
circuit topology, 299-300
comparator design, 301
integrator design, 300
implementations/applications, 277-374
integration, potential for, 191
integrator and comparator:
bandwidth, 297
comparator hysteresis, 298-99
design, 300-301
gain variations, 296
leak, 296-97
nonlinearity, 297-98
slew rate, 297
interpolative modulators for, 196-205
MASH modulator circuit configuration for, 262-63
multistage noise shaping modulator performance, 261-62
multistage noise shaping modulator quantization noise, 259-61
performance of, 168-69
simulating/testing, 168-73
comparison with other techniques, 171-72
practical considerations, 170-71
sinusoidal minimum error method, 169-70
smoothing signal, 189
structures, 270-72
switched capacitor integrator models, 272-74
integrator without offset cancelling, 273
offset cancelling integrator, 272-73
thirteen-bit ISDN-band oversampled A/D converter, 233-36
topology, 184-85
wave digital decimation filters in, 449-52
Oversampled D/A converters, 259-65
MASH modulator circuit configuration for, 263-64
multistage noise shaping modulator performance, 261-62
multistage noise shaping modulator quantization noise, 259-61
Oversampled linear feedback coder structures, 139-41
Oversampled sigma-delta modulation, 73-80, 106

quantization noise, 75-76
quantizer performance, 78-79
using two-stage fourth-order modulator, 322-25
Oversampling converters, theory of, 279-80
Oversampling methods, 1-25
D/A converters, 21-24
demodulator stage, 23-24
demodulating signals at elevated word rates, 21-22
interpolating with sinc^K-shaped filter functions, 22-23
decimating the modulated signal, 18-21
digital modulation, 2-12
modulator architectures, 13-18
popularity of, 1
resolution, 24-25
Oversampling modulator architectures, 13-18
delta-sigma modulators using multibit quantizers, 16-18
first-order delta-sigma modulators, 13-14
higher-order cascade modulators, 16
higher-order single-stage modulators, 15-16
second-order cascaded modulators, 14-15
second-order delta-sigma modulators, 14

Pattern noise, 4-5
PCM quantization noise, 87-91
Performance evaluation, sigma-delta A/D converter, 348-50
Periodic noise, 119-20
Predictive coder, design, 146-48
Pulse-code modulation (PCM), circuit for, 44-45

Quantization, 2-3
two-level, 177-78
Quantization error, 84, 132-34
modeling, 186
predicting i-band values of, 7
PSD of, 186-87
vs. dither distribution, 185-86
Quantization levels, alignment of, 48
Quantization noise, 75-76, 87-91, 116-18, 200
interpolative modulators, 200
oversampled sigma-delta modulation, 75-76
reducing by use of feedback, 131-38
error spectrum, 132-34, 137-38
optimization of system, 134-37
structure of, 52-59
in uniform quantizers, 75
Quantization noise spectra, 81-105
dithering, 91-93
extensions, 103-4
PCM quantization noise, 87-91
second-order sigma-delta modulation, 102-3
single-loop sigma-delta modulator, 93-100
two-stage sigma-delta modulation, 100-102
uniform quantization, 81, 84-87
Quantization thresholds, positioning, 6

Random noise, 58
Reduction, sample rate, 418-20
Running-sum filter, 323-24
filter processor, 324

Sampling rate conversion, 418-23
by bandpass signals, 422-23
by rational factor M/L, 421-22
increase, 420-21
multistage implementations of, 438-46
designs based on specific family of filter designs, 445-46
half-band designs, 442-43
multistage optimization procedures, designs based on, 443-45
parameter specifications, 441-42
two-stage structure, computational efficiency of, 439-40
reduction, 418-20
Scaling signal amplitudes, in modulators, 18
Second-order cascaded modulators, 14-15
Second-order delta-sigma modulators, 14, 102-3
circuit tolerances, 9
Second-order incremental A/D converter, 249-54
compared to sigma-delta converters, 253
experimental results, 253-54
offset and charge injection compensation, 252-53
Second-order interpolative modulator, 197-98
Second-order modulators, influence of circuit parameters on, 9
Settling time:
interpolative modulators, 207-8
oversampled A/D converters, 274-75
16-bit oversampling A/D converters, 239-40
Signal-to-noise ratio, 38-41, 379
numerical calculation of, 39-40
optimization of, 212
using tri-level delta-sigma modulation, 245-48
Signal-to-quantization-noise ratio, 82, 120
Sinc^K filter decoder, 69-70
Sinc decimators, implementing, 20-21
Sinc filters, 111
Single-loop sigma-delta modulation, 93-100, 106
DC input, 94-96
sinusoidal inputs, 96-100
Sinusoidal minimum error method, 169-70
16-bit fourth order noise-shaping D/A converter, 482-85
experimental results, 484-85
system topology, 482-83
3-bit D/A converter topology, 483-84
16-bit oversampling A/D converters, 237-44
circuit considerations, 241-42
analog circuit configuration and operation, 241-42
block diagram, 241
digital filter configuration and operation, 242
measurement characteristics, 242-44
practical accuracy limiting factors, 239-40
capacitance-matching tolerance, 240
integrator gain and settling speed, 239-40
reducing noise from digital circuits, 240
theoretical accuracy limiting factors, 238
three-stage MASH operation, 238-39
Stereo 16-bit delta-sigma A/D converter, 365-74
architecture, 368
delta-sigma conversion, 365-68
filtering and decimation, 367-68
loop analysis, 366-67
loop filters, 367
quantization noise, 365-66
design, 369
digital filter output, 371-73
discrete-time implementation, 368-69
filter/decimator, 370-71
anti-aliasing filtering, 371

decimation, 370
passband shaping, 370-71
filter order, 369
major characteristics, 368
modulator output, 371
one-bit quantizer, 369
oversampling ratio, 369
simulated performance, 370
specifications, 373
stability, 369-70

Table-based simulation of delta-sigma modulators, 163-67
circuit description, 164
generating tables, 164-65
ZSIM, 163-64
ZSIM simulation results, 165-67
enlarged switches, 166-67
increased clock rate, 166
nominal case, 165-66
Third-order modulators, limit cycles in, 9-10
13-bit ISDN-band oversampled A/D converter, 233-36
architecture, 233-34
implementation, 235
performance, 235-36
Triangularly weighted interpolation:
delta-sigma modulators, 377-84
circuit arrangement, 383-84
harmonic distortion and gain tracking, 381
signal-to-noise ratios, 379
spectra, 381-82
thresholding, 379-81
Tri-level delta-sigma modulation:
signal-to-noise ratio using, 245-48
simulation, 246
tri-level coding, 245-46
implementation of, 246-47
Two-level quantization, 6
Two-stage fourth-order modulator:
oversampled sigma-delta A/D converter circuit using, 322-25
running-sum filter, 323-24
Two-stage sigma-delta modulation, 100-102

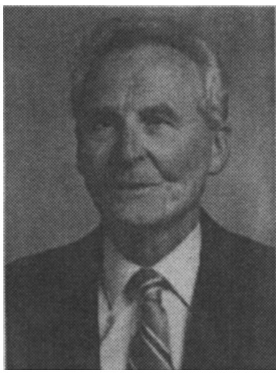
Uniform quantization, 81, 84-87
analyzing, 83
quantization error, 84
quantization noise, 75-76, 116-18
Unity bit coding method by negative feedback, 115-26
frequency characteristics, 120
input-output characteristics, 120
principle, 115-16
signal-to-noise characteristics, 116-20
periodic noise, 119-20
quantizing noise, 116-18
SNR and signal amplitude, 119
signal-to-quantizing-noise ratio, 120
video signal coding, 122-24

VLSI-realizable decimators, 471-74
filter architecture, 473
filter performance, 472-73
problem statement, 471
proposed class of decimators, 471-72
Voiceband codec with digital filtering, 385-99
decoder, 391-94
demodulator, 393-94
interpolator, 392-93
low-pass filter, 391-92
encoder, 385-90
decimator, 386-88
high-pass filter, 390
low-pass filter, 388-90
modulator, 386
quantization noise in, 395-98
simulated encoder response, 390
experimental codec response, 394-95
gain management, 398-99
second-order filter sections, overflow oscillations and roundoff noise in, 398

Wave digital decimation filters, 449-52
architectural issues, 451
design example, 451
filter specifications, 450
finite wordlength effects, 450-51
system overview, 449-50
Wordlength, decimation filters, 455

ZSIM, 163-67
simulation results, 165-67
enlarged switches, 166-67
increased clock rate, 166
nominal case, 165-66

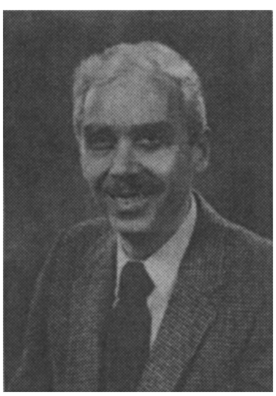
Editors' Biographies



James C. Candy was born in Crickhowell, South Wales in 1929. He received the B.Sc. and Ph.D. degrees in engineering from the University of North Wales, Bangor in 1951 and 1954 respectively.

From 1954 to 1956 he was with S. Smith and Sons, Guided Weapons Department, Cheltenham, and for the next three years he worked on nuclear instrumentation at the British Atomic Energy Research Establishment, Harwell. In 1959 he came to the United States to take up an appointment as Research Associate at the University of Minnesota. A year later he joined AT & T Bell Laboratories, New Jersey where he has been engaged in research on digital signal processing, including efficient encoding of video and speech signals. He has also been interested in methods for converting signals between digital and analog formats.

Dr. Candy is a fellow of the IEEE. He has published 30 technical papers and holds twenty-nine U.S. patents.



Gabor C. Temes received his Dipl.Ing. from the Technical University of Budapest in 1952, his Dipl. Phys. from Eotvos University, Budapest in 1954, and the Ph.D. in Electrical Engineering from the University of Ottawa in 1961.

Dr. Temes was a member of the faculty of the Technical University of Budapest from 1952 to 1956. He was employed by Measurement Engineering Ltd., Arnprior, Ontario, Canada, from 1957 to 1959. From 1959 to 1964 he was with Northern Electric R&D Laboratories, Ottawa, Ontario, Canada. From 1964 to 1966 he was a research group leader at Stanford University, Stanford, California, and from 1966–1969, a Corporate Consultant at Ampex Corporation, Redwood City, California. He is now on the faculty of the University of California, Los Angeles, as a Professor in the Department of Electrical Engineering. Between 1975 and 1979, he was also Chairman of the Department.

Dr. Temes is a former Associate Editor of the *Journal of the Franklin Institute*, a former Editor of the *IEEE Transactions on Circuit Theory*, and a former Vice-President of the IEEE Circuits and Systems Society. He is a Fellow of the IEEE.

In 1968 and 1981, Dr. Temes was a cowinner of the Darlington Award of the IEEE Circuits and Systems Society. In 1981, he received the Outstanding Engineer Merit Award of the Institute for the Advancement of Engineering. In 1982, he was awarded the Western Electric Fund Award of the American Society for Engineering Education and in 1984, he was awarded the Centennial Medal of the IEEE. He received the Andrew Chi Prize Award of the IEEE Instrumentation and Measurement Society in 1985, the Education Award of the IEEE Circuits and Systems Society in 1987, and the Technical Achievement Award of the same Society in 1989. He is coeditor (with S. K. Mitra) and coauthor of *Modern Filter Theory and Design*, Wiley, 1973; coauthor of *Introduction to Circuit Synthesis and Design*, McGraw-Hill, New York, 1977; coauthor of *Analog MOS Integrated Circuits for Signal Processing*, Wiley, 1986, and a contributor to several other edited volumes. He has published approximately 150 papers in engineering journals and conference proceedings.