Design Techniques for Sigma-Delta Modulators in Communications Applications

A Thesis submitted to the

Faculty of Engineering

of the University of Glasgow

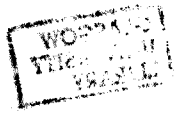
for the degree of

Doctor of Philosophy

by

David MacDonald Hossack

April 1997



Summary

Specialised design techniques for sigma-delta modulators are described in this thesis with all of the examples coming from modern communications systems.

The noise shaping and the signal transfer functions can be optimised using a weighted least squares approach. Numerical problems arising in the optimisation as a result of high oversampling rates are overcome through the use a simple transformation. The application to digitising audio is discussed, with the conclusion that Butterworth response noise shaping is preferable to inverse Chebyshev noise shaping for audio applications. An example of optimising the signal transfer function to provide immunity to instability brought about by large out-of-band signals is also presented.

The use of redundant arithmetic in the implementation of very high speed sigma-delta modulators is introduced, together with a DAC / filter combination suitable for reconstructing an analogue signal from the redundant arithmetic SDM.

An improved topology for a speech compander is described which offers a number of significant advantages over existing published methods. This uses no external components for ac coupling or setting the response time-constant, yet is robust and insensitive to parasitic components and process variations. This has been integrated on a CMOS IC process and the results are compared with the high level simulations.

A simulation method which allows the verification of switched-capacitor schematics with several orders of magnitude speed improvements over commercially available simulation tools is discussed. The method assumes ideal components, with internally controllable switches and reduces the schematic netlist to the few key equations that an experienced designer would derive manually. This process is fully automated and consequently is useful for providing confidence in implementations of complex SC systems.

For Kathy.

Acknowledgements

I would like to thank all those people who have helped in the completion this work. In particular I would like to thank my supervisors Professor J.I. Sewell (academic supervisor) and Professor J.R.C. Reid (industrial supervisor). In addition I would like to thank Dr. A.D. Milne for making the industrial liaison possible.

I would like to acknowledge Wolfson Microelectronics, Edinburgh for financial support throughout the research period. The SERC also contributed financially.

I am very grateful for advice and assistance from Dr. R.K. Henderson and Dr. Lu Yue.

My experience at Wolfson Microelectronics has brought me in contact with a large number of people who have helped direct the aims of this research to real-world problems and I would like to thank all those who have worked at WM during the past years for providing such a unique environment in which to perform this research.

Much of the work of implementing and testing of the fabricated chips was performed by others. Brian Bogie and Colman Clarke performed the measurements on the integrated devices. All of the test boards were constructed by Jake McTique.

Thanks must be given to the various computer system administrators who have helped with the multitude of small and most often trivial problems that occur when I work with computers.

Finally I would like to thank Dr. Katherine Eremin without whose encouragement and persistence this thesis would not have been completed.

TABLE OF CONTENTS

1

INTRODUCTION

1.1 SIGMA-DELTA MODULATION

In recent years sigma-delta modulation has become a widely used technique for analogue to digital and digital to analogue conversion. Sigma-delta modulators offer a number of unique advantages for such converters, particularly when the modulators are implemented using modern integrated circuit technologies [1,2]. In 1985 very few communications systems used sigma-delta modulation [3]. Within ten years this technique rose to become the predominant conversion technology for signals of bandwidths up to around 50kHz [4,5]. Although sigma-delta modulation is not a recent concept [6], its rapid increase in use has spawned much new interest in the research community.

Noise shaping has been an available technique for improving the performance of analogue to digital converters (ADC) for a long time [7]. Essentially the quantisation errors are fed back through a linear filter to influence subsequent quantiser decisions (Figure 1-1). The effect of the filter is to amplify the effect of the quantisation noise over some frequency range and reduce its effect over other frequencies. The result of this is that the quantisation noise can be significantly reduced over a narrow frequency range. Therefore the system must be operated at a higher sample rate than that dictated by the bandwidth of interest. This is termed oversampling. The output signal is often filtered elsewhere in the system to remove the shaped quantisation noise.

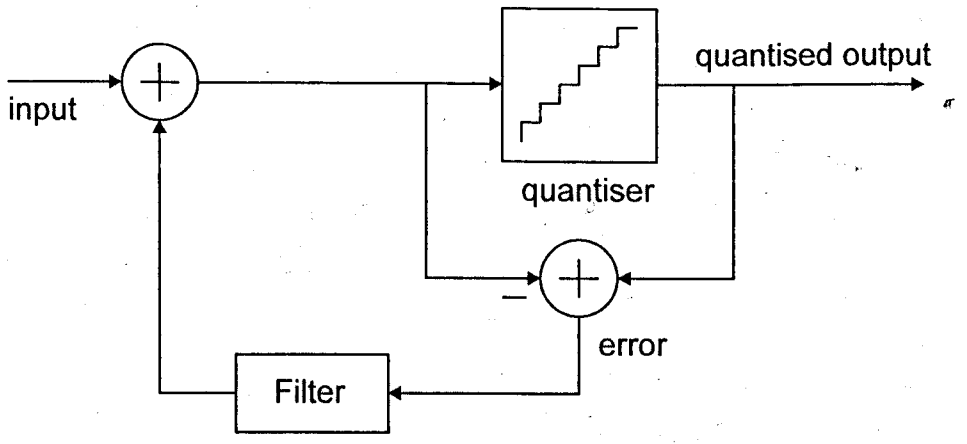


Figure 1-1 Noise-shaping quantiser.

The term sigma-delta modulator (SDM) is usually applied to an extreme implementation of noise shaping where the oversampling ratio has been increased such that only a few quantisation levels are required. Often only 2 to 16 quantisation levels are used with oversampling ratios in the range 50-250 to provide performance equivalent to 16-20 bit quantisers over the signal bandwidth. In particular, the single bit or two level quantiser has the unique property that its output levels inherently lie on a straight line. Whilst the two level sigma-delta modulator system is not inherently linear, it does retain the important feature that fabrication tolerances do not affect the linearity of the converter. Errors in the quantiser levels give rise to gain and offset errors which are usually of less importance than linearity. Thus with the two level system, it is possible to create very low distortion converters without particularly accurately matched components.

Sigma-delta modulators are often constructed using a variation on the noise shaping topology as shown in Figure 1-2 which can be shown to be equivalent to Figure 1-1. Throughout this thesis the transfer function of sigma-delta modulator loop filter is denoted by the symbol $H(z)$. This is a linear filter. The two level quantiser has a single decision level which can be implemented using a single comparator.

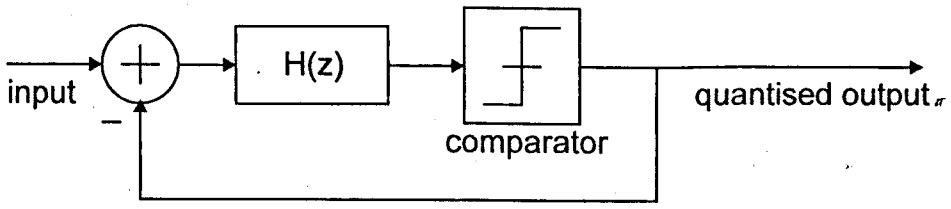


Figure 1-2 Sigma-delta modulator

When the quantiser of the basic sigma-delta modulator (Figure 1-2) is modelled by an additive noise source the resulting system (Figure 1-3) is linear. Transfer functions may be derived for the signal transfer function (STF) and the noise transfer function (NTF).

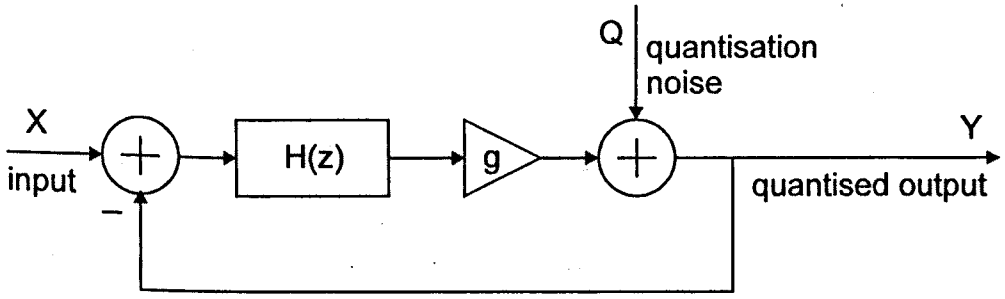


Figure 1-3 Linear Model of sigma-delta modulator

The equations are:

$$NTF = \frac{Y}{Q} = \frac{1}{1 + gH(z)} \tag{1-1}$$

$$STF = \frac{Y}{X} = \frac{gH(z)}{1 + gH(z)} \tag{1-2}$$

Typically *H(z)* has a very high gain at low frequencies giving very little quantisation noise in the signal band. The STF normally has a flat response through out the signal band. The loop filter may be extended to provide a definable numerator for the STF.

For a first order sigma delta modulator *H(z)* is an integrator with the transfer function $z^{-1}/(1-z^{-1})$ and it can be shown that the quantisation noise appears at the output having

been shaped with the high pass noise transfer function $1-z^{-1}$. The noise spectral density at low frequencies rises with the first order slope of +6dB/octave. In general, higher order loop filters of order n may be designed which yield noise transfer where the noise spectral density at low frequencies rises by $6n$ dB/octave.

Doubling the sampling frequency of a system with white quantisation noise gives only a 3dB improvement in the base-band signal to noise ratio (SNR) since the in-band noise is halved. When the noise has been shaped inside a sigma-delta modulator the improvement in SNR becomes $(6n + 3)$ dB for each doubling in sample rate. Thus for a second order SDM the performance improves by 15dB for each doubling in sample rate.

Since the output of a sigma-delta modulator contains significant high frequency noise, in most applications the output is low-pass filtered to remove this noise. If a discrete-time filter is used, it is possible to reduce the output sampling rate. The resulting overall system consisting of the SDM and low pass filter may then be compared to a conventional non-oversampled high resolution quantiser. A m bit quantiser has 2^m output levels. The SNR achievable from this is $1.76 + 6.02m$ dB and thus the SNR achieved from a sigma-delta modulated system may be directly compared to a m bit non-oversampled quantiser. Each doubling in sample rate of a n^{th} order sigma-delta modulator gives the performance equivalent of an additional $n+\frac{1}{2}$ bits.

1.2 APPLICATIONS

Sigma-delta modulators are most often used for analogue to digital and digital to analogue conversion. They can also be used for frequency synthesis purposes which have important applications in communications systems and for deriving the oversampled clock signal required for sigma-delta modulator based converters [8,9].

1.2.1 Analogue to Digital Conversion

For an analogue to digital converter (ADC), the loop filter and the quantiser are implemented using analogue circuits. The oversampled digital output is then digitally filtered and re-sampled to give a lower sample rate (decimated), but higher precision result. It has been the advances in digital technologies which has allowed these digital filters to manufactured cost-effectively. The analogue circuitry is most often implemented using SC techniques, although continuous time loop filters and switched-current based loop filter implementations have also been reported [5]. Continuous time techniques are used for either very high precision, low bandwidth converters, or for very high speed converters. For precision and bandwidths between these extremes, switched-capacitor techniques are preferred. Switched current methods have the advantage of being compatible with purely digital CMOS processes [10]. However because the noise and linearity of switched current integrators is relatively poor, few switched current based modulators give more than about 12 bit performance [11].

The amount of digital circuitry required for the decimation filter can be significant. The most common approach uses two decimation stages, the first being a comb filter ($sinc^n$) which can be implemented efficiently [12]. This filter operates on data at the oversampled rate. The second decimation filter is often a direct implementation of a decimating finite impulse response (FIR) filter [13]. Special techniques for reducing the computational requirements for decimating filters are becoming increasingly common [14].

1.2.2 Digital to Analogue Conversion

For digital to analogue converters (DAC), the sigma-delta modulator is implemented using digital methods. The implementation issues are similar to those for recursive, infinite impulse response (IIR) filters. One problem is that because of the oversampling, the modulator has to operate at high speed. Pipe-line techniques

cannot be used directly since the input of loop filter is a function of the output of the filter in the previous sample instant. This problem is addressed in Chapter 3. The shaped quantisation noise produced by the sigma-delta modulation process must be removed using an analogue filter. The performance of the overall converter will typically be limited by the performance of this filter. Improving the performance of these filters and designing new systems which produce less quantisation noise and so require less filtering are areas of active research.

The digital interpolation filter required to increase the input signal sampling rate up to the oversampling rate can be simpler than the corresponding digital decimation filter in the ADC case. This is because out-of-band images are further reduced by the analogue filter which is present for removing the shaped quantisation noise.

1.2.3 Multi-Level Converters

Modern silicon processes offer component matching accuracy suitable for the direct implementation of ten to twelve bit converters. The two level sigma-delta modulator does not exploit this accuracy.

Multi-level converters use more than two levels to represent the input signal. The correct alignment of these levels is of great importance in determining the linearity of the resulting converter. Multi-level sigma-delta modulators have many advantages. The lower quantisation step gives reduced quantisation noise with less stringent requirements on the filters used to remove the quantisation noise (both for ADCs and DACs) and a consequent reduction in power consumption. Multi-level sigma-delta modulators are also easier to stabilise and have less susceptibility for the generation of spurious tones. Typically components may be integrated which match to within one least significant bit at the 10 to 12 bit level. A multi-level sigma-delta modulator constructed using these components will exhibit linearity in the same 10 to 12 bit range. This has been the prime motivation to using the single bit, two level system, even although higher resolution DACs may be integrated reliably.

A recent trend has been to design multi-level sigma-delta modulators which retain insensitivity to component mismatch while exploiting the level of component matching which is available [15]. This is done by constructing the DAC from multiple unit elements and by ensuring the usage of each of the elements is such that errors have little effect over the band of interest. Errors in each of the unit elements result in additional noise which appears at the output as though it has been high pass filtered by a first order transfer function. Errors for DC signals are completely eliminated, while errors for low frequency signals are very much reduced. It is likely that these methods will become common practise in the future. A technique is discussed in Chapter 3 which provides a multilevel output where the linearity is independent on component matching.

1.3 COMMUNICATIONS APPLICATIONS

Sigma-delta modulators are used for analogue / digital conversion for many modern communications systems. There are two basic types of signal which may be converted: the source signal and the data signal (Figure 1-4). The source signal is the analogue speech or video signal which must be digitised before transmission and reconstructed upon reception. This thesis is limited to voice and audio type source signals. The data signal is the digital signal which has been modulated in some manner to allow transmission of the digital signal along an analogue transmission channel. However even entirely analogue transmission systems may benefit from using sigma-delta modulation to help perform signal processing. An example of this is discussed in Chapter 4.

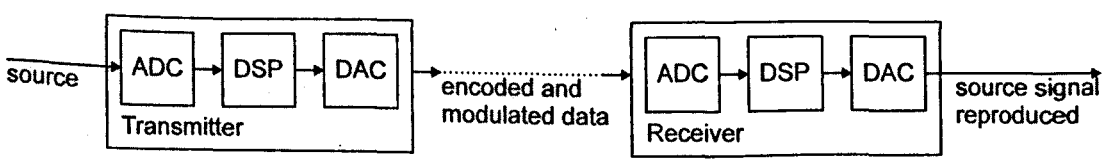


Figure 1-4 Modern Digital Communication Channel

1.3.1 Voice Channel

The key requirements for the voice channel of communications systems such as cellular and traditional telephone line components are for 4kHz bandwidth and 40-50dB signal to noise ratio (SNR). The dynamic range requirement is typically higher than the SNR requirement - in other words the noise is allowed to increase in the presence of large signals where the noise will be unobtrusive. The system discussed in Chapter 4 exploits sigma-delta techniques to yield this characteristic.

Higher performance than traditional telephone line quality is now becoming more common [16] for applications such as ISDN, video-conferencing, speech storage/playback and numerous other audio applications. A design technique to achieve higher perceived audio performance at lower oversampling rates is discussed in Chapter 2.

1.3.2 Data Channel

New methods for communications across noisy channels have become almost exclusively digital. However the digital data still needs to be modulated and transmitted over an analogue channel and high performance analogue / digital conversion is often required. Some recent communications standards have much higher performance requirements in terms of bandwidth, SNR and distortion than the voice channel, even if the data channel is principally being used to transmit the digitised voice signal [16]. This is due to the trend of squeezing channels increasingly tightly into the radio-frequency spectrum, whilst still rejecting large adjacent band or interfering signals. This is discussed further in Chapter 3. Another example of the high performance requirements of data channel converters is the case of the modems used to transmit data between computers along the standard dial-up telephone network. Here high precision is required from the ADC to ensure that no information is lost in the conversion, even in the presence of near-end cross-talk,

echoes and interference. Optimising the ADC for the case of interference with known characteristics is discussed in Chapter 2.

1.4 DESIGN TECHNIQUES

The mathematics required to describe the behaviour of sigma-delta modulators is very complex due to the presence of the quantiser which is very non-linear. The output of simple systems such as the first and second order sigma-delta modulators with constant inputs have been derived but these methods have yet to be applied to higher order modulators. However design techniques for high order sigma-delta modulators based on a linear model are now widely known. The design cycle is similar to the process for filter design - transfer function determination, realisation, and simulation. The simulation requirements are more severe than for the filter case as the modulator must be extensively simulated to determine the performance since analytic methods are not available.

1.4.1 Transfer Function Design

Sigma-delta modulators can be divided into two types: low order and high order. Low order modulators are either first or second order and are relatively simple to design [17]. Low order modulators offer limited noise shaping and require relatively high oversampling ratios. First order modulators particularly suffer from spurious tone effects, although almost all sigma-delta modulators exhibit spurious tone effects to some degree unless a noise signal (dither) is added to de-correlate the quantiser error signal from the input signal. Sigma-delta modulators of order greater than two, with only two quantisation levels were once believed to be unstable, but design techniques are now available which allow stable operation with input signal magnitudes of up to approximately half that of the reference [18].

The design of high order sigma-delta modulators relies on being able to approximate the quantiser as an additive noise source (Figure 1-3). Although this is a rather gross

approximation for the one-bit case, it allows the otherwise intractable mathematics to simplified. In practise the quantisation noise being fed back though the loop filter is enough to dither the quantiser sufficiently to make the approximation good enough to be useful. It is difficult to determine the gain g to assign to the quantiser when it has only two levels and rules of thumb must be used during the design process to ensure that the effective gain g of the quantiser is the same as that used during the design process.

During the course of the research period various pieces of software were written to help design and simulate both analogue and digital SDMs. Central to these is TFN which is used to describe and manipulate transfer functions in many convenient forms. The examples in this thesis are described using this transfer function language and a full description of this format has been included in the Appendix.

1.4.2 Realisation

Unlike the general filter situation, only a few different topologies are used for realising sigma-delta modulator loop filter $H(z)$ for the single quantiser sigma-delta modulator depicted in Figure 1-2. The topology originally suggested for high order modulators [18] was quickly superseded for reasons of sensitivity. The cascade of integrators and resonators topology [19] has remained the most commonly used topology.

The above discussion concentrated on the single loop sigma-delta modulator with a single quantiser. There is also the cascade type sigma-delta modulator which uses two or more simple sigma-delta modulators in cascade, with each modulator digitising the quantisation noise of its predecessor. This is little different from other cascade type conversion techniques such as pipe-line ADC converters, except with noise shaping. This work concentrates on the true single stage sigma-delta modulator, although a two stage cascade modulator is used as an example in Chapter 5.

There has also been some significant research into the area of band-pass sigma-delta modulators which are capable of directly quantising a narrow band intermediate frequency (IF) signal without down-converting to a base-band frequency region [20]. These converters are ideal for communications applications. Whilst many of the design techniques discussed in this thesis are also applicable to band-pass systems, this work concentrates on the base-band cases.

1.4.3 Simulation

Sigma-delta modulators require a well planned simulation strategy. Since to fully determine the performance requires many millions of clock cycles it is not appropriate to try to predict system performance using tools such as SPICE or even digital simulators for digital modulators - the simulation time is too great. Simulation of a switched-capacitor based sigma-delta modulator takes of the order of a few days to obtain the number of output samples required to obtain a single signal to noise ratio value, for a particular input amplitude and frequency. Complete characterisation of the system using SPICE is therefore unrealistic. It is necessary to model the system with various levels of refinement. Transistor level models are simulated over a few cycles to prove the functionality of the low level design. Higher level models are used to prove the topology and component values of the implementation are correct and even higher level models, often compiled from C language code, are used to determine the system performance over different input levels. Proving that a schematic is a correct implementation of the desired function is very important, yet surprisingly difficult, and this is addressed in Chapter 5.

1.5 PURPOSE OF RESEARCH

This research aims to provide techniques to improve performance and cut the implementation cost of sigma-delta modulators used in communications applications. Individual conversion components in communications applications are often only loosely specified since it is the overall system performance which is tightly specified.

Often it is possible to exploit this by using advanced system level design methods to adapt and tailor the performance to that required. Rather than specifying requirements in terms of signal to noise ratio (SNR), out-of-band noise, distortion, phase-linearity and bandwidth, it is often the requirement to meet specifications of in-band signal to noise ratio, bit-error rate, adjacent channel rejection, spurious-free dynamic range, phase error and vector error. A system which is optimised for these parameters will often be simpler and cheaper than a system designed to meet more conventional AC specifications.

Designing real sigma-delta modulators often requires manual optimisation of certain characteristics which cannot be incorporated into a fully computer-automated procedure. Design software for sigma-delta modulators needs to be sufficiently flexible to allow a designer to manually optimise, even if this mandates designer intervention throughout the process. Flexibility is often more important than design automation incorporating prior knowledge - the designer must be required to provide the knowledge and the ideas. This contrasts the situation with filter design where standard filter design procedures may be completely automated with the software making multiple design decisions which the designer may not be aware of and need not understand. This difference is the result of the relative youth of sigma-delta modulator knowledge and the fact that the mathematics behind sigma-delta modulators is intractable and approximations must be made which are often barely adequate.

1.6 STATEMENT OF ORIGINALITY

The following papers were published during the course of this research:

[i] D.M. Hossack and J.I. Sewell, "A Method for the Evaluation of Multirate Sigma-Delta Systems," *IEEE International Symposium on Circuits and Systems*, San Diego, California, pp. 1328-1331, May 1992.

[ii] D.M. Hossack and J.I. Sewell, "Computer-Aided Design of High Order Sigma-Delta ADCs," *IEE Colloquium on Advanced A-D and D-A Conversion Techniques and Applications*, London, UK, Digest No: 1993/128, May 1993.

[iii] D.M. Hossack and J.I. Sewell, "Design of High Order Audio Sigma-Delta Modulators with Minimum Weighted Noise," *IEEE International Symposium on Circuits and Systems*, Chicago, Illinois, pp. 180-183, May 1993.

[iv] D.M. Hossack, J.I. Sewell and J.R.C. Reid, "Issues in the Design of Low Oversampling Ratio Single Bit Sigma-Delta Modulators," *IEE Colloquium on Oversampling Techniques and Sigma-Delta Modulation*, London, UK, Digest No. 1994/083, March 1994.

[v] D.M. Hossack and J.I. Sewell, "The Application of Redundant Number Systems to Digital Sigma-Delta Modulators," *IEEE International Symposium on Circuits and Systems*, London, UK, pp. 481-484, May 1994.

The following results of the research in this thesis are, as far as is known, original:

In Chapter 2, the use of least squares methods to minimise weighted noise in high order sigma-delta modulators is demonstrated with the application to minimise audible noise and to exploit this to allow reduction in oversampling ratio. The result that the simpler classical Butterworth high pass noise transfer function exhibits lower audible noise than elliptic and inverse Chebyshev designs. The requirement to dither high order modulators is noted.

The suitability of using redundant number arithmetic in the implementation of digital sigma-delta modulators is demonstrated in Chapter 3, allowing arbitrarily high accuracy and very high speed digital sigma-delta modulation without any word-length / speed trade-off. This high speed sigma-delta modulator can be used to perform a high speed FIR filter / DAC combination. The advantage of the FIRDAC approach for reducing the effect of performance degradation arising from noise and interference on the reference making it a suitable technique for very high speed sigma-delta DACs.

The improved topology of the sigma-delta based compander discussed in Chapter 4 and the implementation using mixed signal techniques to ensure robust performance are the subjects of a patent application.

The method for reduction of switched capacitor circuit schematics to simple algebraic equations method described in Chapter 5 allows order of magnitude speed improvements over existing circuit verification and simulation methods. The use of the C++ operator overloading capability allow bit-level accurate digital simulations to be concisely and efficiently coded in a standard computer language.

1.7 REFERENCES

[1] P.M. Aziz, H.V. Sorensen, J.Van Der Spiegel, "An Overview of Sigma-Delta Converters," *IEEE Signal Processing Magazine*, pp. 61-84, January 1996.

[2] J.C. Candy and G.C. Temes (editors), *Oversampling Delta-Sigma Data Converters*, IEEE Press, Piscataway, New Jersey, 1992.

[3] Y. Tsividis and P. Antognetti (Editors), *Design of MOS VLSI Circuits for Telecomunications*, Pretice-Hall, Englewood Cliffs, New Jersey, 1985.

[4] B. Leung, "Theory of sigma-delta analog to digital converter," *IEEE International Symposium on Circuits and Systems Tutorials*, IEEE Press, Piscataway, New Jersey, pp. 196-223, May 1994.

[5] S.R. Norworthy, R. Schreier, G.C. Temes, *Delta-Sigma Data Converters, Theory Design, and Simulation*, IEEE Press, Pitscataway, New Jersey, 1996.

[6] Inose and Y. Yasuda, "A Unity Bit Coding Method by Negative Feedback," *Proceedings of the IEEE*, pp. 1524-1535, November 1963.

[7] H.A. Sprang and P.M. Schultheiss, "Reduction of Quantizing Noise by Use of Feedback," *IRE Transactions on Communications Systems*, pp. 373-380, December 1962.

[8] T.A.D. Riley, M.A. Copeland, and T.A. Kwasniewski, "Delta-Sigma Modulation in Fractional-N Frequency Synthesis, *IEEE Journal of Solid-State Circuits,* vol. 28, no. 5, pp. 553-559, May 1993.

[9] T. Kwan, R. Adams, and R. Libert, "A Stero Multibit ΣΔ DAC with Asynchonous Master Clock Interface," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, pp. 1881-1887, December 1996.

[10] N.C. Battersby, *Switched-Current Techniques for Analogue Sampled-Data Signal Processing*, PhD Thesis, Imperial College of Science, Technology and Medicine, University of London, 1993.

[11] N. Tan, *Oversampling A/D Converters and Current-Mode Techniques*, PhD Thesis, Linkoping University, Linkoping, Sweden, 1994.

[12] J.C. Candy, "Decimation for Sigma Delta Modulation," *IEEE Transactions on Communications*, vol. 34, pp 72-76, January 1986.

[13] R.E. Chochiere and L.R. Rabiner, "Interpolation and Decimation of Digital Signals - A Tutorial Review," *IEEE Proceedings*, vol. 69, pp. 300-331, March 1981.

[14] Y.C. Lim, "Design of Discrete-Coefficient-Value Linear Phase FIR Filters with Optimized Normalized Peak Ripple Magnitude," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 12, pp. 1480-1486, December 1990

[15] R.T. Baird and T.S. Fiez, "Improved ΣΔ DAC Linearity using Data Weighted Averaging," *IEEE International Symposium on Circuits and Systems*, Seattle, Washington, pp. 13-16, May 1995.

[16] J.D. Gibson (Editor), *The Mobile Communications Handbook*, CRC Press / IEEE Press, 1996.

[17] B.E. Boser and B.A. Wooley, "The Design of Sigma-Delta Modulation Analog-to-Digital Converters," *IEEE Journal of solid-state Circuits*, vol. 23, pp. 1298-1308, December 1988.

[18] K. Chao, S. Nadeem, W. Lee, and C. Sodini, "A Higher Order Topology for Interpolative Modulators for Oversampling A/D Converters," *IEEE Transactions on Circuits and Systems*, pp. 890-893, March 1990.

[19] P. Ferguson, A. Ganesan, R. Adams, "One Bit Higher Order Sigma-Delta A/D Converters," *IEEE International Symposium on Circuits and Systems*, New Orleans, Louisiana, pp309-318, March 1990.

[20] R. Schreier and M. Snelgrove, "Bandpass Sigma-Delta Analog-to-Digital Conversion," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 11, pp. 1406-1409, November, 1991.

2

DESIGN OF HIGH ORDER SIGMA-DELTA MODULATORS

2.1 INTRODUCTION

In this chapter the design of high order single bit sigma-delta modulators is discussed. The determination of stable noise transfer functions (NTF) is described using a frequency weighted approximation method to minimise the effect of quantisation noise. The design of the signal transfer function (STF) is then discussed including the possibility of introducing pre-emphasis or signal frequency shaping to give a frequency dependent overload characteristic.

The approximation techniques for optimised signal transfer functions and the noise transfer functions give, in many applications, better performance than classical functions. These functions are then used to determine the coefficients required for a digital implementation or the capacitor values required for a switched capacitor network.

Two examples of the design methods are presented. The first considers the design of sigma-delta modulators specifically for audio applications, where the noise transfer function may be optimised to be minimally audible under the very non-flat response of the ear. The second example is for a telephone modem analogue to digital converter where the design of the loop filter can be made to act as an input filter and gain stage while simultaneously providing noise shaping. This eliminates the requirement for a high performance filter and gain stage before the analogue to digital conversion.

2.2 TRANSFER FUNCTION DESIGN

The design procedure starts by determining suitable transfer functions for the loop filter.

The signal transfer function (STF) and the noise transfer function (NTF) must first be designed subject to a number of constraints [1]. These transfer functions are then used to design the coefficients for the desired topology used to implement the loop filter. The transfer functions may be optimised to improve performance and to make the modulator simpler to implement for both analogue and digital systems. In particular, the noise transfer function may be designed to have minimal noise present over the frequencies where the application is most sensitive to noise. Where the modulator is being used to process audio information, the noise transfer function should be designed to have least noise over the frequency around 3kHz where the human ear is most sensitive [6]. This improves the perceived noise performance of the modulator. For the case of processing data signals, the modulator may be optimised to produce less noise in the adjacent signal bands at the cost of increased in-band noise which often is less critical.

The design of the loop filter starts with the determination of a stable denominator. Two numerators are required for this denominator: a high pass function for the NTF and a low-pass function for the STF. These functions can then be realised by a single loop filter with two inputs.

2.2.1 NTF/STF denominator design

The key to the design of stable high order sigma-delta loops is the NTF denominator $D_{ntf}(z)$. The poles must be chosen to limit the high frequency gain to be around 3 dB [1]. It is also necessary to ensure that the coefficient of the highest powers of z in both the numerator and the denominator are unity - this ensures that the sigma-delta loop is not delay free [2]. A convenient way to satisfy these requirements is to use Butterworth or Inverse Chebyshev high pass filter pole positions and then by adjusting the cut-off frequency move the poles towards or away from the unit circle until the gain has the desired value. A simple computerised iteration process can be used to achieve this. In practice a few manually invoked iterations using a program such as MATLAB [3,4] or TFN (Appendix 1) is all that is required.

2.2.2 NTF numerator design

The zeros of the NTF can be chosen almost independently from the poles because, for most over-sampling ratios, moving the zeros through the signal band has very little effect on the response in the noise band and consequently, on stability. Rather than using classical approximations (either Butterworth with all zeros at DC, or inverse Chebyshev with an equiripple noise response), the method adopted here involves placing the zeros so as to minimise the frequency weighted effect of the shaped quantisation noise. As noted in [5], even if the noise weight function is flat, the inverse Chebyshev response does not give the minimum noise integrated over the signal band. A good example of improved performance with a non-flat weighting curve is the design of sigma-delta modulators for audio applications. Figure 2-1 shows the typical response of the ear to low level sounds (15-phon) [6]. The sensitivity of the ear varies with sound amplitude and the optimisation is performed for the case of small amplitude signals where additive noise will be the most apparent [7].

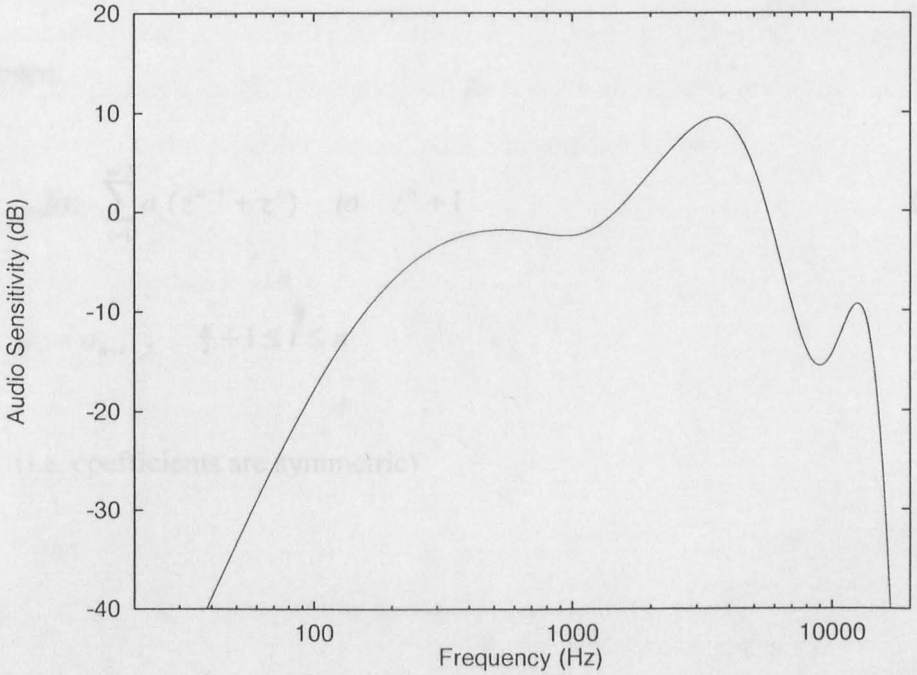


Figure 2-1 The sensitivity of the ear to low level sounds.

We want to minimise:

$$\int_{\substack{signal\\band}} |W(z)|^2 \frac{|N_{ntf}(z)|^2}{|D_{ntf}(z)|^2} dz \quad , \qquad (z = e^{jw}) \tag{2-3}$$

with

$$N_{ntf}(z) = \sum_{i=0}^{i=n} a_i z^i \quad , \qquad a_n = 1 \tag{2-4}$$

and $D_{ntf}(z)$ is the stable denominator and $W(z)$ is the chosen weighting function. This is equivalent to fitting $\sum_{i=1}^{i=n} a_i z^i$ to the function z^n over the signal band, with a frequency dependent weighting function. This can be achieved by using least squares techniques [8,9]. The order of the problem can be halved by forcing the zeros to lie on the unit circle:

For n even:

$$fit: \sum_{i=1}^{n/2} a_i (z^{n-i} + z^i) \quad to \quad z^n + 1 \tag{2-5}$$

$$a_i = a_{n-i} \quad , \quad \tfrac{n}{2} + 1 \le i \le n$$

(i.e. coefficients are symmetric)

For n odd:

$$fit: \sum_{i=1}^{(n-1)/2} a_i (z^{n-i} - z^i) \quad to \quad z^n - 1 \tag{2-6}$$

$$a_i = -a_{n-i} \ , \quad \tfrac{n+1}{2} \le i \le n$$

(i.e. coefficients are anti-symmetric)

The basis functions are:

$$X_j(z) = z^{n-j} + z^j \qquad \text{n even} \tag{2-7}$$

$$X_j(z) = z^{n-j} - z^j \qquad \text{n odd}$$

The weight function and the desired function are evaluated over a uniformly distributed set of frequencies f_i spanning the full signal band f_0 to f_{N-1} to obtain the vectors w_i and y_i respectively (Equations (2-8) .. (2-11)). If the real decimation filter response is known, then the frequency set f_i can be carried into the noise band to take into account the noise which is aliased back into the signal band.

$$f_i = f_0 + \frac{(f_{N-1} - f_0)}{N-1} i \tag{2-8}$$

$$x_i = e^{j.2.\pi f_i} \qquad \left(j = \sqrt{-1}\right) \tag{2-9}$$

$$w_i = \left|W(x_i)\right| \tag{2-10}$$

$$y_i = x_i^n + 1 \qquad \text{n even} \tag{2-11}$$

$$y_i = x_i^n - 1 \qquad \text{n odd}$$

2.2.3 Direct Solution

The merit function that is to be minimised is the mean-square sum of the weighted error between the desired response and the obtainable responses from the M basis functions $X_k(z)$ sampled at the N points x_i:

$$\varepsilon^2 = \sum_{i=1}^{N}\left[w_i\left(y_i - \sum_{k=1}^{M} a_k X_k(x_i)\right)\right]^2 \qquad (2\text{-}12)$$

The minimum ε^2 occurs when the derivative of (2-12) with respect to all M parameters a_k is zero:

$$0 = \sum_{i=1}^{N} w_i^2\left[y_i - \sum_{j=1}^{M} a_j X_j(x_i)\right]X_k(x_i) \qquad k = 1,\ldots,M \qquad (2\text{-}13)$$

Interchanging order of summation:

$$\sum_{j=1}^{M}\left[\sum_{i=1}^{N} w_i^2 X_j(x_i)X_k(x_i)\right]a_j = \sum_{i=1}^{N} w^2 y_i X_k(x_i) \qquad (2\text{-}14)$$

This represents a set of linear equations. In matrix form:

$\boldsymbol{W}$ is diagonal matrix containing w_i^2

$\boldsymbol{A}$ is the matrix of basis functions $A_{ij} = X_i(x_j)$

$\boldsymbol{b}$ is the vector of the desired function

$\boldsymbol{a}$ is the vector of coefficients (unknown)

$$\varepsilon^2 = (A.a - b)^T.W.(A.a - b) \tag{2-15}$$

with the solution by solving equation (2-16) for a:

$$A.W.A^T.a = A^T.W.b \tag{2-16}$$

This can be simplified by redefining A and b as being the weighted versions:

$$A_{ij} = w_j X_i(x_j) \tag{2-17}$$

$$b_i = w_i y_i \tag{2-18}$$

$$(A^T.A).a = A^T.b \tag{2-19}$$

The set of equations given by equation. (2-19) are known as the normal equations. Direct solution from the normal equations can be numerically problematic.

2.2.4 Transformed Least Squares Noise Minimisation

Direct solution of the least squares problem (2-19) is numerically ill-conditioned. A better method is the singular value decomposition method [8]. Other advanced methods are also available [10]. The minimisation problem of (2-19) is ill-conditioned since the weight functions are only non-zero over a small range of frequencies and the basis functions have quite similar shape over this small range. It is better to solve the minimisation problem over a wider range of z-domain frequencies and then transform the resulting solution. This may be achieved by using a bilinear transformation of the form [11]:

$$z \rightarrow \frac{z - \alpha}{1 - \alpha z} \tag{2-20}$$

This maps the z-domain unit circle on to itself. The factor α determines how the warping of frequencies occurs. The frequency ω transforms to frequency ω' for the value

$$\alpha = \frac{\sin\frac{(\omega - \omega')}{2}}{\sin\frac{(\omega + \omega')}{2}} \qquad (2\text{-}21)$$

The typical bandwidth of the signal band is of the order of 0.01 for a normalised sample rate of 1. The minimisation problem is transformed so that the band edge frequency of 0.01 becomes (say) 0.25 with the result that the whole noise band from 0.01 to 0.5 would become compressed into 0.25 to 0.5. Since the transformation (2-20) introduces n poles of the form $(1-\alpha z)$ for an n^{th} order transfer function, the problem must be pre-warped to cancel this effect.

2.2.5 STF Design

Only the numerator of the STF can be designed independently from the NTF. A number of possibilities are available. Once choice is a constant (or a constant multiplied by a power of z^{-1}). This gives a low pass response with a flat response through out the signal band with attenuation at high frequencies. This also has a structural advantage in that the input signal needs to be applied only to a single point in loop filter and not to multiple nodes. Thus the realisation is simpler for both the analogue and digital implementations. An alternate choice of STF is to place the zeros on top of the poles (which are the same as the NTF poles). This is seen to reduce the coefficient spread after scaling and has the advantage of quicker recovery from out-of-range inputs [12]. However unless an additional summer is used, the order of the numerator is restricted to be one less than the denominator (for the cascaded resonator topology) and so at least one pole must remain uncancelled. For odd orders, the result is a smooth first order low-pass STF with cut-off well beyond the signal band.

Alternatively, the STF may be modified to include some kind of pre-emphasis. For audio signals, this can give an improvement by better matching the dynamic range as a function of the frequency of the audio source material to the ADC. Typically less dynamic range is required at high frequencies and this can be traded for more dynamic range at low frequencies [19]. The pre-emphasis can be removed by the decimation filter to give an overall flat frequency response, but with a frequency dependent overload range. It is important to also consider the time-domain response from the input to the comparator, since transient overshoot can take the modulator beyond its stable region. Thus the amount of pre-emphasis that can be applied in practice is limited.

A second reason for choosing an unusual STF is to reduce the susceptibility of the modulator being made unstable by a large unwanted signal. The STF is designed to have high attenuation at frequencies where unwanted signals may occur. This is illustrated later in this chapter for a modem example.

2.3 LOOP FILTER REALISATION

For the case of analogue to digital converters it is possible to directly convert the NTF and STF into the capacitance values used in the cascaded switched capacitor (SC) resonator topology . The loop filter is formed by a cascade of the first and second order sections shown in Figure 2-2 [2]. Each section has a cascade input, and inputs from the input signal and the comparator feedback signal. A similar topology can also be used for digital modulators. The capacitor values could be determined by solving the difference equations representing the network to determine the transfer function in terms of the capacitor values and then solving the sets of equations arising by equating coefficients of z. This is cumbersome for higher orders and is not amenable to automated design by computer. A general design procedure is described here.

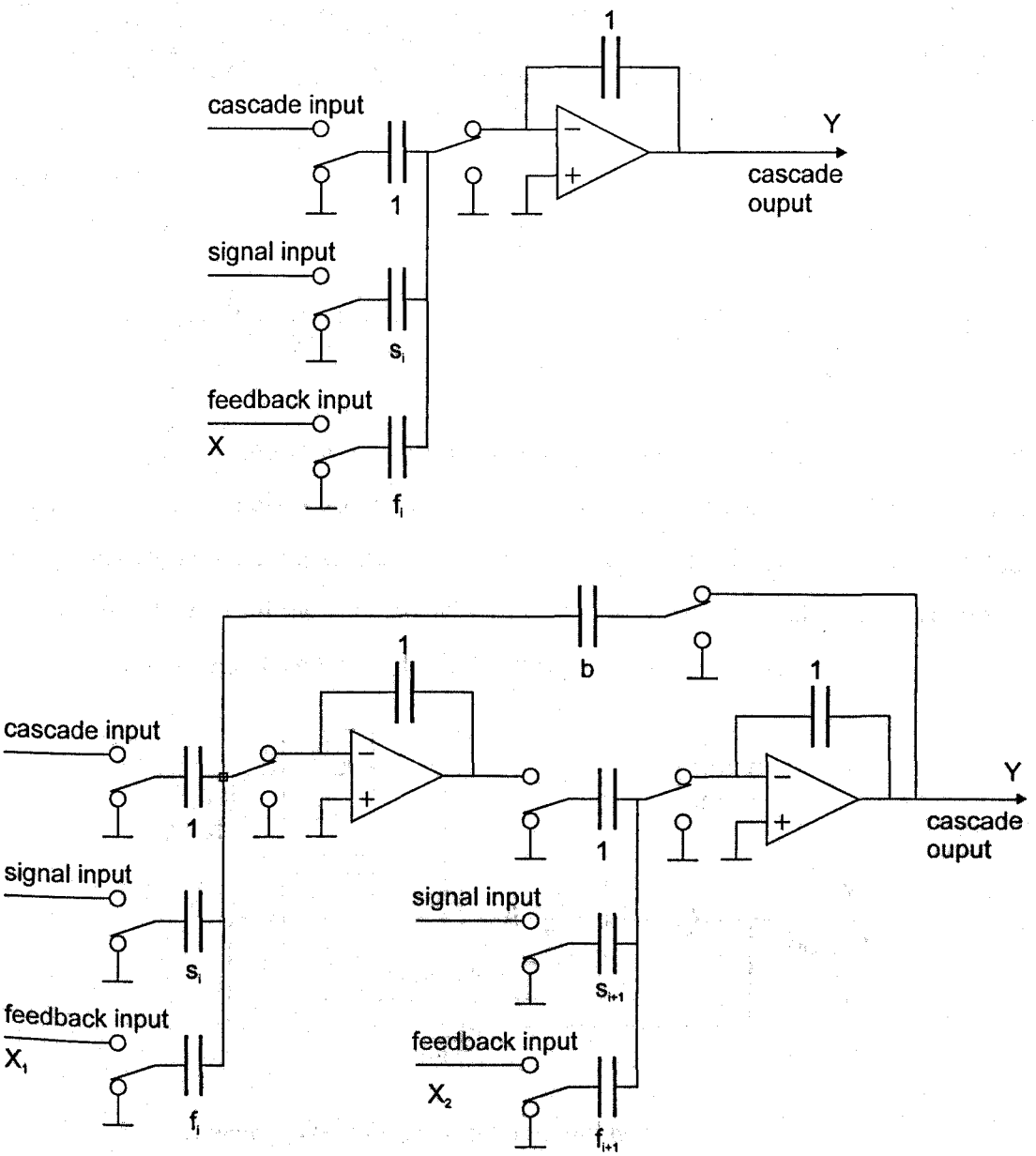


Figure 2-2 The first and second order SC sections used to construct high order loop filters.

For the first order section in Figure 2-2:

$$\frac{Y}{X} = \frac{f_i z^{-1}}{1 - z^{-1}} \tag{2-22}$$

and for the second order section:

$$\frac{Y}{X_1} = \frac{f_i z^{-2}}{1+(b-2)z^{-1}+z^{-2}} \tag{2-23}$$

and:

$$\frac{Y}{X_2} = \frac{f_{i+1} z^{-1}(1-z^{-1})}{1+(b-2)z^{-1}+z^{-2}} \tag{2-24}$$

As only capacitor ratios are required, the integrating capacitors and the coupling capacitors are initially chosen as unit capacitances, leaving only the resonance capacitors (b) and the signal and the feedback feed-in capacitors unknown (s_i and f_i). The final values of all the capacitances will be determined by scaling the network to limit maximum signal swings and minimise capacitances.

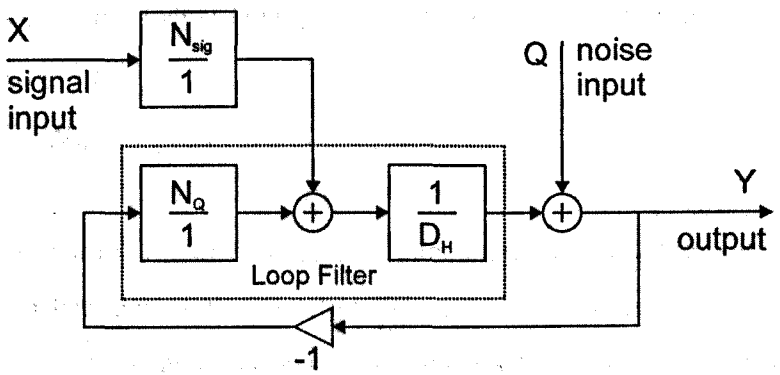


Figure 2-3 Linear model of sigma-delta modulator.

A linear model of the sigma-delta modulator is shown in Figure 2-3. Equations for the noise transfer function (NTF) and the signal transfer function can be obtained:

$$NTF = \frac{N_{ntf}}{D_{ntf}} = \frac{Y}{Q} = \frac{D_H}{D_H + N_Q} \quad \Rightarrow \quad \frac{N_Q}{D_H} = \frac{D_{ntf} - N_{ntf}}{N_{ntf}} \tag{2-25}$$

and:

$$STF = \frac{N_{stf}}{D_{stf}} = \frac{Y}{X} = \frac{N_{sig}}{D_H + N_Q} \quad \Rightarrow \quad N_{sig} = N_{stf} \tag{2-26}$$

Thus from (2-25), the zeros of the NTF form the poles of the sigma-delta loop filter. The zeros are easily obtained from the NTF coefficients because they are known to lie on the unit circle. The loop filter therefore has poles on the unit circle and so is not stable and the response is unlike standard filter responses. The resonance capacitors (*b* in Figure 2-2) can be determined directly by equating coefficients of the NTF numerator (N_{ntf}), and the denominator of (2-24). As in ordinary cascaded biquaduadratic filter design, there is a choice over section ordering Experiments have shown that implementing the lowest frequency zeros toward the input lowers the final capacitance spread.

The feedback capacitor values (f_i) can be obtained from the solution of:

$$A.f = N_Q \tag{2-27}$$

where f is the column vector of unknown capacitances, N_Q is the column vector of desired loop filter numerator coefficients and the columns of A contain the coefficients of the transfer functions from each feedback input to the output which are now known, as the values of the *b* coefficients have already been determined. The transfer function coefficients for A are easily determined from the representation of the loop filter given in Figure 2-4. Assuming the NTF denominator has the highest power of z equal to unity, the loop filter will contain the necessary one sample delay. Thus the n^{th} order loop filter transfer function has a numerator with only n non-zero coefficients as opposed to $n+1$ for a general n^{th} order polynomial. These n coefficients are used to determine the n unknown feed-in capacitors (f_i) by the set of equations (2-27). Since from equation (2-26), $N_{stf} = N_{sig}$, the same approach can be used to obtain the input capacitance values (s_i).

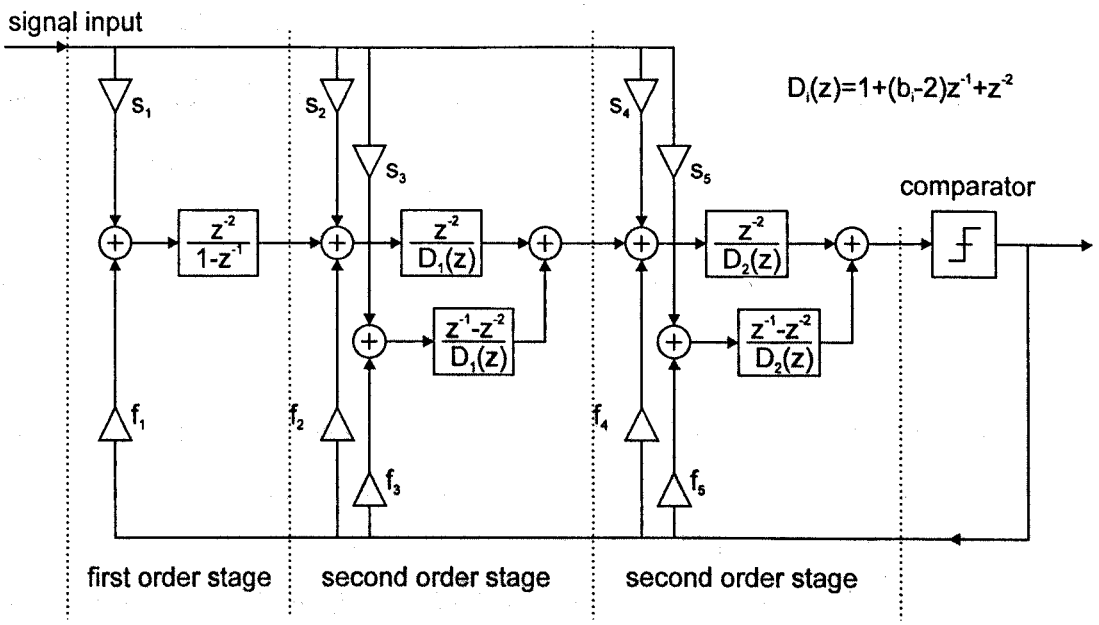


Figure 2-4 Simplified representation of 5th order loop filter used for determining coefficients.

The system must then be simulated in the time domain and capacitors scaled to limit the maximum internal signal swings. This simulation is performed at the highest level possible and is often a compiled C program, since many millions of input samples should be used to ensure that the worst case signal swings are determined [12]. It is possible to reduce capacitance spread by reducing some of the internal signal swings [13]. Indeed it has been proposed to deliberately do this in order that analogue circuit noise acts as a dither source [14]. While in the case of ordinary SC filters this scaling is at the expense of dynamic range, the action of the noise shaping serves to greatly reduce the effect of thermal noise from the integrators and switches towards the comparator. Only the noise from the first op-amp and the switches around it (and to a lesser extent, the second) are critical to the performance of the complete modulator. Consequently useful reductions in component spread may be made with little impact on SNR.

This whole procedure (including simulating for the purposes of scaling) has been automated and can convert NTFs and STFs of any order into SC loop filters. The output files are suitable for direct analysis and simulation by the SCNAP suite

[15,16], SWITCAP [17] and a discrete time simulator [18]. The sigma-delta modulator topology and the design method discussed here are also applicable for band-pass systems.

2.4 EXAMPLE - PSYCHO-ACOUSTIC NOISE SHAPING

When sigma-delta modulators are to be used for audio applications, the performance aim is to introduce the least amount of *audible* artefacts rather than specifying a signal-to-noise ratio over a particular bandwidth. Using the methods above it is possible to design sigma-delta modulators which shape the noise to have minimum audible power [19]. This concept has also been reported in [20], and developed further in [21].

The noise transfer function shapes the quantisation noise to be least audible to the ear. Figure 2-1 shows the frequency-weighting curve used which is referred to as the F-curve, and was derived from ISO data [7]. This curve can be described as a s-domain transfer function in the TFN format as follows:

```
gain 1k -2.515dB
freq_scale 1rad 1k
{
     cascade 3  1:root 0
     2:root -0.58 1.03
     cascade 3  2:root -3.18 8.75
     / {
          cascade 3  1:root -0.18
          cascade 2  1:root -1.63
          cascade 4  2:root -2.51 3.85
          cascade 20 2:root -6.62 14.29
          }
     }
```

The example is a high order, low over-sampling ratio (32) modulator. This was chosen to investigate the potential performance of very low over-sampling one bit modulators. The bit rate at the output of this modulator is only a factor of two more than ordinary 16 bit pulse code modulation (PCM).

The optimisation is performed by a C program which outputs the resulting NTF in TFN format as follows:

```
{
fclk 1.536e+06 {
     rm_constant
     zeros_of
     transform {
          factorise      6:    1
                               -3.39587
                               6.22577
                               -7.44375
                               6.22577
                               -3.39587
                               1
               / 1 }
          { 1: 1 -0.939063
               / 1: -0.939063 1 }
          }
     file tf_denom
     }
```

The result of the least squares minimisation is the six coefficients forming the numerator. Since this is the solution of the transformed problem, the result must be corrected by applying the transformation (2-20) and removing the 6 poles resulting from the this transformation. The transformation is applied using the "transform" facility, and the poles are removed by using "zeros_of." The system poles are then read in from the file "tf_denom" resulting in the final 6^{th} order transfer function (both numerator and denominator are 6^{th} order.) The "rm_constant" statement multiplies by whatever constant is required to ensure that the resulting loop filter has a unit delay in it so that the modulator can be realised. The response of the optimal NTF is compared with the classical Chebyshev response in Figure 2-5 and Figure 2-6.

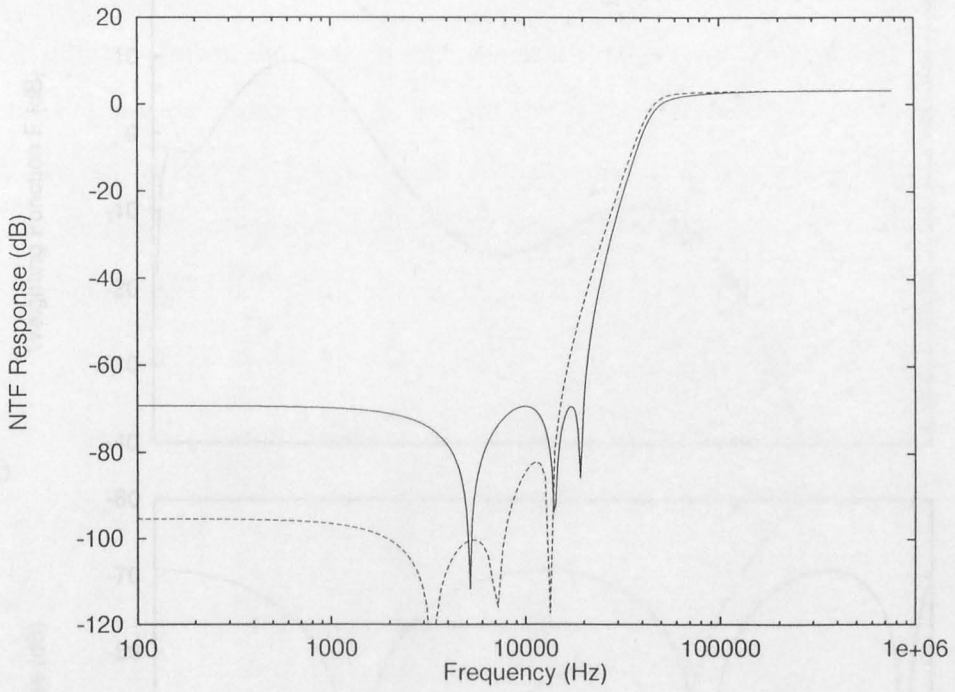


Figure 2-5. The classical NTF (solid) compared to the optimised transfer function (dashed).

The improvement in weighted SNR due to optimum placing of NTF zeros compared with the inverse Chebyshev design can be measured by evaluating the squared error equation (2-12). In this case, the optimised placement resulted in a 10 dB improvement in weighted SNR. Further improvement can be achieved by adding the pre-emphasis to the STF [19].

A comparison of the classical and optimised NTFs over the audio frequency range is illustrated in Figure 2-6. Figure 2-6a shows that the weighting function decreases rapidly at frequencies greater than 15kHz, reflecting the insensitivity of the ear at these frequencies. Figure 2-6b shows that the equiripple response of the classical function gives a higher level of shaped quantisation noise than the optimised NTF for frequencies up to 15kHz. The optimised NTF has a lower NTF response over these frequencies at the cost of a higher response at the high audio frequencies where the ear is very sensitive. The weighted curves in Figure 2-6c are obtained by multiplying the curves of Figure 2-6b by the weighting function (Figure 2-6a). This shows that total weighted power in the audio band is reduced when the optimised NTF is used.

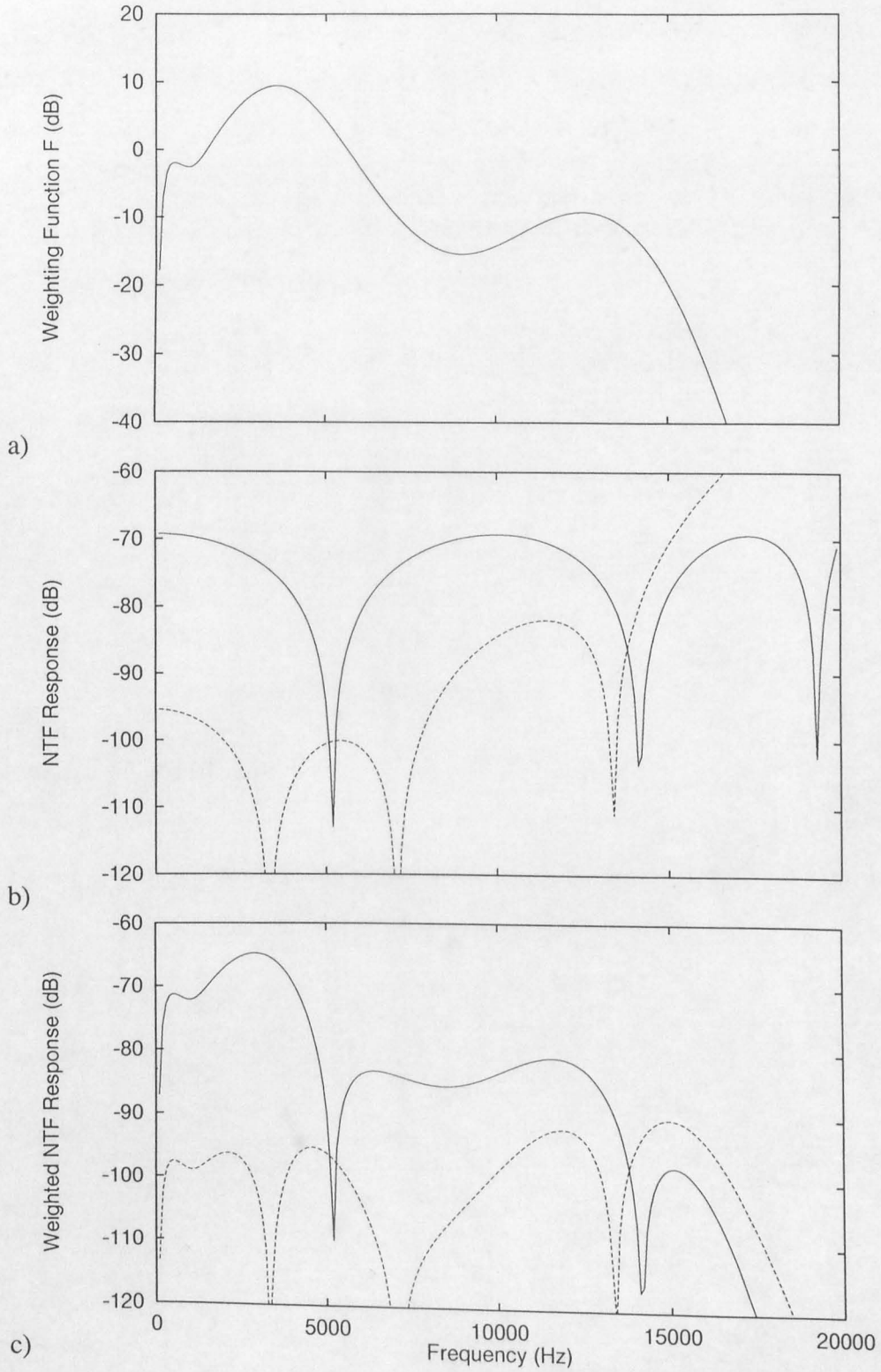


Figure 2-6 Magnitude of weight function and response of NTF (unweighted and weighted) for both the optimised NTF (solid line) and the classical function (dashed line).

The weighted responses of two classical NTFs are shown in Figure 2-7. Of the two classical designs shown, the Butterworth one (all NTF zeros at DC) performs better under the F-curve weighting than the inverse Chebyshev design. This is because the noise power in the Butterworth case is most significant at the high audio frequencies where the ear is very insensitive, and has less power in the low audio frequency range where the ear is most sensitive.

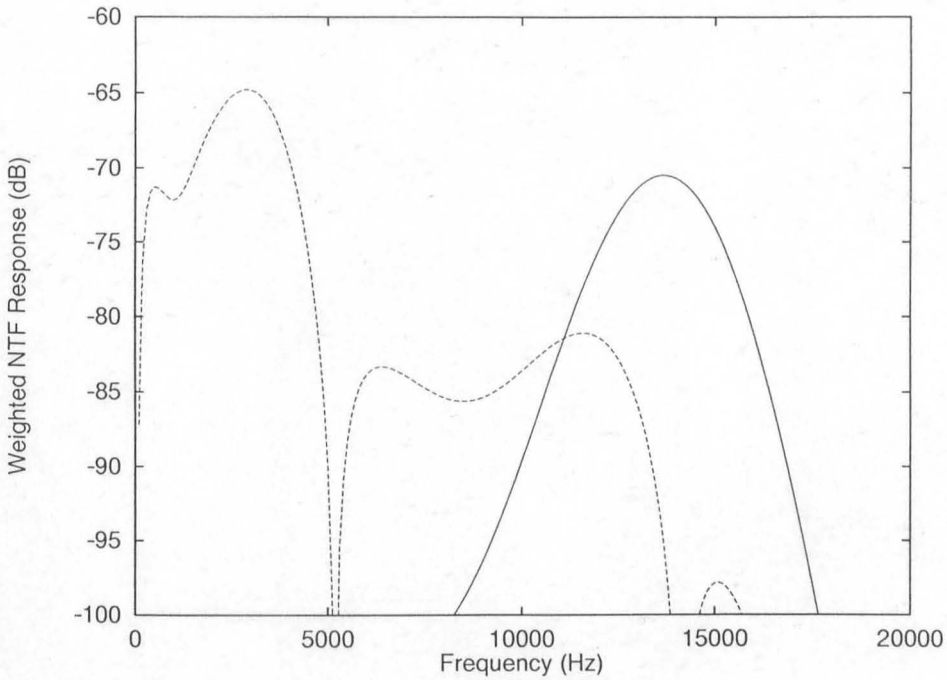


Figure 2-7 Comparison of Butterworth (solid) and Inverse Chebyshev NTF (dashed) responses under a F-curve weighting.

The STFs were given a DC gain of -6 dB so that the modulator becomes unstable for inputs above 0 dB (0 dB defined to be power in a sinusoid with peak amplitude equal to the quantiser output.) The onset of instability can be detected and with proper design it is possible to limit multiple reset cycles from the output to give a cleanly overloading system [2].

2.5 EXAMPLE - MODEM

The previous example illustrated the design and optimisation of the NTF. It is also possible to achieve performance gains through careful design of the STF. For the audio example, pre-emphasis may be used to improve SNR for high signal frequencies at the expense of reduced overload range for these high frequency signals [18].

Optimisation of the STF is limited since because the circuitry implementing the STF is common to that implementing the NTF, only the zeros of the STF may be designed independently. This optimisation step is often performed manually.

In this example, a fifth order sigma delta modulator was designed for use as the analogue to digital conversion for a modem. A typical modem system is shown in Figure 2-8. The requirement for the transmit section (the DAC), is significantly less than the requirements for the receive side (the ADC). High dynamic range is required of the ADC since the received signal is often corrupted by cross-talk from the transmitted signal (near-end cross-talk) and by reflections (echoes) from the far end of the channel. The digital signal processing in the modem must adaptively equalise to the line conditions in order to provide the maximum possible data transfer rate with minimal bit errors. Signalling tones which can be present during the telephone call for metering purposes can interfere with the modem signal if they are of sufficient amplitude to cause limiting in the analogue to digital conversion. Since the tones can be significantly larger than the modem signal they must first be removed by a filter and then the signal may be amplified to allow higher resolution analogue to digital conversion. This filter must have extremely low noise and high linearity since the output is amplified and applied to an ADC with typically 14 bit performance.

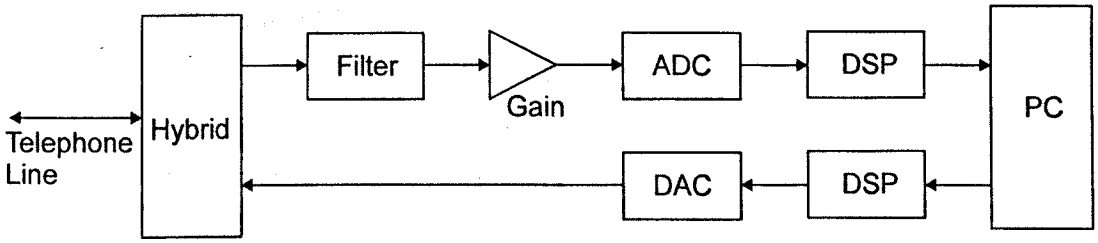


Figure 2-8 Modem System

Using a sigma-delta modulator based ADC gives the advantages of not requiring accurately matched components, and reducing the requirements of the anti-alias filter. However in this case it does not immediately remove the requirement of the filter in front of the ADC since it serves to attenuate large out-of-band signals. The filter and gain stage may be eliminated by merging it with the sigma-delta modulator. This is achieved by designing the signal transfer function to have gain over the 300-4000Hz telephone bandwidth and to have high rejection at the frequencies where signalling tones may be present. The simultaneous optimisation of the STF and NTF can be performed manually.

The NTF was designed to have low noise over the full telephone bandwidth (0-4kHz). The NTF poles were chosen to have as low as possible frequencies since these poles are also the poles of the STF. Low frequency poles imply a low frequency cut-off and greater out-of-band attenuation for the STF.

The NTF in TFN format is:

```
rm_constant
bilinear 4k 4k 480k {
     highpass 1rad 4k inv_cheby 5 -80dB
     }
```

The STF in TFN format is:

```
gain 0 20dB {
     matched_z 480k {
          2:fo 12k
          2:fo 16k
          / 1
          }
     1 / poles_of {
               rm_constant
               bilinear 4k 4k 480k {
                    highpass 1rad 4k
                         inv_cheby 5 -80dB
                    }
               }
     }
```

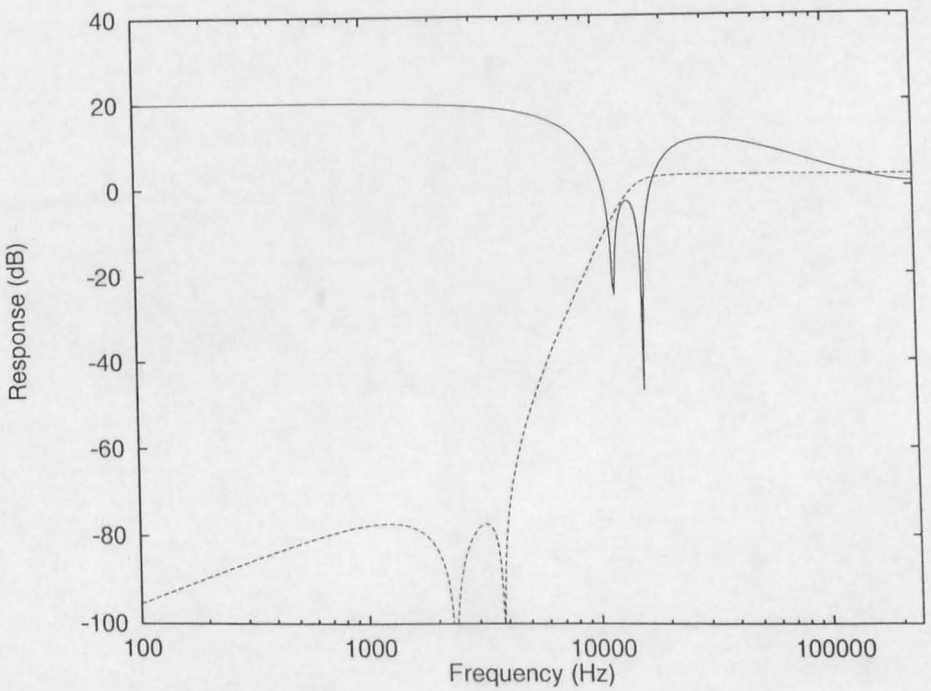


Figure 2-9 Response of the STF (solid line) and the NTF (dashed line).

The signal transfer function was formed by using the poles of the NTF and adding two zero pairs corresponding to frequencies of 12kHz and 16kHz; the frequencies used in Europe for signalling purposes. The low frequency gain was set to 20dB. This results in the transfer function response plotted in Figure 2-9. There is little in-band droop while between 12kHz and 16kHz there is less than 0dB of gain. Thus small in band signals and large signalling tones may be simultaneously present without the signalling tones overloading the ADC. This is shown in Figure 2-10. The input has been formed from a pseudo-random data sequence modulated and filtered to represent modem data combined with a single tone at 12kHz. For the V32 modem standard the data is modulated and filtered using a root-raised cosine filter [22]. The random data signal was scaled such that the peak value was 0.05 whereas the interfering tone had a peak amplitude ten times larger. This amplitude is five times greater than the maximum DC level before the modulator becomes unstable. The output of the sigma-delta modulator is shown having been band-limited to two different bandwidths. The wider bandwidth shows the modulated signal at an amplitude 20dB greater than the input. Further filtering removes more quantisation noise yielding a clean digitised representation of the original modem signal. The spectrum (Figure 2-11) shows that the power in the tone is insignificant compared to the quantisation noise around 12kHz. Clearly the 12kHz tone has been sufficiently attenuated to keep the modulator stable.

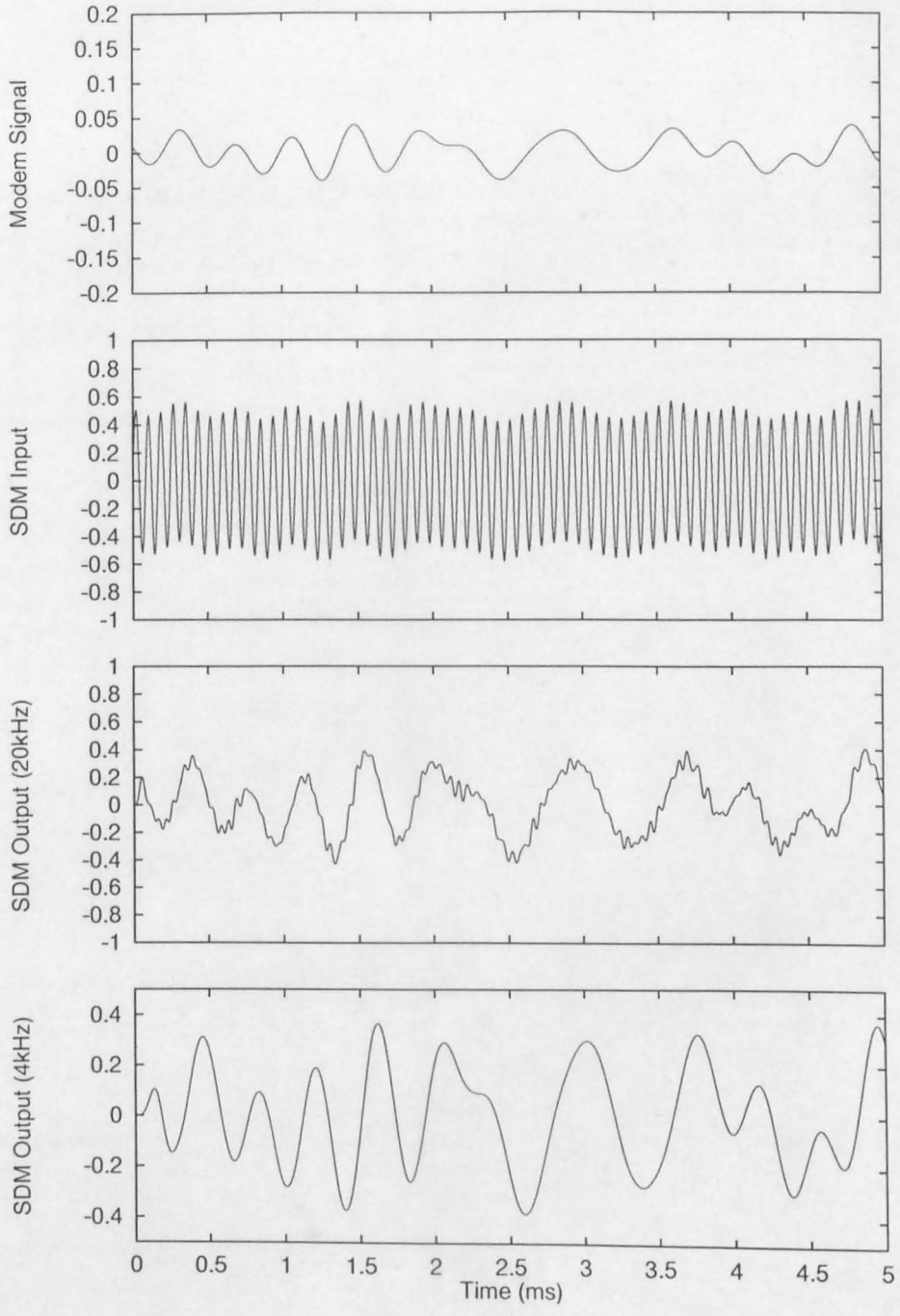


Figure 2-10 The sigma-delta input is dominated by an interfering sinusoid which is rejected by the SDM whilst the base-band signal is amplified

Gain may be applied in a sigma delta modulator simply by reducing the reference voltage [23], or equivalently in SC implementations by reducing the charge fed back by reducing the size of the quantisation feedback capacitors. This is equivalent to applying gain to the input by increasing the input capacitors and re-scaling the capacitors using standard capacitance scaling rules [24].

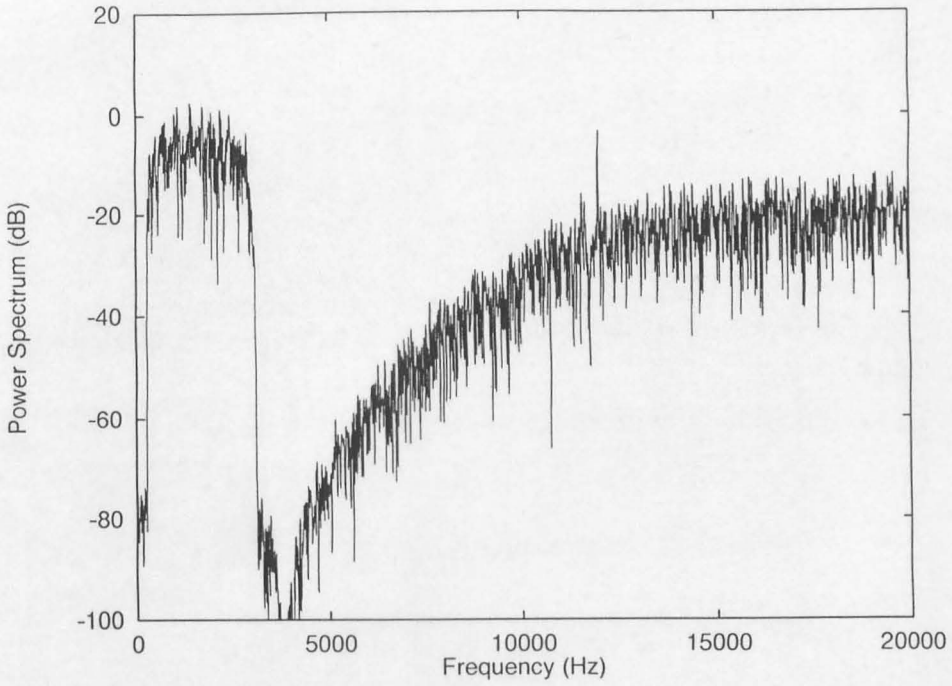


Figure 2-11 Output spectrum of system with modem data and +20dB interference tone at 12kHz.

2.6 SUMMARY

The design of the noise and signal transfer functions for sigma-delta modulators has been discussed and it has been shown that significant improvement can be achieved by shaping the NTF and STF with non-classical functions. The direct transformation of these functions into the favoured cascade-of-resonators topology has also been illustrated. For audio applications significant improvements in weighted noise may be achieved compared to classical transfer functions for the NTF. Of the classical functions the Butterworth response is better than the Inverse-Chebyshev. The modem example demonstrated how an unusual STF design could be used to provide an overall simpler system, in this case saving a filter. The methods should also be applicable to other application areas where a non-flat noise spectrum and/or non-flat overloading characteristics are appropriate.

2.7 REFERENCES

[1] K.C.S. Chao, S. Nadeem, W.L. Lee and C.G. Sodini, "A higher order topology for interpolative modulators for oversampling A/D converters", *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 309-318, March 1990.

[2] R.W. Adams, "Theory and practical implementation of a fifth order sigma-delta A/D converter", *Journal of the Audio Engineering Society*, vol 39, no.7/8, pp. 1515-528, July/August 1991.

[3] J.N. Little and L. Shure, *MATLAB Signal Processing Toolbox - user's guide*, The Mathworks, Inc., 1993.

[4] S.R. Norsworthy, R. Schreier, G.C. Temes, *Delta-Sigma Data Converters Theory, Design, and Simulation*, IEEE Press, Pitscataway, New Jersey, 1996.

[5] P.F. Ferguson, A. Ganesan, R.W. Adams, "One Bit Higher Order Sigma-Delta A/D Converters", *IEEE International Symposium on Circuits and Systems*, New Orleans, Louisiana, pp. 890-893, May 1990.

[6] ISO 131-1979(E), "Acoustics - Expression of Physical and Subjective Magnitudes of Sound or Noise in Air," International Organisation of Standardisation, Geneva, Switzerland, 1979.

[7] R.A. Wannamaker, "Psychoacoustically optimal noise shaping", *Journal of the Audio Engineering Society*, vol. 40, no. 7/8, pp. 611-620, July/August 1992.

[8] W.H. Press, S.A. Teukolsky, B.P. Flannery and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.

[9] W.K. Chen (Editor), *The Circuits and Filters Handbook*, CRC Press, Florida 1995.

[10] B.N. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole, California, 1995.

[11] J.G. Proakis, *Digital Signal Processing* (Second Edition), Macmillan, New York, pp. 648-650, 1992.

[12] R. Adams, "Design Aspects of High-Order Delta-Sigma A/D Converters," *Circuits and Systems Tutorials, IEEE International Symposium on Circuits and Systems,* London, UK, pp. 235-260, May 1994.

[13] R.K. Henderson, *Computer-Aided Design of Switched Capacitor Filters*, PhD Thesis, University of Glasgow 1989.

[14] D.R. Welland, "Method for tone avoidance in delta-sigma converters," US Patent No. 5055846, 1991.

[15] L.B. Wolovitz and J.I. Sewell, "General Analysis of Large Linear Switched Capacitor Networks," *IEE Proceedings*, vol. 135, pt. G, no. 3, pp. 119-124, June 1988.

[16] Z.Q. Shang and J.I. Sewell, *SCNAP4 User's Guide*, Department of Electronics and Electrical Engineering, Glasgow University, December 1993.

[17] *User's Guide for SWITCAP Version 5*, Columbia University, New York, NY, August 1987.

[18] D. Hossack and J.I. Sewell, "A method for the evaluation of multirate sigma-delta systems", *IEEE International Symposium on Circuits and Systems*, San Diego, California, pp. 1328-1331, May 1992.

[19] D. Hossack and J.I. Sewell, "Design of High Order Audio Sigma-Delta Modulators with Minimum Weighted Noise," *IEEE International Symposium on Circuits and Systems*, Chicago, Illinois, pp. 180-183, May 1993.

[20] S. Jantzi, C. Ouslis, and S. Sedra, "Transfer Function Design for Sigma-Delta Converters," *IEEE International Symposium on Circuits and Systems*, London, UK, pp. 433-436, May 1994.

[21] C. Dunn and M. Sandler, "Psychoacoustically Optimal Sigma Delta Modulation," *Audio Engineering Society 99th Convention*, Preprint 4134 (M-2), October 1995.

[22] Jere Babst (editor), *Digital Signal Processing Applications using the ADSP-2100 Family*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[23] D. Kim, J. Park, S. Kim, D. Jeong, W. Kim, "A Single Chip ΣΔ ADC with a Built-In Variable Gain Stage and DAC with a Charge Integrating Subconverter for a 5V 9600-b/s Modem," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 940-943, August 1995.

[24] R. Gregorian and G.C. Temes, *Analog MOS Integrated Circuits for Signal Processing*, Wiley, New York NY, pp. 339-349, 1986.

3

HIGH SPEED DIGITAL SIGMA-DELTA MODULATORS AND RECONSTRUCTION FILTERS

3.1 INTRODUCTION

The previous chapter discussed the use of sigma-delta modulation to implement analogue to digital converters. Sigma-delta modulation is also used for digital to analogue conversion using digital sigma-delta modulators. The PCM input words are interpolated to increase the sample rate and digitally modulated and output through a one bit DAC followed by an analogue reconstruction filter. The reconstruction filter serves to reduce images due to imperfect interpolation and the shaped quantisation noise. Communications applications often have stringent specifications on the out of band power, which requires the use of high performance reconstruction filters to adequately remove the shaped quantisation noise whilst not introducing distortion products which may lie out of signal band. This requires a trade off between the modulator order, the filter order and the sample rate.

The digital modulator is similar to the analogue one in that it takes as input, a finely resolved input level and converts this to a single bit representation. The cascade of integrators and resonators topology as discussed in the previous chapter may be used except that the signal processing is performed digitally. Since the modulator operates on oversampled data, the circuitry used must be fast. Conventional digital pipe-lining techniques cannot be applied to increase throughput due to the recursive nature of the loop. The digital implementation of the sigma-delta modulator must perform all the calculations required to obtain a single output sample before processing on the next sample can start. This dependency of the current output on the immediately previous one limits the application of pipe-lining techniques to speeding up the loop operation. The delay between an input and its corresponding output is termed the latency. For high speed operation it is important to minimise the latency in feedback loops.

Recent work has overcome this latency problem for the case of high speed infinite impulse response (IIR) filters [1]. This has been achieved by adopting a redundant number system instead of the two's complement number system normally used for

digital signal processing applications. This allows the digital part of the modulator to operate considerably faster by removing carry propagate paths and by allowing extensive pipe-lining around the modulator feedback loop.

Since high speed digital sigma-delta modulators are most likely to be used for digital to analogue conversion, the interface between the digital signal and the analogue signal must be considered. The design of one bit DACs is not trivial [2], particularly if operating at very high sample rates. A solution to easing the design of the analogue reconstruction filter with some significant practical advantages has been developed.

3.2 REDUNDANT NUMBER SYSTEMS

In a redundant number system, the individual digits can take on more values than the radix of the number system. Only the signed binary number representation (SBNR) will be discussed here. Valid SBNR digits are -1, 0, and +1 and each digit has a significance of two times that of the digit to its left (as in normal binary). Thus the numbers $\bar{1}0.11$ and $0\bar{1}.\bar{1}1$ both represent the decimal value -1.25. The symbol $\bar{1}$ is used to indicate a weight of -1.

The advantage of the redundant number system is that additions can be performed without long carry propagation chains. Figure 1 shows a SBNR adder. Each adder cell consists of three sub-cells A, B and C which perform an addition of digits with various ranges of values. The final result is in the same SBNR format as the input and the carry/borrows can only propagate a maximum of two digits. Thus the addition time is independent of the word length. A detailed discussion for implementing these cells is given in [1]. The cells are of low complexity and can be implemented using standard logic gates. Two digital lines must be used for each SBNR digit. A schematic for a single digit adder is shown in Figure 3-2.

A disadvantage of SBNR based systems is that normally it is required to convert the result to ordinary two's complement representation and this requires a carry propagate path from the least significant digit (LSD) to the most significant digit (MSD). This is not an issue here since the sigma-delta modulator has a single bit output.

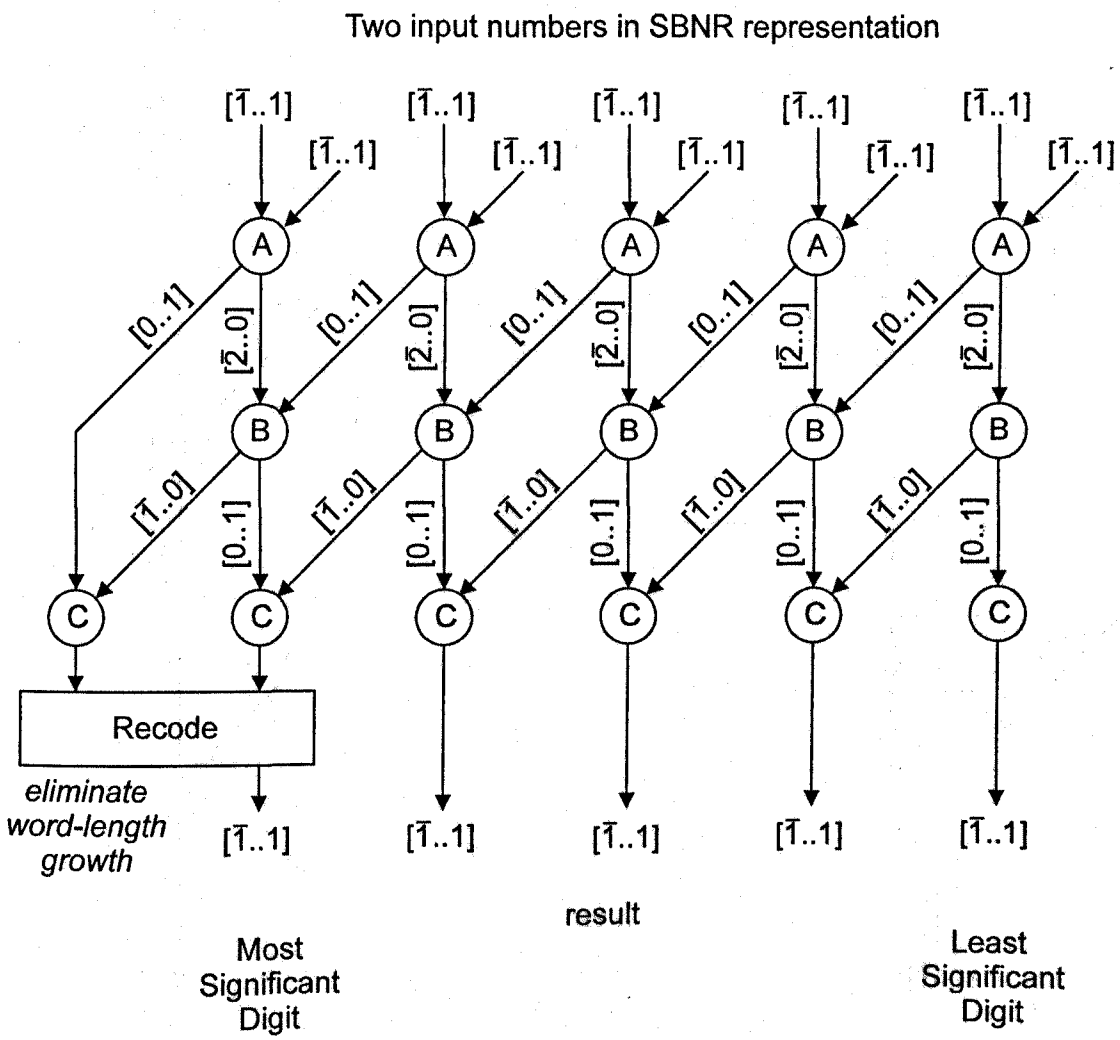


Figure 3-1 SBNR Adder

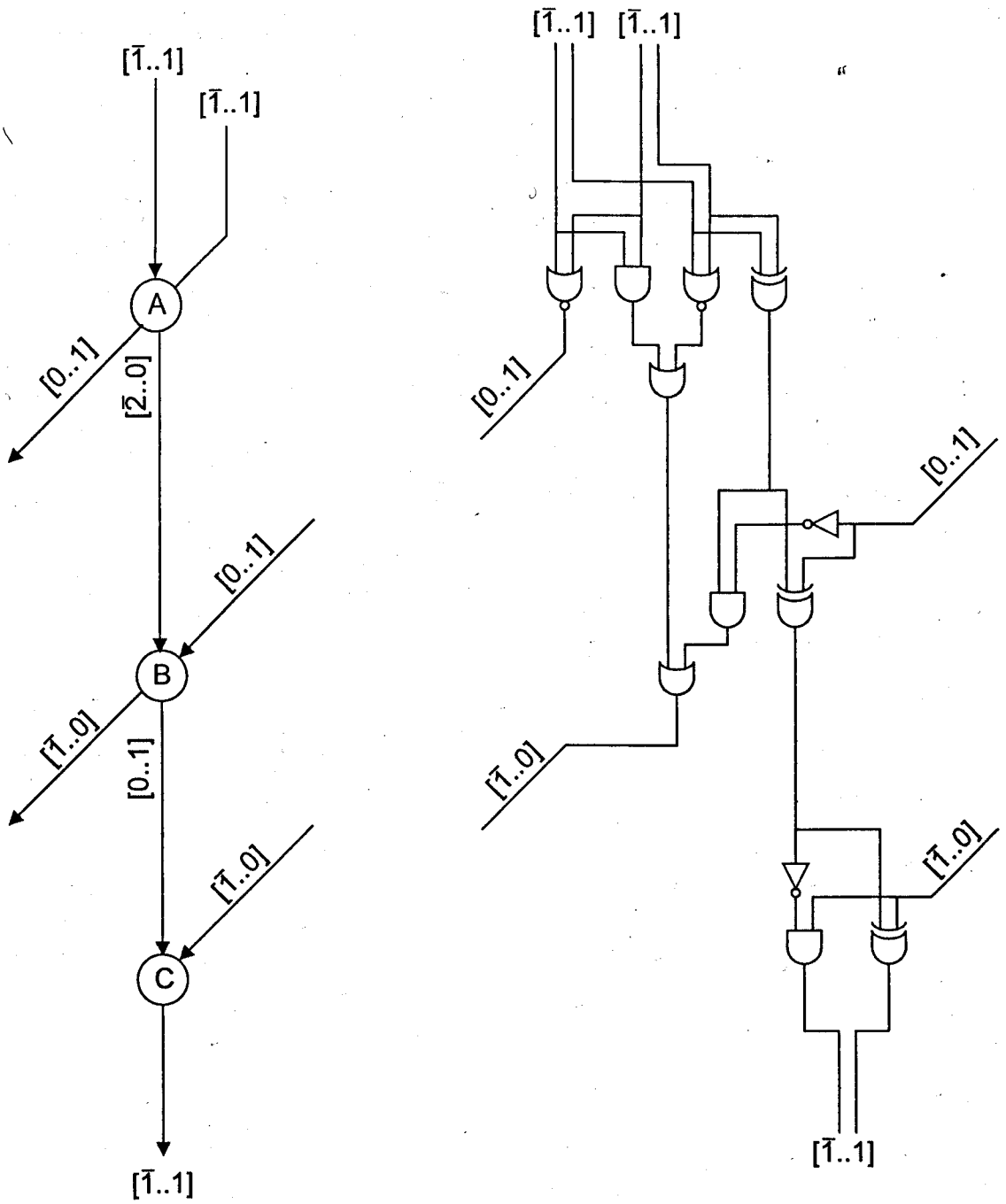


Figure 3-2 Schematic showing an implementation of single digit SBNR adder.

Ideally, in a properly designed sigma-delta modulator, the additions will never overflow, although this is difficult to guarantee. Furthermore, the left most cells may produce carry/borrows even when overflow has not occurred. The cases of the two most significant digits as $\bar{1}1$ and $1\bar{1}$ must be re-coded as the equivalent values of $0\bar{1}$ and 01 to limit word length growth. Definite overflows 11 and $\bar{1}\bar{1}$ and

possible overflows 10 and $\bar{1}0$ must also be caught and the output set to an appropriate saturation level. This is performed by the "Recode" block in Figure 3-1

The SBNR adder of Figure 3-1 may be used directly to construct a high speed sigma-delta modulator. However there is a further feature of redundant arithmetic which may be exploited. When conventional two's complement arithmetic is used in a systolic type structure, it is the least significant bits which are calculated first since the more significant bits depend on the least significant bits through carry propagation. This is not the case with redundant arithmetic which allows the order of execution to be reversed giving systolic type structures which operate with the most significant digit (msd) first. This is in line with the requirements for a sigma-delta modulator where it is only the top few digits of the filter output which are required. Data skewed msd first has the property that gains of less than one may be performed with negative latency. This is illustrated in Figure 3-3. Parallel data enters at the top and is skewed such that the digits become available in groups of three. However after the right shift by three digits (a divide by eight operation), the digit representing the output is available before the digit of equal significance has entered at the top. Thus the divide by eight operation can be considered to have a latency of -1. This can be exploited to decrease the pipeline delay around loops.

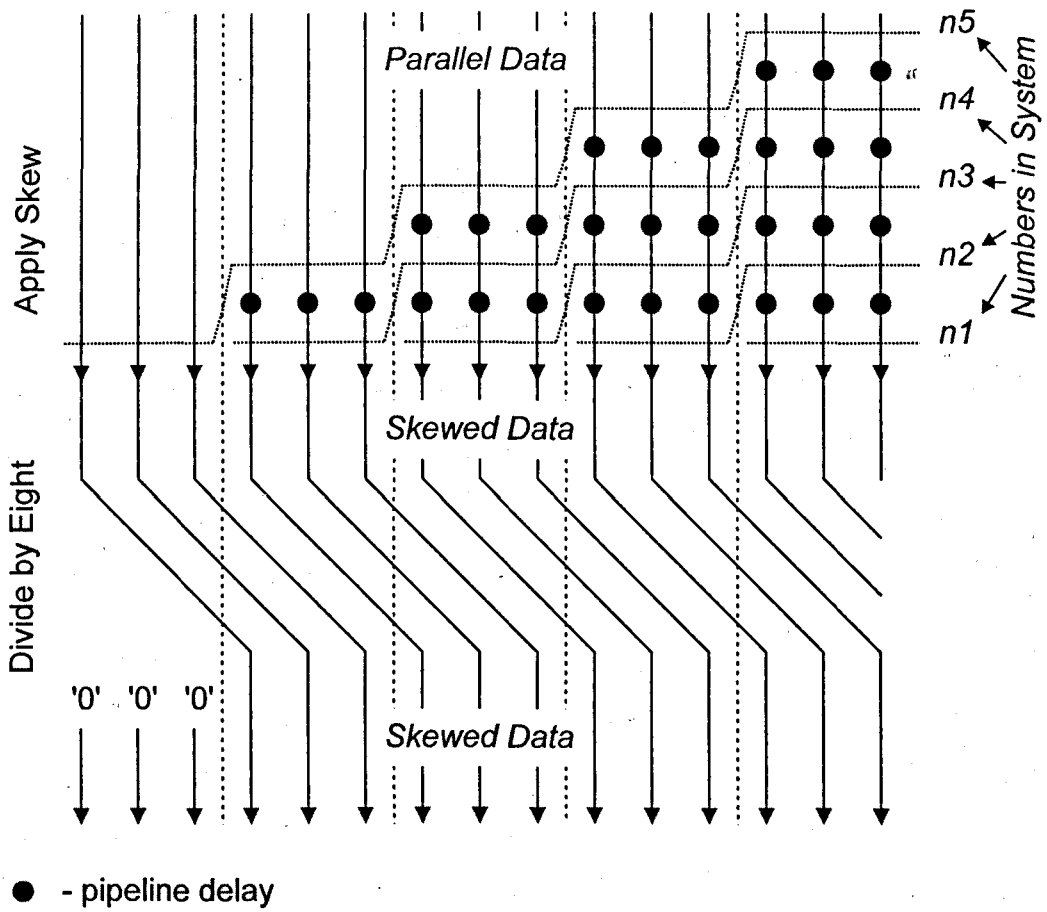


Figure 3-3 A pipe-lined SNBR representation with a divide-by-eight digit-shift operation.

3.3 SIGMA-DELTA TOPOLOGY

The sigma-delta modulator topology used here is based on a loop filter constructed from the cascade of integrators and resonators [3]. Figure 3-4 shows a single resonator. The values for the coefficients are determined using the methods discussed in Chapter 2. Since the input to the gains *f1* and *f2* can only take on the values +1 and -1, no multiplier circuit is required. The values of the resonator coefficients (*b*) are directly related to the open loop poles which are at the same locations in the z-plane as the NTF zeros. This is the only gain element for which a full SBNR number needs to be multiplied.

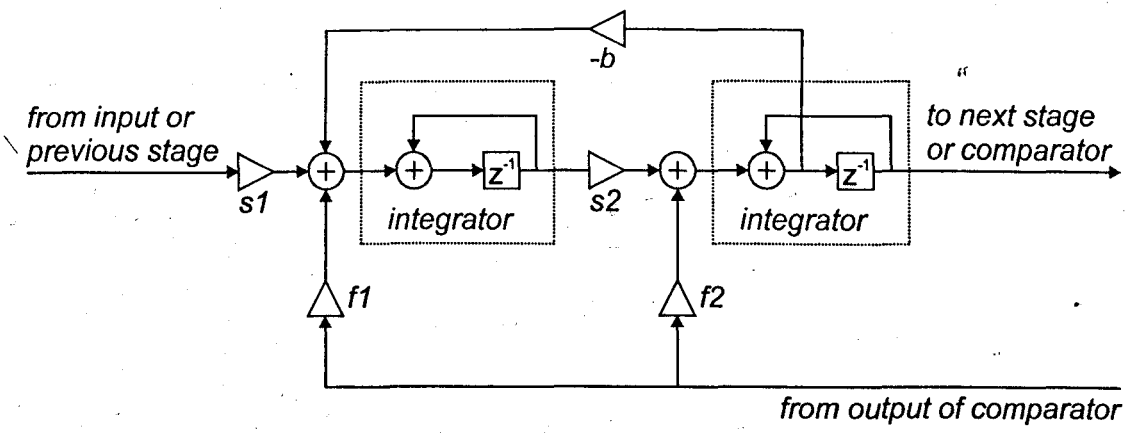


Figure 3-4 Resonator stage of a high order sigma-delta loop filter.

The single delay around the two integrator loop places stringent demands on the latency of the integrators. For base-band sigma-delta modulators an additional delay can be tolerated with the consequence of poles moving off the unit circle giving the NTF zero at a similar frequency as before, but not infinitely deep. The depth of the NTF notches has little affect on the output noise level.

It can be shown that the magnitude of the coefficient b in the feedback loop is small while the feedback coefficients f_i tend to be much larger [4]. If the feedback coefficient is sufficiently truncated, the addition of the two words can be achieved by concatenating the two words rather than by using an arithmetic circuit.

3.3.1 SBNR Adder

For the example sigma-delta modulator, it was decided that the data would be skewed by one pipeline delay every third digit, as was illustrated in Figure 3-3. This allows parallel computation of the top three digits which are required in parallel by the comparator. Three numbers need to be added for each integrator. Rather than cascading two dual input adders of the type in Figure 3-1, a two stage design was derived using the carry save concept of conventional two's complement arithmetic. This is shown in Figure 3-5. Three SBNR numbers are added and the effect of

transfer propagation limited to the next three more significant digits. This adder produces its output after a one clock cycle delay, from the pipeline cuts indicated.

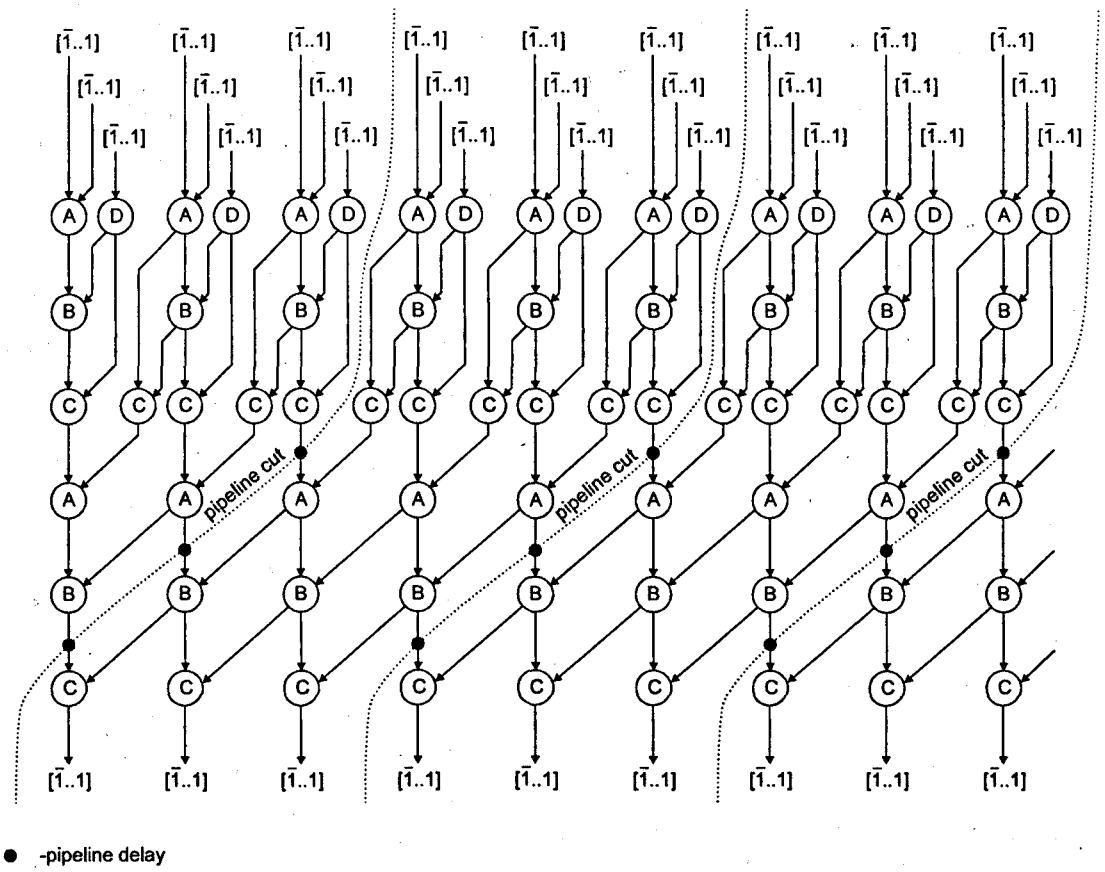


Figure 3-5 Adder for three inputs using skewed SBNR.

3.3.2 SBNR Comparator

The sigma-delta modulator requires a comparator. In conventional two's complement arithmetic, the output of the comparator is simply the sign bit of the input. However this is the result of a long carry propagation chain. In SBNR arithmetic there is no specific sign bit. The sign of the overall number is the sign of the left most non-zero digit. Logic is required to extract this from a parallel SBNR word. Fortunately the comparator performance in sigma-delta modulators is not critical, because any errors are reshaped with the full noise shaping function of the loop. In practice only the top three or four digits need to be examined as the additional truncation noise is less significant than the inherent quantisation error.

In the circuit presented, the comparator has one full pipeline delay during which to come to its decision and thus the speed of the comparator circuitry is not a limiting factor.

3.4 SBNR SIGMA DELTA MODULATOR

A sixth order modulator consisting of a cascade of three resonators was designed (Figure 3-6). The signal transfer function was designed with signal band gain of -6dB to ensure that a full amplitude input signal (+1..-1) did not cause the modulator to become unstable. The noise transfer function was designed with Butterworth poles so that the signal transfer function could have a low pass Butterworth response and be implemented with a single feed-in coefficient into the first integrator.

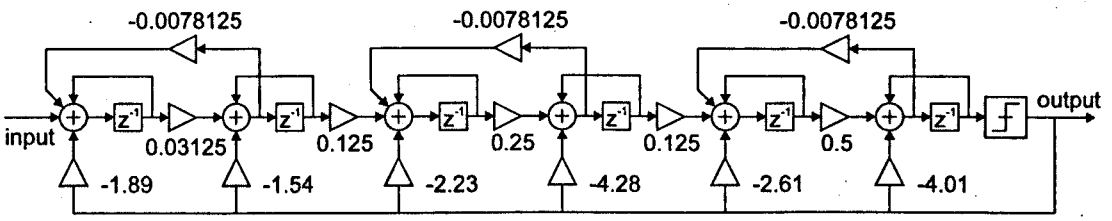


Figure 3-6 Sixth order sigma-delta modulator

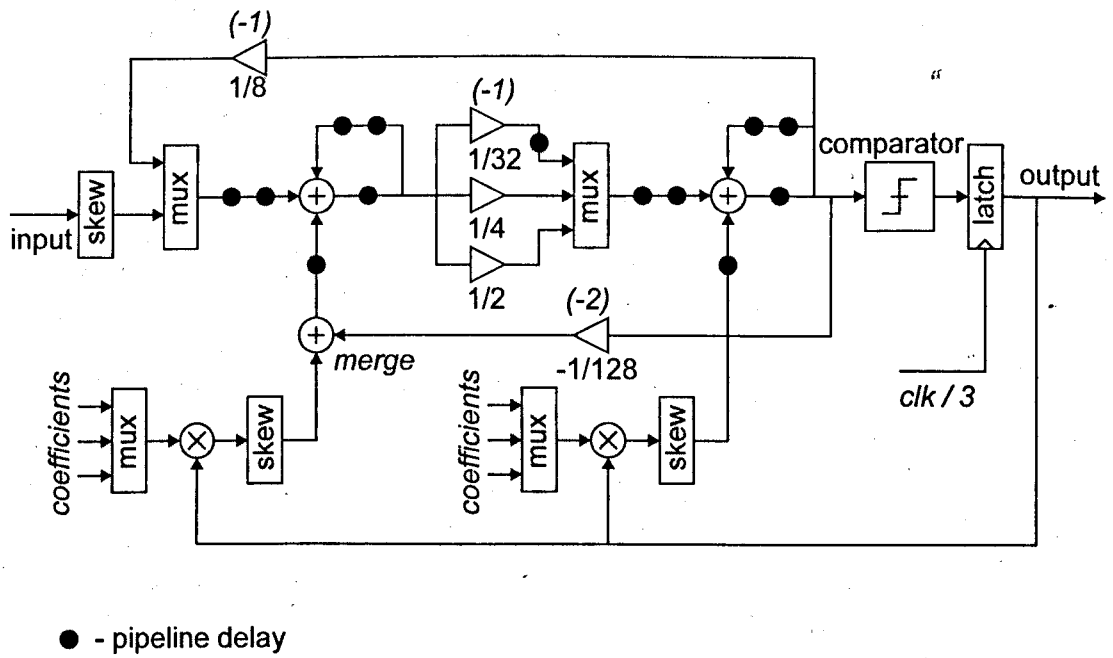


Figure 3-7 Sixth order modulator implemented using a single pipe-lined resonator.

The sixth order sigma-delta modulator was designed to use a single resonator stage to reduce the amount of digital circuitry. Each integrator output in the cascaded resonator topology shown in Figure 3-6 is dependent only on the previous output of the preceding integrator. This fact is used to calculate the data in the three resonators by using a single pipe-lined resonator with three pipeline delays corresponding to a single z^{-1} delay. The input of coefficients to the single resonator are multiplexed and the state information for the three resonators contained in the pipelines (Figure 3-7). The multiplexers are arranged such that the function of the third resonator is performed first, followed by the second and then the first. This is to allow time for the comparator to come to a decision and for the feedback coefficients (with sign applied by the comparator output) to start entering the adder pipelines as early as possible. A single pipeline delay on each of the three inputs of the two adders represents the pipeline delay internal to the adder.

Note that whilst it is convenient to consider the resonator in Figure 3-7 as being multiplexed, this is not truly the case. At the same time as one adder is computing the most significant digit of the output, other parts of the same adder are computing

other digits of the other resonators and even different digits of the previous values of the same word. The multiplexer for adjusting the scaling gain (between 1/32, 1/4 and 1/2) and the multiplexer for feeding the output of one resonator back into the input for the next must both take the skew into account. These multiplexers are actually taking different digits from each of their inputs to form the output.

The number in brackets beside some of the gain blocks indicates negative effective latency. These numbers need be taken into account when counting pipeline delays around loops. Note that there is the correct multiple of three pipeline delays around all the resonator loops. However there are only five delays around the loop which links the output of the resonator back to its input. This ensures that the data representing the output of the first resonator arrives at the right time to be interpreted as input to the second and similarly for data leaving the output of the second resonator and entering the third. The output of the final resonator is not feed back to the input, but instead the system input is sampled and multiplexed into the first integrator.

The scaling factors were chosen to be the nearest power of two in order to roughly equalise the signal amplitudes at each of the integrator outputs. The resonator coefficients were also quantised to powers of two. Because the resonator coefficients are small (1/128), two extra pipeline delays become available which permits closure of the resonator loop with only three pipeline delays (corresponding to a single z^{-1} delay) and so gives NTF zeros which lie exactly on the unit circle. Additional adders could use the three additional delays available to place the NTF zero frequencies more accurately. It is likely that a low latency SBNR multiplier could be used to give programmable NTF zero frequencies with arbitrary precision.

Since the two's complement number system is a subset of SBNR there is no hardware overhead in converting input in two's complement form to the SBNR form required by the first adder. The word length of all SBNR data was 5 integer digits and 20 fractional digits and this could be increased without decreasing the throughput rate.

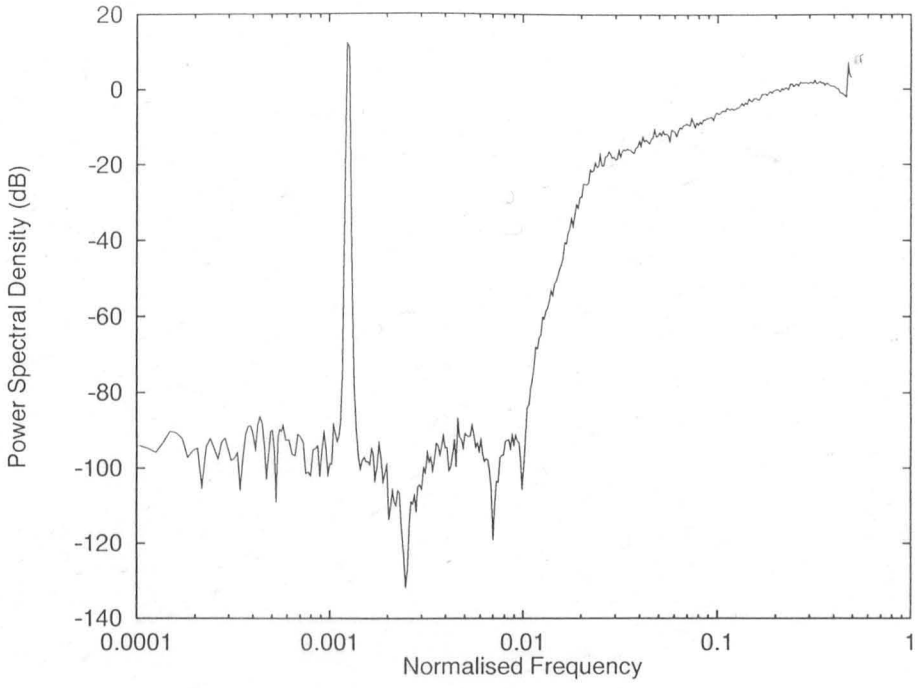


Figure 3-8 FFT of output of digital simulation of sixth order SBNR sdm.

The sixth order system of Figure 3-7 has been simulated at the gate level using a commercial digital simulator. Figure 3-9 shows the top level of the gate level schematics which bares some resemblance to Figure 3-7. The output spectrum (Figure 3-8) shows the expected sixth order shaping with an undistorted single tone. The signal to noise ratio from this simulation is 85dB with the input 20dB smaller than full scale. The modulator was designed for an over-sampling ratio of 50.

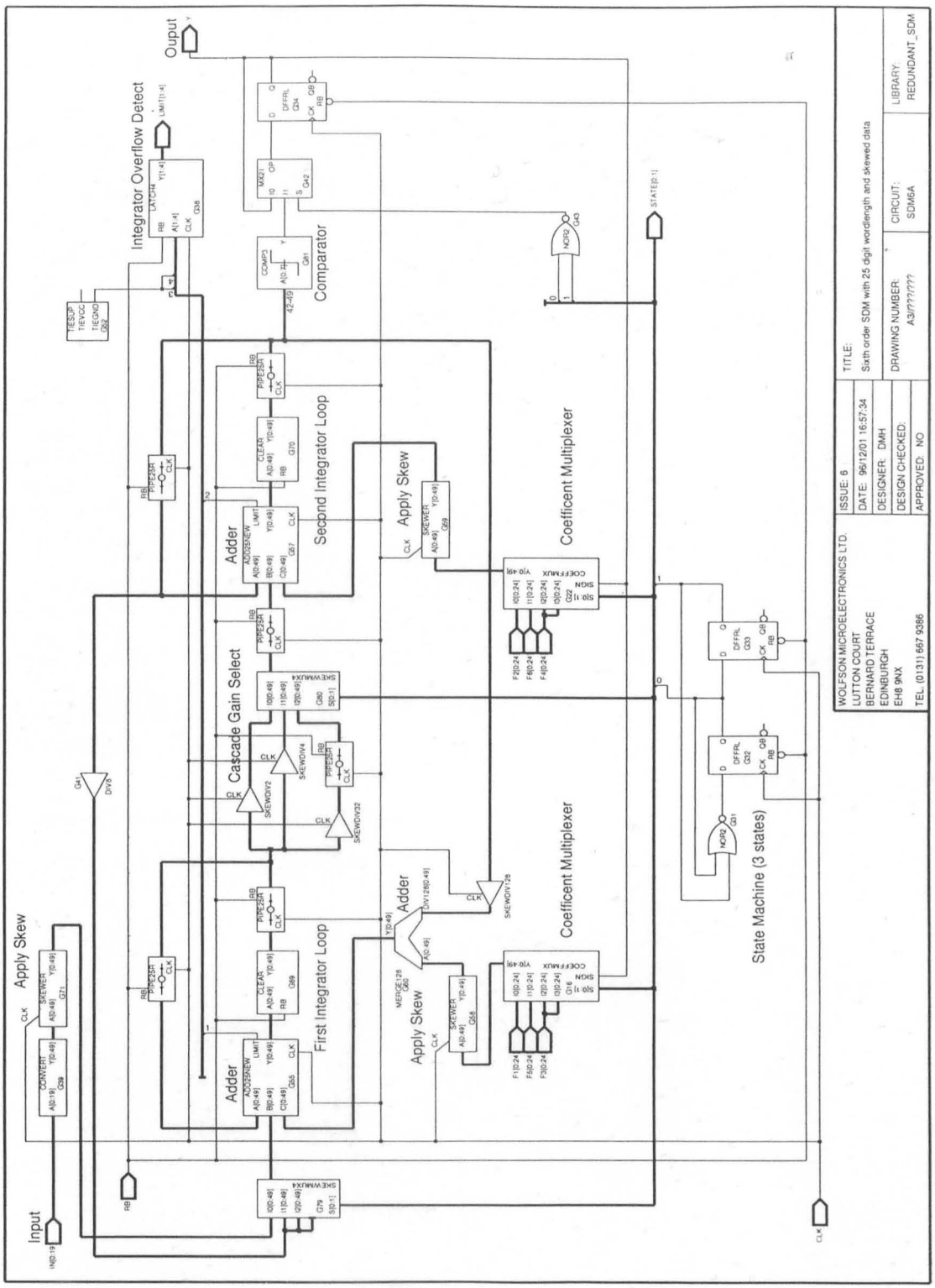


Figure 3-9 Top level schematic for 6th order SBNR based Sigma-Delta Modulator.

3.5 ALTERNATE REDUNDANT IMPLEMENTATIONS

The implementation described above used standard two-level logic to represent SBNR quantities utilising two binary digits to represent each SBNR digit. Alternatively each digit may be represented by more voltage or current levels or maybe exploit a different redundant arithmetic.

3.5.1 SBNR implemented using ternary level logic

Digital IC process technologies are almost exclusively optimised for two level (binary) logic circuits. However there is some scope for implementing arithmetic circuits using multiple-valued logic (MVL) implemented directly on silicon using either multiple voltages or multiple current levels to represent the numeric quantities [5]. The design of these circuits is far more difficult than for conventional logic, but the resulting circuits have demonstrated benefits in both size and speed.

3.5.2 Higher radix based systems

The fact that about three or four SNBR digits are required as input to the comparator, suggests that a higher radix number system may be more suitable. For example a redundant number system of radix 16 could be implemented with conventional two complement arithmetic using 5 bits to represent values from -16 to +15. The resulting circuit would then only be 25% larger than a circuit using two's complement representation. This overhead is similar to that incurred if conventional arithmetic speed up techniques are used [6] - the advantage of the redundant number based approach is that the word length may extended without limit, and without any loss of speed.

3.6 RECONSTRUCTION FILTER DESIGN OPTIONS

Telecommunications systems often have a strict requirement for out of band noise suppression. This is to ensure that spurious emissions from a transmitted signal do not interfere with a weak signal in a channel using an adjacent frequency band. The out of band performance depends on how effectively the shaped quantisation noise is removed and also upon the linearity of the reconstruction filter. Distortion products may lie out-with the signal band and contribute to poor out of band performance.

A high order sigma-delta modulator gives the advantage of allowing the analogue reconstruction filter to operate a lower rate, but the filter must then also be of high order to remove the highly shaped noise. Alternatively a simpler, lower order modulator may be used together with a simpler reconstruction filter, both of which must operate at a higher sampling rate. Over-sampling ratios for signal reconstruction purposes are most commonly in the range of 384 down to about 48, while the modulator order typically varies over the range of two to about six. For the highest signal bandwidths, high order modulators must be used so as to keep the reconstruction filter speed practical.

The requirements on the performance of the analogue reconstruction filter in an over-sampled DAC are more stringent than the requirements on the analogue loop filter in over-sampled ADCs. This is because, in the DAC reconstruction filter, each operational amplifier (or equivalent active processing block) contributes to the output distortion and once in-band distortion has been introduced it cannot be removed. This can be contrasted with the ADC case where the gain distributed throughout the loop filter reduces the distortion contribution of the all the components with the exception of the first stage. Therefore high order loops (5-7th order) may be constructed without the additional opamps contributing to increased distortion and circuit noise.

The best compromise between modulator order and oversampling ratio is very difficult to ascertain. The power requirements for both high and low oversampling

rates are approximately equivalent; since the high speed option requires fewer active components, but these components must operate a higher rate and will consume more power.

An alternative to the typical reconstruction filter is discussed here and which offers good performance with low component ratio spread.

The typical SDM DAC system uses an IIR filter operating on sampled data followed by a low order continuous time filter. The one bit DAC is typically built into the front end of the filter which can be designed using one of the many existing design methods. The alternative solution described here is to use a FIR filter, giving a FIRDAC [7,8,9]. This can be implemented using IC process technologies which are not normally particularly suitable for analogue filter design. This is because the input signal is a one bit digital signal and the delay element needs only to transfer the single bit of data. Thus the delay line part of the FIR filter is digital, but the coefficients and the summer are implemented using analogue methods. Many one bit DACs are required for each of the delay line taps, but the one bit DAC is usually very small. Figure 3-10 illustrates the divide between the digital and analogue sections.

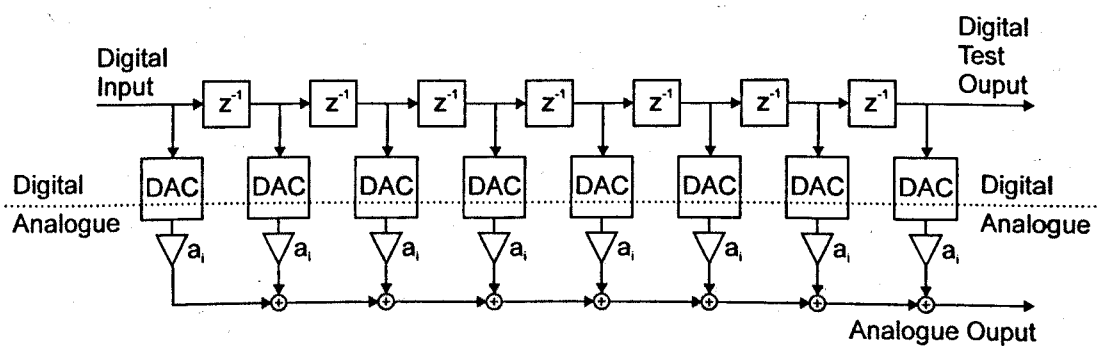


Figure 3-10 Partitioning of analogue and digital circuitry in the FIR DAC filter.

The obtainable responses of the filter transfer function are very dependent on the impulse length. Since in this case, the filter acts on over-sampled data, the filter length in terms of number of taps must be quite long. Additionally, component matching issues limit the range of coefficients and the accuracy with which they may

be reliably integrated. This restricts the maximum attainable stop-band rejection. Various design techniques may be used to try to limit the filter length and the spread on the coefficients. The most important frequency region is that closest to the signal band. The design techniques optimise attenuation over this region at the cost of poorer attenuation at higher frequencies. A simple first or second order IIR filter may be then used to remove this high frequency energy.

3.7 FIR TRANSFER FUNCTION DESIGN

The FIR transfer function can be optimised to reduce its length, the number of non-zero taps and the coefficient spread, by making some compromises. The FIR filter is to be principally used to provide attenuation at frequencies just beyond the signal band.

The transfer function:

$$H(z) = 1 + z^{-d} \tag{3-28}$$

provides zeros at the frequencies

$$f = \frac{(1+2k)}{2Td} \quad , \quad k = 0, 1, 2..d-1 \tag{3-29}$$

where T is the sampling period (in seconds) and f is the notch frequency in Hertz. Simple filter responses may be generated by combining several factors of the type above. The example response illustrated in Figure 3-11 was developed to meet the stringent out of band noise requirements for a digital mobile radio standard [10,11]. The filter response is compared against a 4th order Bessel filter since phase linearity is of crucial importance because there is no adaptive equalisation in the base-station receiver. This system has a requirement for less than -70dB of the base-band power to be present in the first adjacent band, yet the channels which are 24.3kHz wide are separated by only 1.4kHz. Response notches were placed at 50kHz, 75kHz, 100kHz,

150kHz and 200kHz (and also the resulting odd multiples) to meet the requirement. The sigma-delta modulator used was only second order, but incorporated a noise transfer function zero at 25kHz to provide sufficient first adjacent channel noise performance. Incorporating a zero in a second order modulator has the effect of reducing the low frequency gain of the loop filter with the consequence that the quantisation noise is attenuated less, resulting in increased in-band noise and distortion. The result was that the in-band SNR was only 66dB - however this is more than sufficient to meet the system requirements for the data signal which are in this case specified as peak and average (root-mean-square) vector errors. The specification allows more noise to be transmitted in the signal band than the adjacent bands and this has been exploited to reduce component cost and power consumption.

The transfer function can be described in TFN format as a product of factors as follows:

```
gain 0 0dB fclk 3.6M expand {
     36:  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
     24:  1 0 0 0 0 0 0 0 0 0 0 0
          0 0 0 0 0 0 0 0 0 0 0 0 1
     18:  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
     12:  1 0 0 0 0 0 0 0 0 0 0 0 1
     9:   1 0 0 0 0 0 0 0 0 0 1
     / {
          // poles have flat contribution
          // but ensure that group delay is correct
          36:  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
          24:  1 0 0 0 0 0 0 0 0 0 0 0
               0 0 0 0 0 0 0 0 0 0 0 0 0
          18:  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
          12:  1 0 0 0 0 0 0 0 0 0 0 0 0
          9:   1 0 0 0 0 0 0 0 0 0 0
          }
     }
```

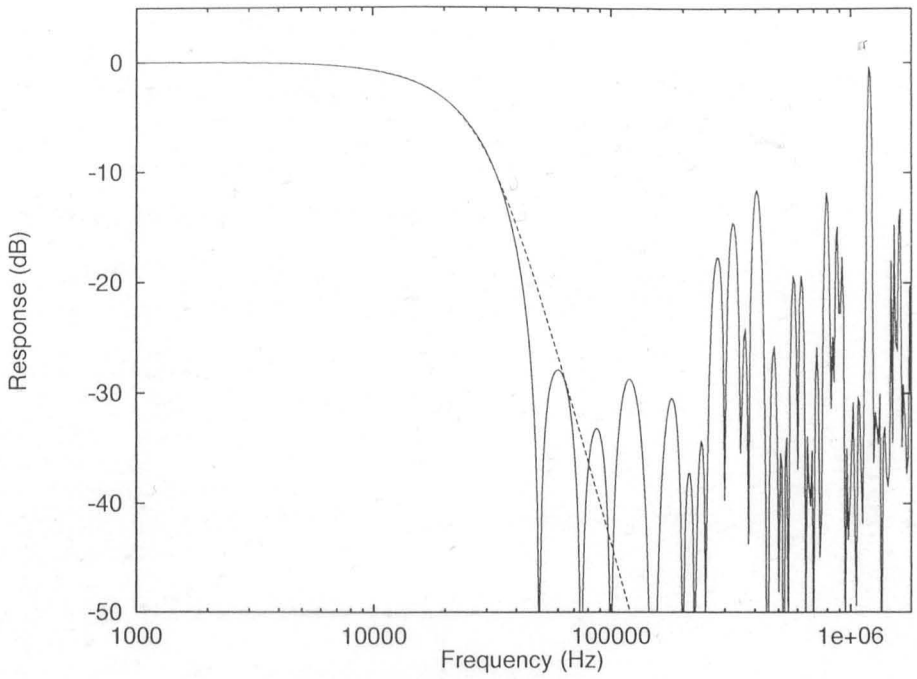


Figure 3-11 Response of FIR filter compared with a 4th order Bessel response.

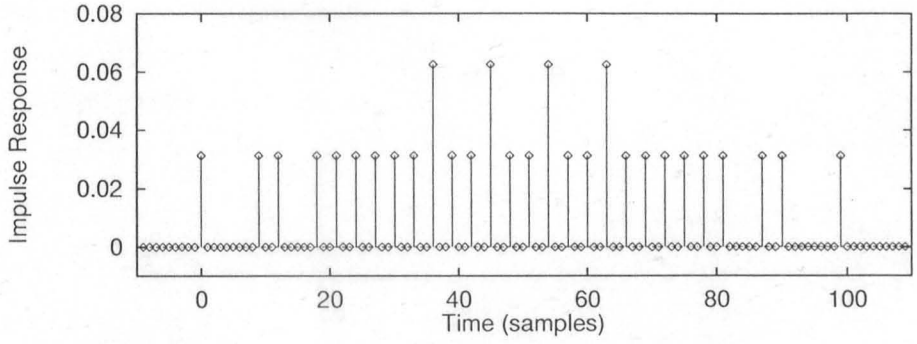


Figure 3-12 The impulse response of the FIR filter. Only 28 of the 100 taps are non-zero.

Figure 3-13 shows the adjacent root-raised-cosine (RRC) channel receiver responses and the simulated transmitted spectrum of random data after being encoded, modulated, RRC filtered, sigma-delta modulated and filtered by the FIR filter, a second order switched capacitor IIR filter and a second order continuous time smoothing filter. Measurements from this simulation result predict that the immediately adjacent channel power transmitted to be -79dB relative to the total transmitted power. The power in channels beyond this are less than -90dB.

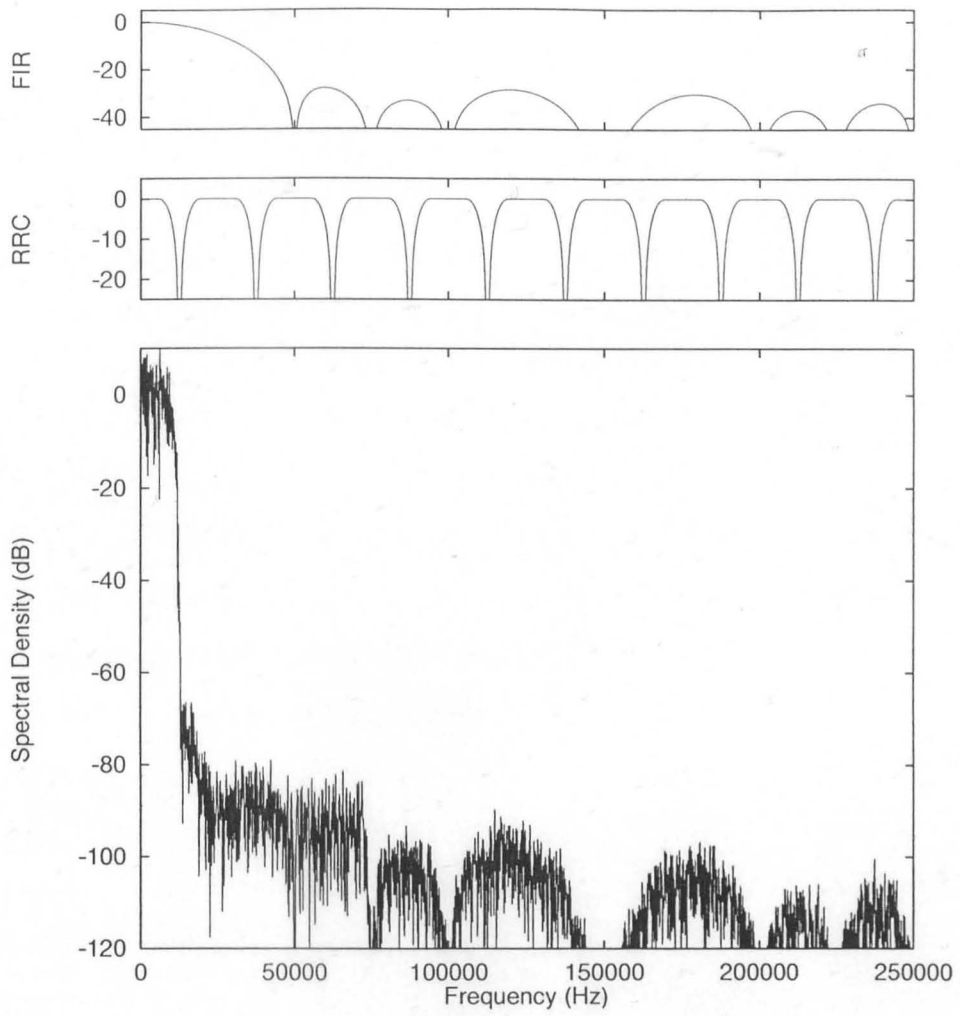


Figure 3-13 Spectrum of modulated random data after the analogue FIR and IIR filters.

3.7.1 Filter Implementation

The FIRDAC filter discussed above was implemented using switched capacitor techniques. The FIR filter was 100 delay elements long, but only 28 of these were used (the other coefficients are zero). The FIR delay elements were implemented using minimum sized transistors forming a dynamic shift register. Each of the 28 taps required a small non-overlapping clock generator to drive the one bit DAC switches. The process used was a double poly-silicon, 1.2 micron CMOS process and the area devoted to the shift register and non-overlapping clock generators would

shrink markedly on smaller geometry processes. As process geometry shrinks this will become an important advantage of this technique.

Production test of the filter is relatively simple - the integrity of the shift register can be confirmed entirely digitally by checking the output of the final delay element for the delayed input sequence. The integrity of the taps can be checked by using a square wave digital source with frequencies corresponding to the notch frequencies and checking that there is sufficient attenuation. Should any single tap be non-functioning, a component of the input signal will leak through. Finally a low frequency signal can be used to check that filter passes the in-band signals. The test time required for this procedure is negligible compared to the time required to test conventional analogue filters.

The FIRDAC filter has some similarities with some recent work on multi-level DACs [12,13,14,15]. These new methods, referred to as dynamic element matching (DEM), implement an n level DAC using n individual elements. Each of the elements are selected or not depending on various algorithms. The result is that the error due to mismatch are less correlated with the input signal and the effect of component value errors is frequency shaped such that most of the error power is outside of the signal band. The FIRDAC filter described above can be considered to be a multilevel DAC since it uses 32 individual unit elements and the output can take on one of 32 different levels. In this case the unit selection algorithm is more complex and has many more states. The difference with the FIRDAC approach is that errors in the magnitude of individual units contribute to leakage of the quantisation present on the input signal - there is no additional noise or distortion source. With the DEM methods component mismatches leads to a new source which can contain in-band energy (even if reduced by error frequency shaping.) Another difference is that despite producing multiple output levels, the FIRDAC has a single bit input which is convenient if the digital modulator is separated from the analogue DAC and filters. Both methods result in lower out of band noise than single bit DACs. The FIRDAC has lower noise because of the filtering and the DEM schemes have lower noise because the multi-level quantiser has finer resolution.

Alternative technologies for implementing FIRDAC filters include the switched current and current mirror based methods. These have the advantage of being compatible with purely digital CMOS processes. FIRDAC filters implemented in this way will scale very well with smaller process geometry. Since the response comes from a single stage, there is a reduced number of active components in the signal path with the consequential benefits of lower noise and distortion and lower power.

3.7.2 Alternate Design Method

The above design method is somewhat pragmatic, in that it requires the designer to place every zero manually. It would be difficult to design particularly high performance transfer functions using this method. An alternative is to use sigma-delta modulation to arrive at a quantised version of a prototype filter [16]. Figure 3-14a shows the response of a prototype filter, which in this case is a length 128 FIR low-pass with an equiripple Chebyshev response, designed using the Remes exchange method [17]. Figure 3-14b shows the response when the coefficients have been quantised using sigma-delta modulation. Since coefficient accuracy does not contribute to distortion, it is not necessary to limit the quantisation to two levels. In this case three non-zero levels have been used. It is possible to combine this method with the manual method. Figure 3-14c includes an extra manually added length 4 comb filter to reduce high frequency quantisation noise. The resulting filter coefficients are the convolution of the FIR coefficients and the comb filter coefficients. This improvement comes at the cost of requiring many more unit elements (the ideal coefficients have effectively been more accurately quantised). This is clear from the comparison of corresponding impulse responses shown in Figure 3-15. However good rejection at high frequencies has significant practical advantage and this is discussed in the next section.

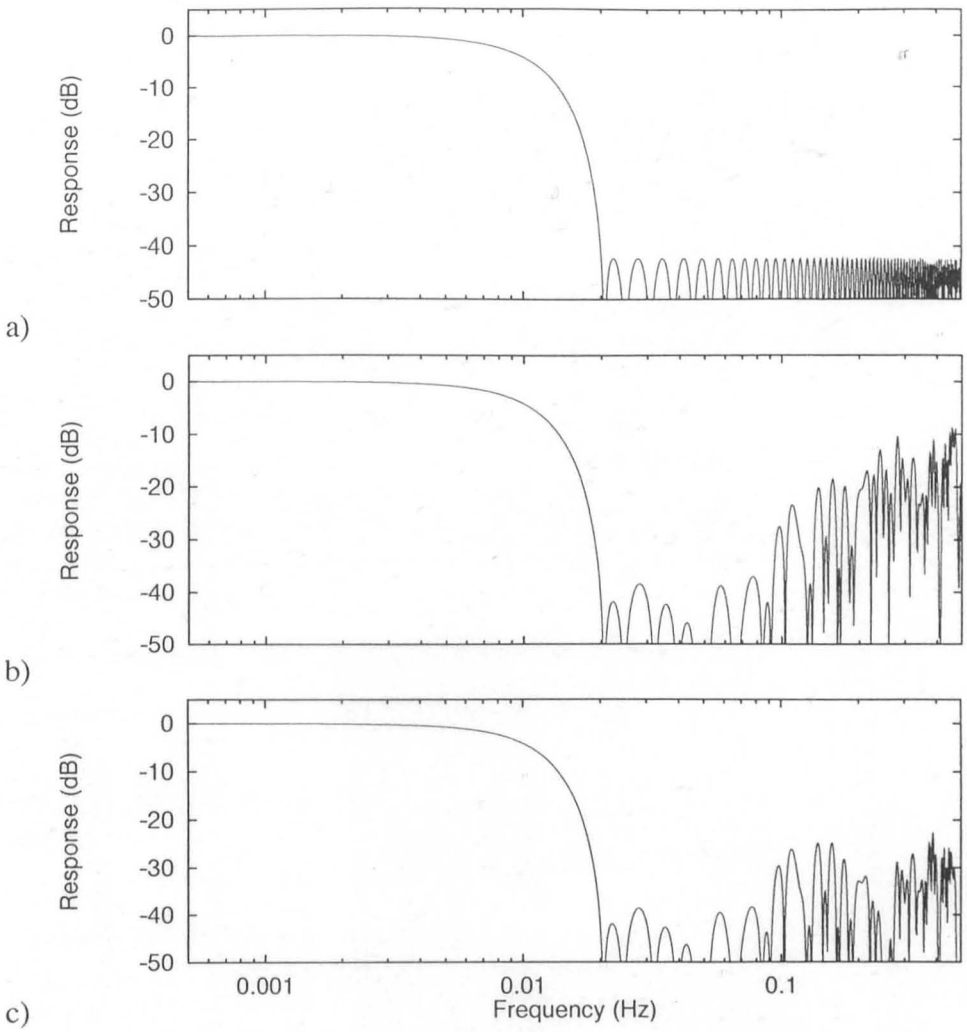


Figure 3-14 Design of FIRDAC coefficients using sigma-delta modulation.

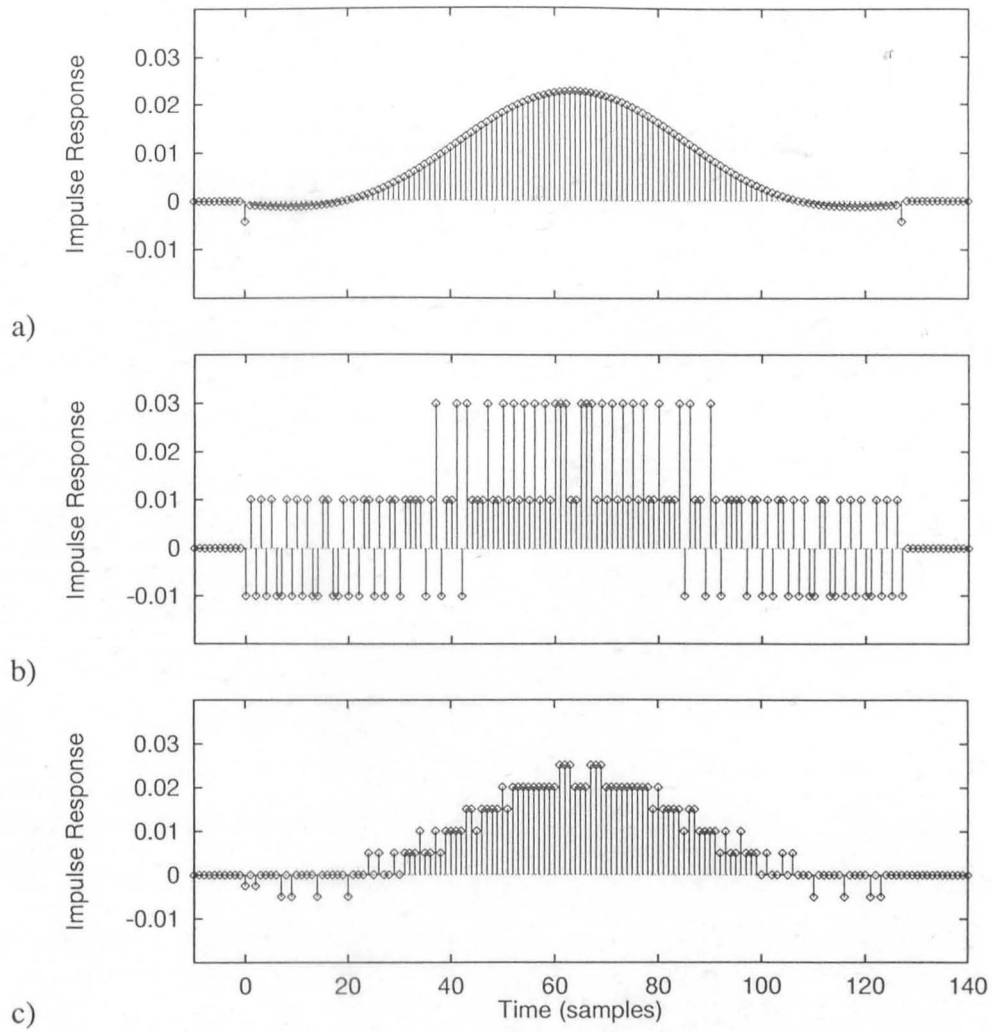


Figure 3-15 The impulse responses corresponding to Figure 3-14

3.8 REJECTION OF MODULATED TONES

Spurious tones plague practical implementations of sigma-delta modulators [18]. The tones arise through a number of different ways. For DACs it should be assumed that the digital signal processing has produced a signal which is tone free and that any systematic tone sources have been removed through dithering. However tones still arise in practical implementations, by coupling from the reference and from inter-modulation distortion in the reconstruction filter. The problem is severe because it is high frequency digital noise around fs/2 which is modulated down to the base-band by the digital input of the DAC. The digital input can contain strong tones near half the sample rate which are closely related to the base-band signal [19]. An ordinary

Nyquist rate DAC does not exhibit this noise mixing menace, since the digital input is normally band-limited such that high frequency signals which are modulated remain out-of-band. Another source of tones in the base-band is the difference frequency component resulting from inter-modulation distortion in the first stage of analogue filtering between two high frequency tones. Methods for dithering sigma-delta modulators to eliminate tones at high frequencies have been studied in [19]. However, even if discrete tones are removed, there is still the potential for the in-band SNR to be degraded by modulation of any high frequency content on the reference and the high amplitude / high frequency sigma-delta quantisation noise.

For a DC input V_{DC}, the dominant tone is at $(1-V_{DC}/V_{REF})f_s/2$ [19]. For a small sinusoidal input the result is frequency modulated side-bands at high frequencies. This can be seen in Figure 3-16 which shows the spectrum arising from a second order sigma-delta modulator, with a -40dB sinusoid with a small dc offset. Figure 3-17a shows that the base-band performance is considerably poorer when a clock signal of fs/2 is coupled on to the reference at a low level (-60dB). When a FIRDAC is used, the base-band performance is not affected (Figure 3-17b).

The tone problem is lessened if the FIRDAC response has good attenuation at high frequencies. This is because the input digital signal is effectively filtered before being multiplied by the reference. High frequency digital noise will still be modulated, but the modulation products will be attenuated. The FIRDAC system is shown in Figure 3-18. This can be redrawn as the equivalent system in Figure 3-19, which clearly shows the input signal being filtered before being multiplied by the reference.

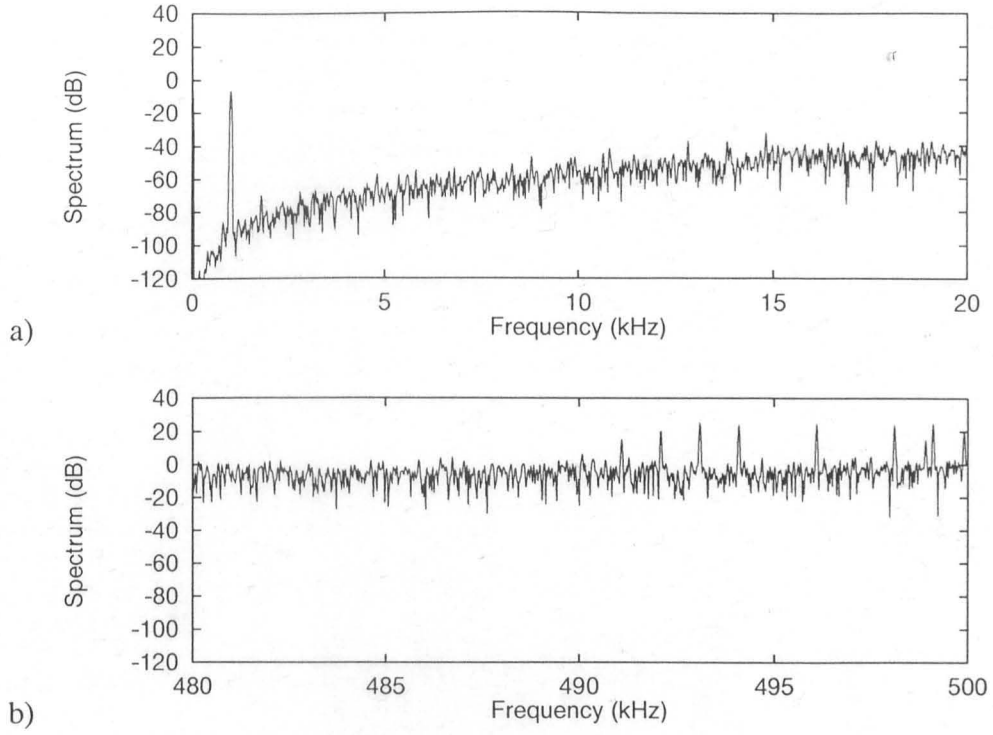


Figure 3-16 Bottom and top 20kHz of the output spectrum from a small ac input to a second order sigma-delta modulator, showing distinct tones at high frequencies.

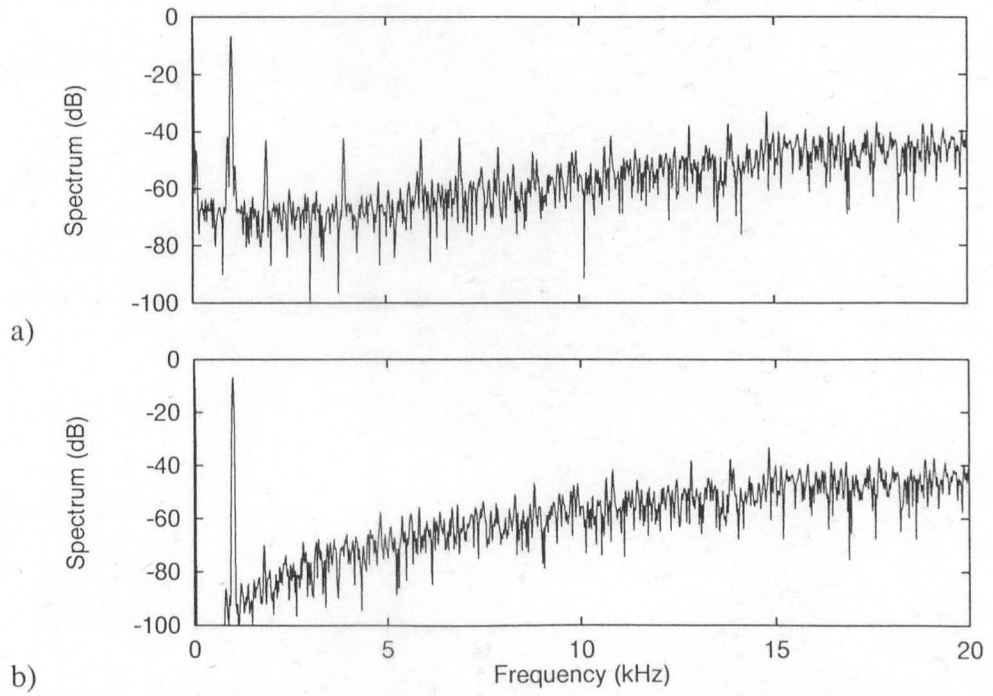


Figure 3-17 The base-band spectrums when a small high frequency signal is present on the reference.

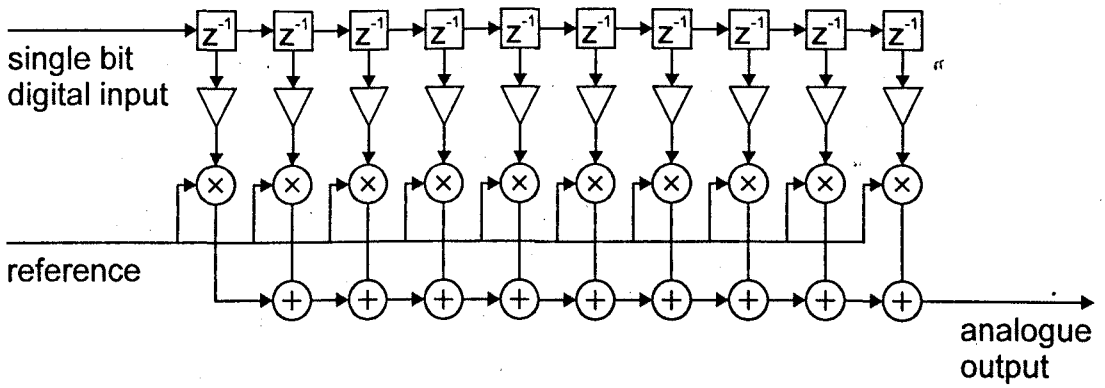


Figure 3-18 Signal flow for FIRDAC.

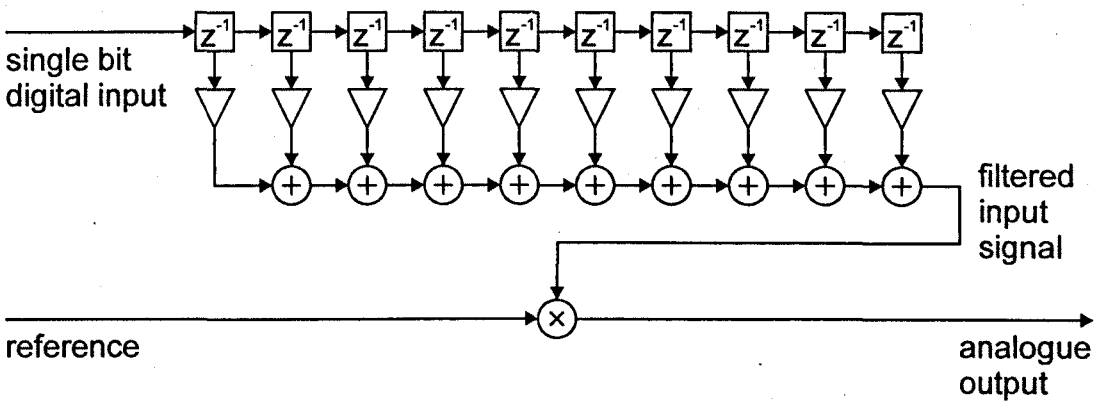


Figure 3-19 Equivalent system of FIRDAC.

Multi-level SDM DACs suffer less from tone modulation than one bit systems since there is less high frequency energy. The FIRDAC system shares this advantage. The decreased high frequency content reduces both the direct modulation of high frequency signals on the reference and also inter-modulation products due to non-linearity in the first stage of the subsequent analogue filter.

3.9 SUMMARY

This chapter has studied the problems arising in the design of high speed digital sigma-delta modulators and their associated analogue reconstruction filters. Features of SBNR arithmetic have allowed the design of a sixth order modulator (Figure 3-7) which produces a new sample every third clock cycle, with the clock rate limited by a

single propagation delay through the adder of Figure 3-5. Many other design decisions can be made, ranging from fully parallel designs where the order and word length could both be increased without affecting throughput, to more heavily pipelined systems using a single adder. The problem of using the high speed single bit output of the SBNR sigma-delta modulator has been addressed by making use of a semi-digital FIR approach which uses multiple DAC elements in a topology which does not require precise matching. This example also raised the issue that the design of noise shaping in SDM must take account of out of band specs together with the reconstruction filter. It was also shown that the FIRDAC has important practical advantages in its insensitivity to high frequency signals on the reference which make it an attractive choice even if particularly low out-of-band noise is not a design issue.

3.10 REFERENCES

[1] R.F. Woods, J.V. McCanny, "Design of a High Performance IIR Digital Filter Chip," *IEE Proceedings Part E*, vol. 139, no. 3, pp. 195-202, May 1992.

[2] S.R. Norworthy, R. Schreier, G.C. Temes, *Delta-Sigma Data Converters, Theory Design, and Simulation*, IEEE Press, Pitscataway NJ, 1996.

[3] R.W. Adams, P.F. Ferguson and A. Gansesan, "Design of Single Bit Noise-Shaping Loops with High Order Loop Filters", *Audio Engineering Society 89th Convention*, Los Angeles, Preprint #2974, September 1990.

[4] D.M. Hossack, J.I. Sewell, "Computer-Aided Design of High Order Sigma-Delta ADCs," *IEE Colloquium on Advanced A-D and D-A Conversion Techniques and Applications*, pp 2/1-2/4, London, May 1993.

[5] C.T. Clarke, *The Implementation and Applications of Multiple-Valued Logic*, PhD Thesis, University of Warwick, Warwick 1993.

[6] N.H. Weste, K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, 1993.

[7] D.K. Su and B.A. Wooley, "A CMOS Oversampling D/A Converter with a Current-Mode Semi-digital Reconstruction Filter," *IEEE Journal of Solid State Circuits*, vol. 28, no. 12, pp. 1224-1233, December 1993.

[8] H. Qiuting, "Linear-Phase Filters Configured as a Combination of Sigma-Delta Modulator, SC Transversal Filter and a Low-Q Biquad," *IEEE International Symposium on Circuits and Systems*, Chicago Illinois, pp. 1306-1309, May 1993.

[9] Huang Qiuting, "Mixed Analog/Digital, FIR/IIR Realization of a Linear Phase Lowpass Filter," *IEEE Journal of Solid State Circuits*, vol. 31, no.9, pp. 1347-1350, September 1996.

[10] R. Wilson, "Air Apparent," *Electronics Weekly*, Reid Business Information Ltd., Sutton, UK, p15, April 24, 1996.

[11] B. Razavi, "Challenges in Portable RF Transceiver Design," *IEEE Circuits & Devices*, pp. 12-25, September 1996.

[12] R.T. Baird and T.S. Fiez, "Improved ΣΔ DAC Linearity using Data Weighted Averaging," *IEEE International Symposium on Circuits and Systems*, Seattle, Washington, pp. 13-16, May 1995.

[13] H. Lin, J.B. da Silva, B. Zhang, R. Schreier, "Multi-Bit DAC with Noise-Shaped Element Mismatch," *IEEE International Symposium on Circuits and Systems*, Atlanta, Georgia, vol. I, pp. 235-238, May 1996.

[14] R. Henderson and O. Nys, "Dynamic Element Matching Techniques with Arbitrary Noise Shaping Function," *IEEE International Symposium on Circuits and Systems*, Atlanta, Georgia, vol. I, pp. 231-234, May 1996.

[15] O. Nys and R. Henderson, "A Monolithic 19-bit 800Hz Low Power Multi-bit Sigma Delta CMOS ADC using Data Weighted Averaging," *Proceedings of the 22nd European Solid-State Circuits Conference*, Neuchatel, Switzerland, pp. 252-255, September 1996.

[16] P.W. Wong and R.M. Gray, "FIR Filters with Sigma-Delta Modulation Encoding," *IEEE Transactions on Acoustics and Signal Processing*, vol. 38, no. 6, pp. 879-990, June 1990.

[17] T.W. Parks and C.S. Burrus, *Digital Filter Design,* Wiley, 1987.

[18] S. Harris, "How to Achieve Optimum Performance from Delta-Sigma A/D and D/A Converters," *Journal of the Audio Engineering Society,* vol. 41, no. 10, pp. 782-790, October 1993.

[19] Lars Risbo, Σ-Δ *Modulators - Stability Analysis and Optimization*, PhD Thesis, Technical University Of Denmark, Lyngby, June 1994.

4

COMPANDERS EXPLOITING SIGMA-DELTA MODULATION

4.1 INTRODUCTION

Signal amplitude compression is a well established technique to increase the perceived performance of voice communication systems with noisy transmission channels [1,2]. The technique is essentially an automatic gain control system whereby the amplitude of small signals is increased before transmission and this gain compensated for in the receiver. The resulting system gives a higher signal to noise ratio over a wider dynamic range when the channel has additive noise. The rapid development of cordless and cellular mobile phones stimulated design research into deriving cheap and robust implementations on modern CMOS IC processes operating at low voltage (3.0 volts) and with low power. Companders can be implemented on standard CMOS processes [3] although efficient and robust implementation of compander systems has been problematic due to the lack of a reliable analogue multiplying element. This chapter describes how the single bit output obtained from sigma-delta modulation can enable the design of robust companders on CMOS processes.

The companding process exploits the psycho-acoustic characteristic that the ear is more sensitive to additive noise when the signal level is low. For louder signals noise is masked by the signal and is less intrusive. Therefore quiet signals should be boosted before transmitting across a noisy channel. This operation is termed *compression* since the signal is processed to reduce its dynamic range.

To ensure accurate reproduction at the receiving end, the effect of the boosting must be compensated by using an *expander*, which recovers the dynamic range by attenuating quieter signals. The use of a compressor and expander together is termed a *compander*. Figure 4-1 illustrates how the signal level is adjusted throughout the process. This is an analogue system with an analogue input and output.

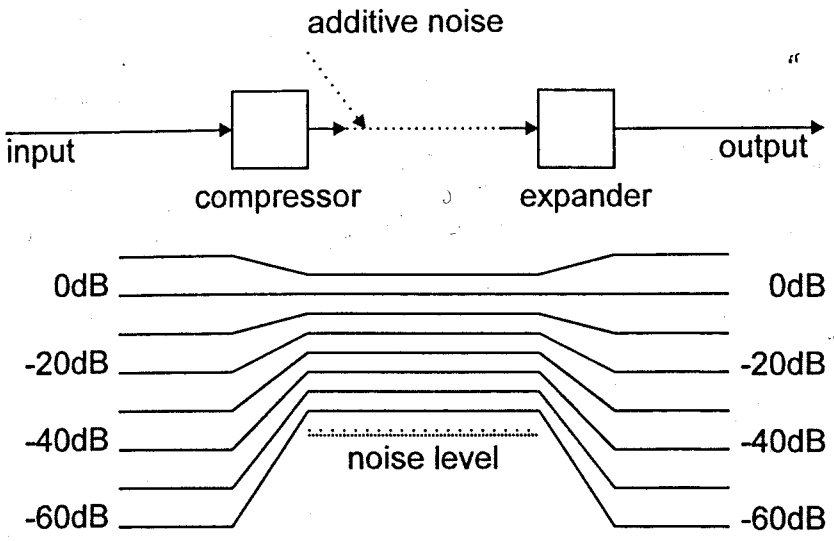


Figure 4-1 Compander System.

The recovered signal level becomes more sensitive to the gain through the system, since a 1dB error through the channel results in a 2dB error at the output. This is not a problem if the channel characteristics are constant (e.g. a fixed line) or if frequency modulation is used to modulate the signal through the channel. In the latter case, signal fading results in SNR degradation but not in a loss of recovered signal amplitude. The combination of FM modulation and companding was chosen for use in all the major analogue cellular telephone standards throughout the world.

The ratio in the dynamic range (in dB) of the input signal to the compressed signal is most often chosen to be 2:1 since this can be implemented using solely multiplicative elements without requiring logarithmic and antilogarithmic (or other non-linear function) elements.

The requirements for the performance of companders are set down by various international telecommunications standards. The specifications include the maximum gain error over a specified input range, step response characteristics and the signal to noise ratio.

4.2 CONVENTIONAL IMPLEMENTATION

The conventional topologies for implementing expanders and compressors are discussed in [4,5]. A full wave rectifier followed by a low pass filter is used to extract the signal level from the signal. In the expander this low bandwidth signal is used to multiply the input signal (two quadrant multiplication) to increase the dynamic range. The compressor is formed by incorporating an expander inside a feedback loop.

4.2.1 Design Issues

Several problems are encountered when integrating companding systems on to silicon.

One fundamental problem is the requirement to multiply analogue quantities. This is most easily accomplished by using a silicon process which provides bipolar transistors which allow use of the Gilbert cell translinear multiplier [6,7]. This is not an option when using CMOS processes which are required for digital circuitry and for other audio band signal processing functions using switched capacitor techniques. The use of BiCMOS processes allow both the compander and SC circuits to be implemented on a single chip [8], but the extra process steps add cost and are not always available. CMOS multipliers have been reported but these are much less robust than the bipolar implementation.

The implementation of the envelope detection operation gives rise to two additional design problems. The full wave rectifier must have a little input offset, since offset cannot be subsequently removed by ac coupling and any offset will dominate for small signal levels resulting in low level mis-tracking. To meet the system requirements the expander must be able to measure the amplitude of a -30dB (relative to full scale) signal with a maximum error of 1dB. For a system with a one volt peak signal swing (typical for a 3 volt system), this implies that the amplitude

must be measured to within $1.0\times10^{-30/20}\times(10^{1/20}-1)$ which is less than 4mV. Achieving this level of offset is very difficult.

The second problem with the design of the envelope detector in the implementation of the long time constant required for the smoothing filter. This is typically 20ms, corresponding to a -3dB frequency of 9Hz. Implementing this using switched-capacitor methods, whilst keeping the clock rate above the audio band requires a very large capacitor ratio. Offsets arising in this filter give the same problems as for offsets in the rectifier. Typically this filter requires the use of external components.

The above problems are exasperated when low voltage operation is required. Running the analogue circuitry at the same 3V level as modern digital circuits is very desirable.

An alternative solution is to digitise the input signal and digitally process the signal before reconverting it back into an analogue form. This solution would avoid all the problems which occur in analogue implementation, but at the expense of a high resolution ADC and DACs and significant digital circuitry. Providing 40dB SNR over a 30dB range would require ADC and DAC capable of 70dB dynamic range. This is at the limit of component matching accuracy of today's IC processes. For high yield, the ADC and DAC functions would be best implemented using sigma-delta modulators. The digital solution requires digital multipliers and additional decimation and interpolation filters which have significant cost. An advantage of the digital solution is that the DSP required could also be made to provide all the other base-band signal processing tasks required for the system such as filtering, signalling tone generation and detection. Whilst in the future this method is likely to become an economic solution, with today's technologies, analogue solutions are to be preferred.

4.3 SIGMA-DELTA BASED COMPANDER

A radically different topology for implementing the compander function was introduced in [9,10]. This utilises sigma-delta modulation to modulate one analogue signal into an equivalent oversampled single bit signal which may then be used to gate the other analogue signal and thereby performing an approximation to the multiplication operation.

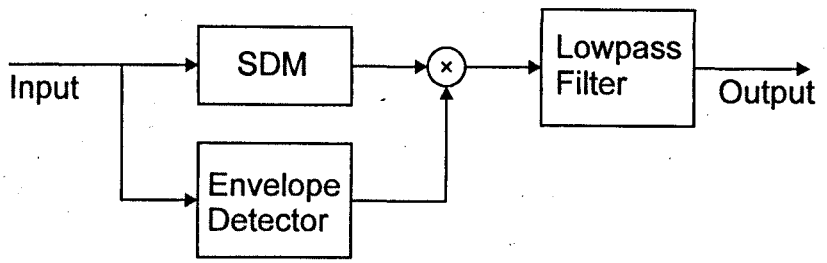


Figure 4-2 Expander using sigma-delta modulator.

The sigma-delta based expander is shown in Figure 4-2. The expander exploits the sigma-delta modulator as a multiplying DAC. The input signal is digitised by the sigma-delta modulator and is immediately reconverted back to analogue form using a two level DAC with reference proportionate to the signal level as measured by a full wave rectifier and filter combination. Small signals are thus made smaller by the use of a smaller reference, giving the desired expansion characteristic. The sigma-delta modulator operates on the speech signal over its compressed dynamic range.

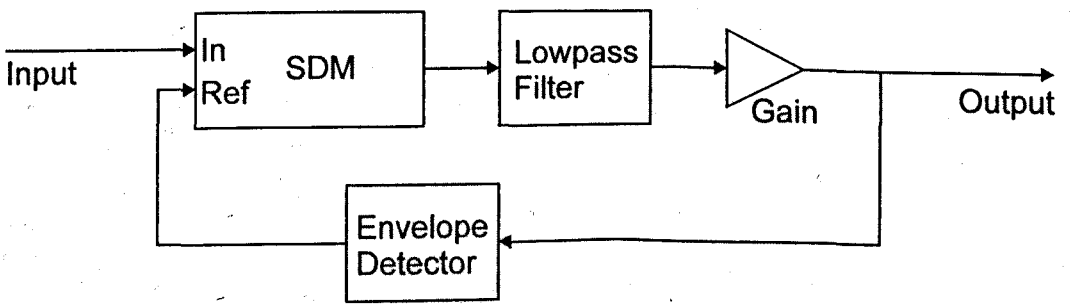


Figure 4-3 Compressor using sigma-delta modulator.

The compressor exploits the sigma delta modulator as a dividing ADC (Figure 4-3). For any ADC, a division operation can be performed by varying the reference. The compressor output signal level is detected using the envelope detector and this used to set the reference level to the sigma-delta modulator. The sigma-delta modulator output is converted back to analogue by a two level DAC with fixed references to form the output signal. The quantisation noise must be removed by the reconstruction filter before the output level is measured by the envelope detector. The sigma-delta modulator operates on the input signal which has the full uncompressed dynamic range, although for small inputs the reference is reduced by the system giving an effective sigma-delta operating range equivalent to the compressed dynamic range.

4.3.1 Disadvantages

Whilst elegantly solving the multiplication problem, there are still difficulties encountered when implementing this system. The problem of implementing the rectifier and time constant circuit with little offset remains but has been lessened in the reported implementation by using offset compensating SC techniques [9]. These techniques do not eliminate offset completely, and it is difficult to predict what the offset performance will be before fabrication. The long time constant which requires a large capacitance ratio has been implemented in acceptable silicon area by using a T-network [11] which has a parasitic sensitive node. A parasitic insensitive solution such as that given in [12] would further compound the offset problem.

The conversion of the analogue input signal to a digital form and back to analogue exposes the speech signal to all the quantisation noise inherent in the conversion process, together with other undesirable effects such as tones which plague the design of simple low order and low oversampling ratio converters.

A major disadvantage with the compressor is that the sigma-delta modulator operates on the uncompressed input signal and therefore requires a noise performance commensurate with the uncompressed dynamic range. It is true, however, that the for

smallest input signals, the reference to modulator is reduced by the system, such that the modulator only converts over the compressed dynamic range. This reduces the modulator order and the oversampling ratio required, but not the circuit noise performance required at the input of the modulator which must still be commensurate with the uncompressed dynamic range.

The entire compressor loop must have low offset unless AC coupling is possible. The gain required in this loop makes this requirement difficult to meet.

4.4 IMPROVED COMPANDER SYSTEM

An alternate topology for sigma-delta based companding has been developed [13]. This reduces the requirements on the sigma-delta modulator allowing the use of first order modulation to perform the signal multiplication. Additionally all the problems involved with analogue rectifier and smoothing filter may be completely eliminated by implementing this part of the circuit digitally. The complete compander system may be integrated on a pure CMOS process with no external components.

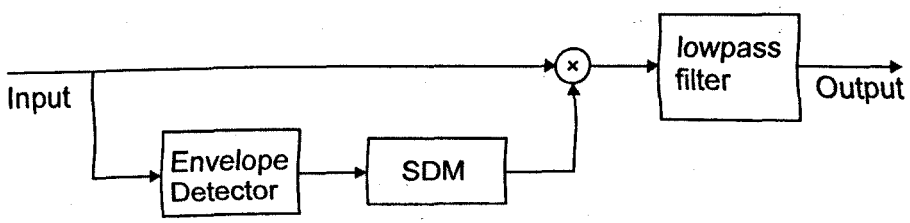


Figure 4-4 The improved expander.

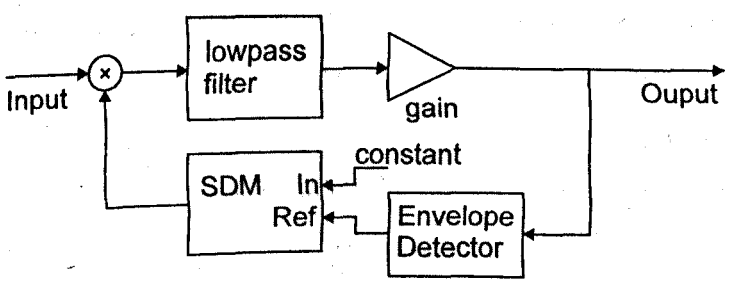


Figure 4-5 The improved compressor.

The significant difference with the new systems depicted in Figure 4-4 and Figure 4-5 is that it is the low bandwidth envelope signal which is sigma-delta modulated. The speech path remains completely analogue. The quantisation noise of the sigma-delta modulator which appears at the output is scaled by the amplitude of signal. When the signal disappears in the idle channel condition, the output of the sigma-delta modulator is not seen at the signal output. Since the multiplication of the speech signal and the sigma-delta modulator output can only attenuate the speech signal and the compressor is required to amplify low level signals, the compressor requires gain to placed after the low-pass filter. In practise this gain is incorporated in the filter.

On sub-micron CMOS processes it is efficient to digitise the input of the envelope detector and to perform the envelope detection digitally giving an easily predicted and repeatable level of performance. This digitisation can be conveniently implemented using an analogue sigma-delta modulator with a digital low-pass and decimation filter (Figure 4-6 and Figure 4-7). The signal being digitised has only the compressed dynamic range, and since it is only the signal level which is ultimately required, this part of the system is tolerant of spurious tones and distortion. The digitised signal may be high-pass filtered to remove any DC offset before the signal is digitally rectified. The time constant in the envelope filter can also be implemented digitally giving a very predictable performance. The complete system may be completely integrated with no external components such as AC coupling capacitors or time-constant capacitors required, saving not only the cost of the external component, but also the cost associated with the pad area and pin connections. This solution is very robust since it does not rely on a high level of component matching. Another significant advantage is that the test time can be considerably reduced as compared to analogue solutions. In particular, the circuits implementing the high pass and the low pass filters may be tested as digital circuits rather than as filters which takes considerable test time due to the long time constants involved.

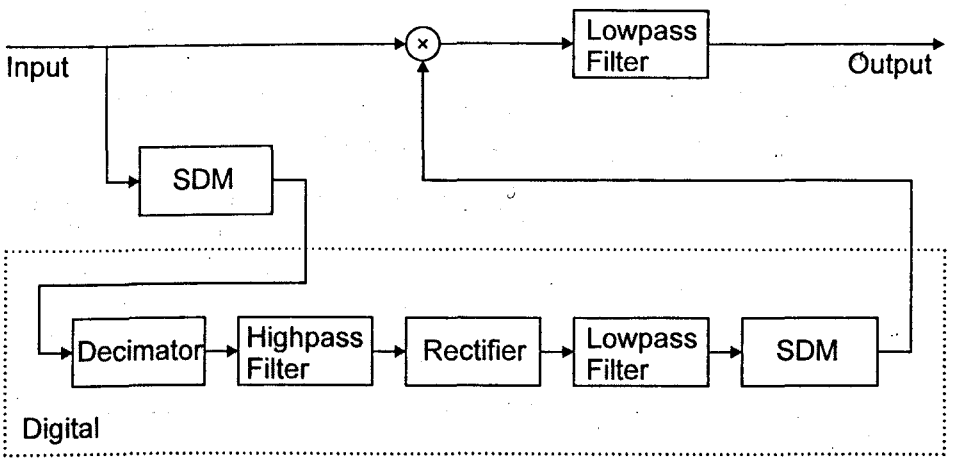


Figure 4-6 Mixed signal implementation of the improved expander.

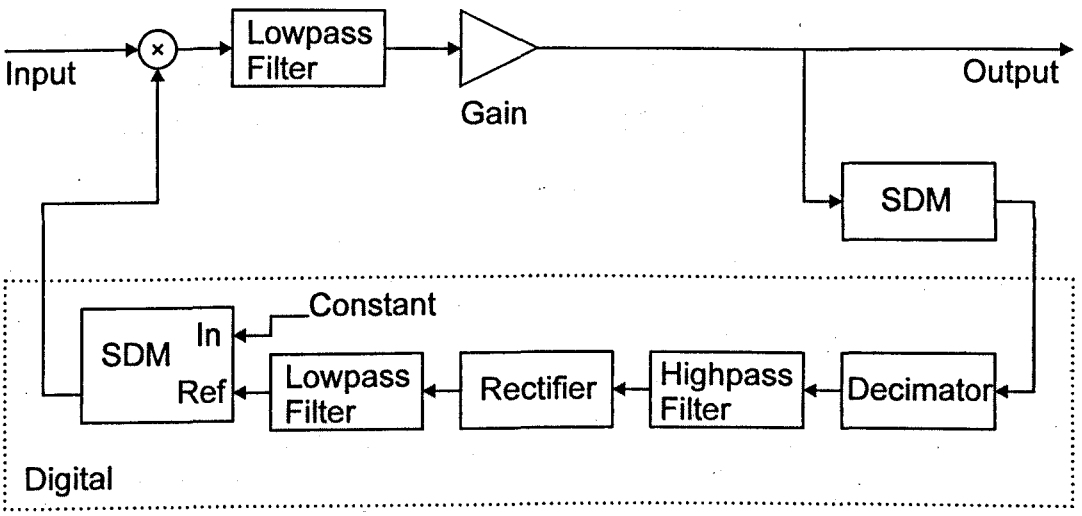


Figure 4-7 Mixed signal implementation of the improved compressor.

It may be noted that the compressed analogue signal is digitised in the compressor (Figure 4-7) and this system may be used as an ADC with a compressing action, rather than an analogue-in / analogue-out compressor.

4.4.1 Unipolar and Bipolar Sigma-Delta Modulation

At this point, an important distinction will be discussed between what will be referred to as bipolar and unipolar sigma-delta modulation. The sigma-delta modulation most commonly referred to in the literature assumes a bipolar input and output signal with normalised output levels of +1 and -1. An alternate form of

modulation accepts only positive inputs (unipolar) which are encoded such that the output represents normalised levels of +1 and 0. It is clear from Figure 4-8 that bipolar and unipolar sigma-delta modulation are identical, other than a trivial linear scaling and offset operation on the input and output signals. Hence the distinction is rarely made.

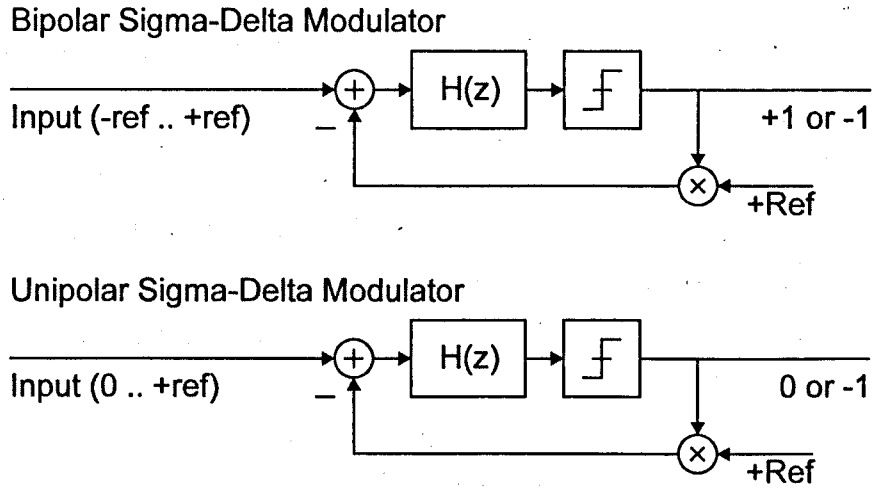


Figure 4-8 Difference between bipolar and sigma-delta modulator.

The input signal range for stable operation of high order sigma-delta modulators is centred between the two reference levels and does not normally include either reference level. Thus bipolar signals and bipolar references are usually considered so that the signal range is in the centre of the stable range. For first and second modulators, inputs covering the full range between the references may be implemented without instability problems. Thus first or second order modulation may be used to encode the unipolar signals over the range of zero to the reference level, and so are ideally suited for encoding the unipolar signal level quantity in the companding system. The performance of both first and second order modulators is severely degraded when operated very close to reference levels [14].

A key distinction between unipolar and bipolar modulation occurs when the sigma-delta modulated signal is used in a multiplier. In the bipolar case, the multiplier multiplies by either +1 or -1 and the output always represents the input with unity (magnitude) gain. In the unipolar case, the multiplier multiplies by either +1 or 0,

and the output is isolated from the input when multiplying by 0. This results in information loss and aliasing type phenomena may occur. However it is simpler to ensure that the gain of zero is accurate, rather than in the bipolar case where multiplication by zero relies on the cancellation of the positive and negative references. Although SC circuits may be designed whereby the same capacitor is used to multiply for both +1 and -1, so retaining good matching between the positive gain level and the negative gain level, it is difficult to predict how non-ideal and parasitic effects will degrade the performance.

Compander systems require good accuracy for multiplicands near zero to ensure good low level gain tracking and consequently the unipolar form of sigma-delta modulator is to be preferred. However consideration needs to be given to the aliasing effects.

4.4.2 Performance of first order sigma-delta modulation

First order sigma-delta modulation is simple enough for limited mathematical analysis. In the compander application it is the response to dc inputs and relatively slowly varying inputs which is of interest. The noise arising from first order sigma-delta modulation for different dc inputs is plotted in Figure 4-9 which shows that the in-band noise level is seen to be very dependent of the input dc level. Noise peaks occur close to where ever the ratio of the input level to the reference level forms a rational number with a small numerator and denominator [14].

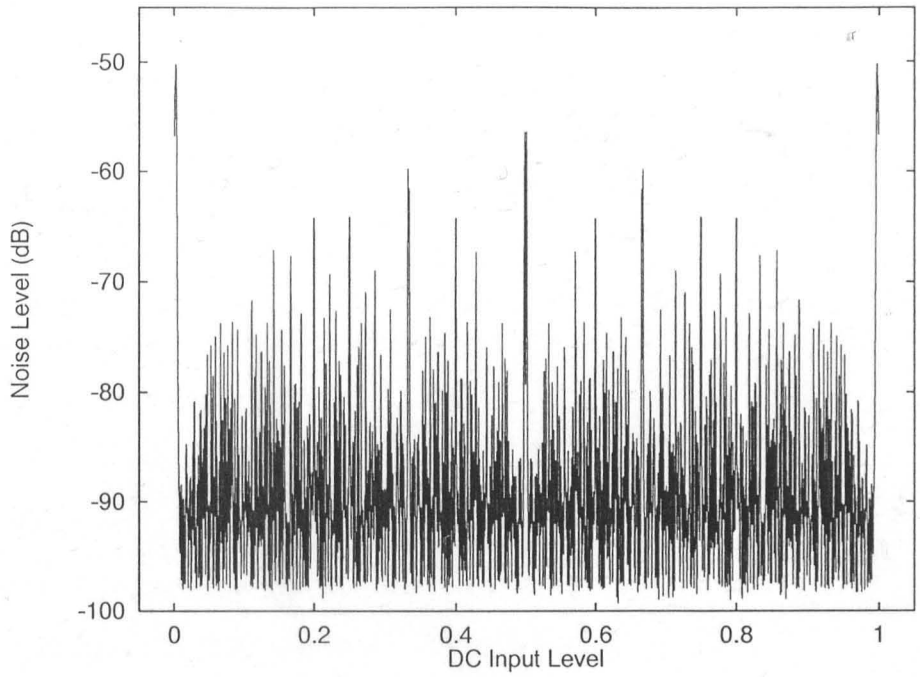


Figure 4-9 Noise from 1st order sigma-delta modulator.

The in band noise level is the highest when the fundamental component of the idling pattern lies in the signal band. The peaks near inputs of 0 and +1 are due to the long pattern of continued 0's with the occasional 1 for the case of dc levels near 0, and the long pattern of continued 1's with the occasional 0 for the case of dc inputs near +1. In the audio compander, the only stationary dc level expected is the idle channel condition, and this operating condition should be kept out with the extreme peaks in Figure 4-9. For the expander this can be achieved by using a sufficient oversampling ratio and deliberately providing a minimum digital sigma-delta modulator input. The idle channel tone problem does not arise for the compressor since in the absence of an input signal, the compander feedback loop will turn the gain to the maximum by closing the switch all the time.

When the graph of Figure 4-9 is plotted using a logarithmic axis (Figure 4-10), it can be seen that the peak noise power is proportionate to the input level (assuming that the large peak at very low amplitudes is not considered). This is an important observation since it implies that the minimum signal to noise ratio (SNR) is relatively constant over a wide range of input levels and this is a desirable characteristic for the compander application.

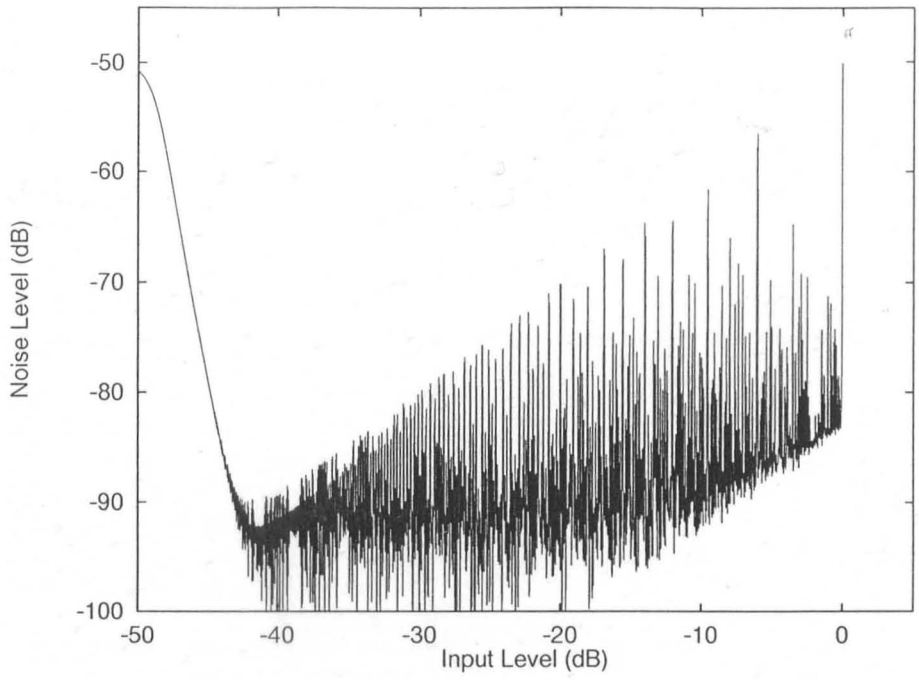


Figure 4-10 Noise from 1st order sigma-delta modulator (logarithmic axis).

4.5 SIMULATION RESULTS

The SDM compander was simulated using custom written C code models and a switched-capacitor simulator program. The C code was used to determine the SNR and output level for different input levels, while the SC program was used to verify the correctness of the implementation. These methods are discussed in Chapter 5.

The response to step changes in input level was verified using the SC simulation. The input signal, the output of the digital low-pass filter (the envelope signal) and expander output are shown in Figure 4-11, with closer detail in Figure 4-12. The ripple on the low-pass filter is clearly visible, this ripple modulates the speech signal giving rise to distortion products (this distortion is systematic and all companders exhibit it.).

The response of the compressor to step changes in input level is depicted in Figure 4-13.

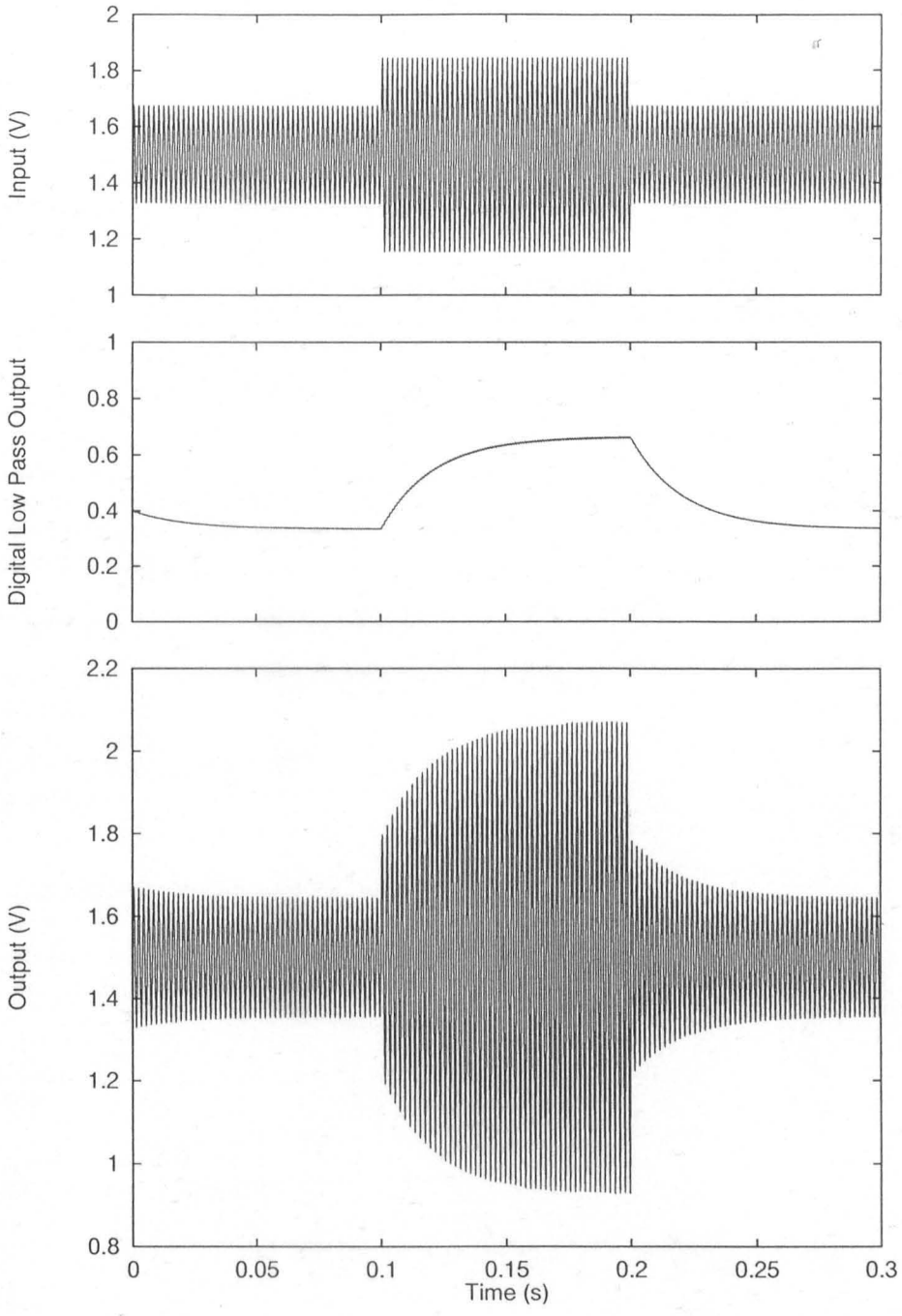


Figure 4-11 Simulated expander step response.

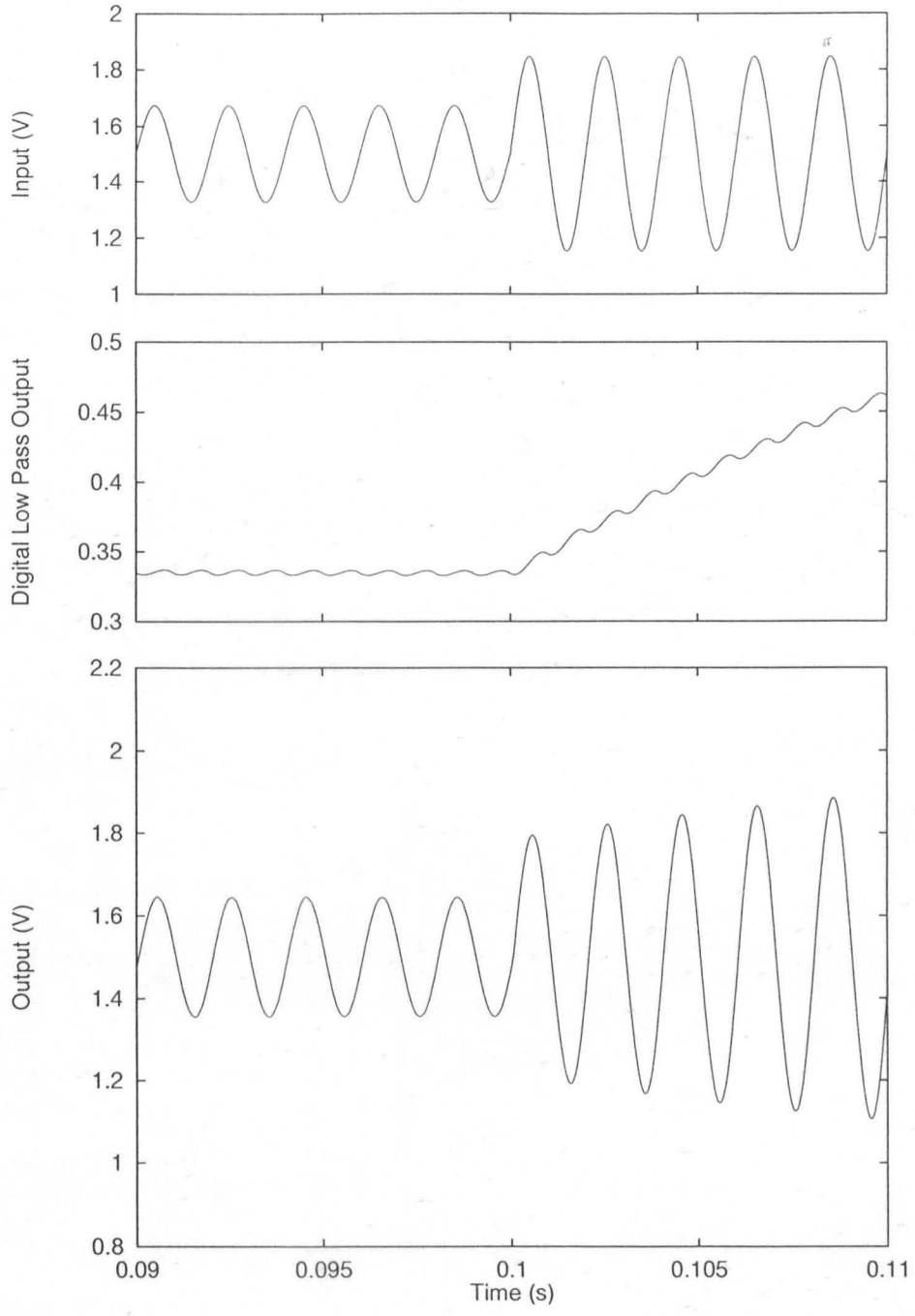


Figure 4-12 Expander step response (detail) showing ripple on digital output.

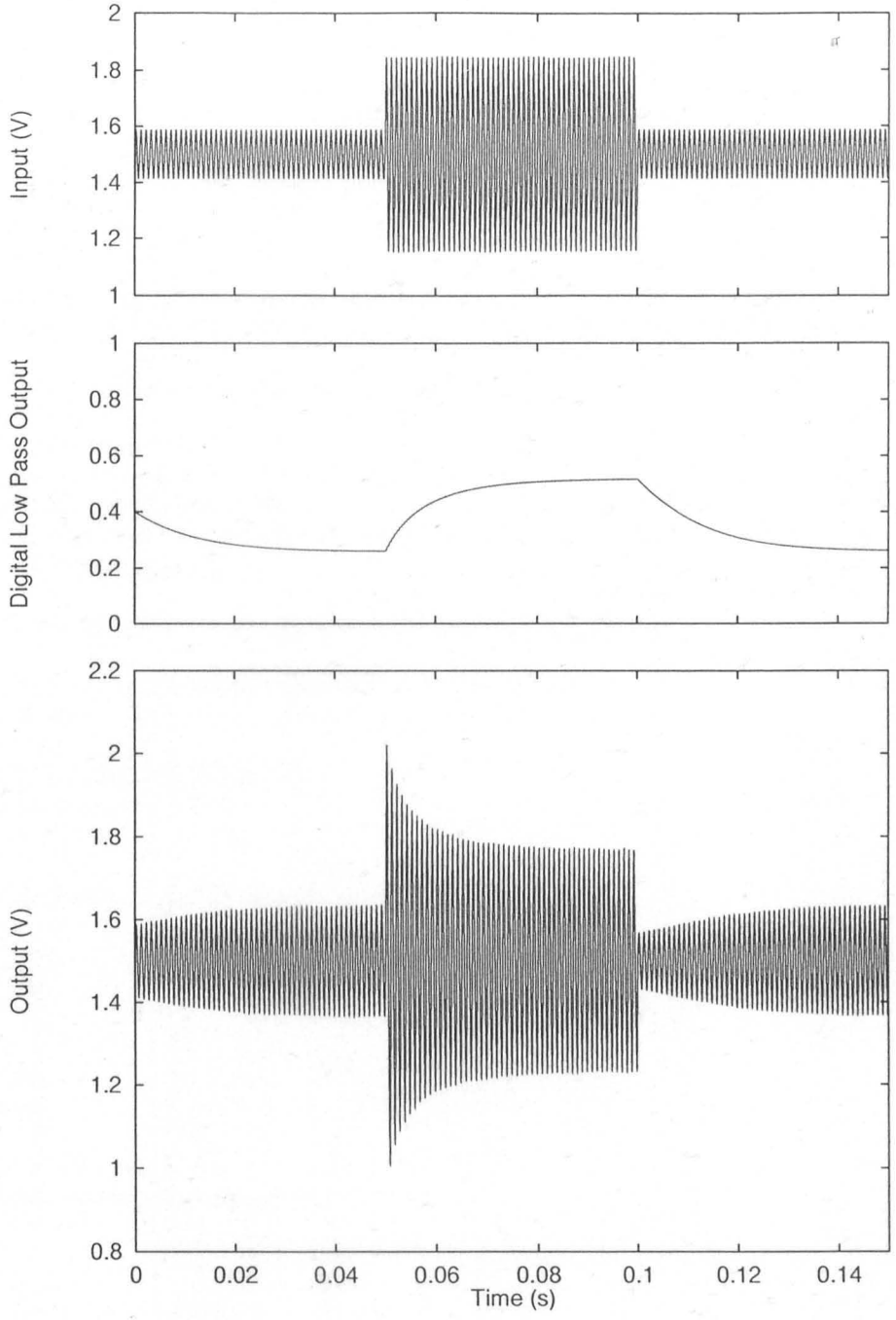


Figure 4-13 Simulated compressor step response.

The SNR and gain tracking error were simulated using a C program and shown in Figure 4-14 and Figure 4-15. These results are from a simulation representing three minutes of real time - over $2x10^8$ clock cycles. Figure 4-16 and Figure 4-17 are similar except that they are for the compressor path. The dotted lines in these figures show the required performance.

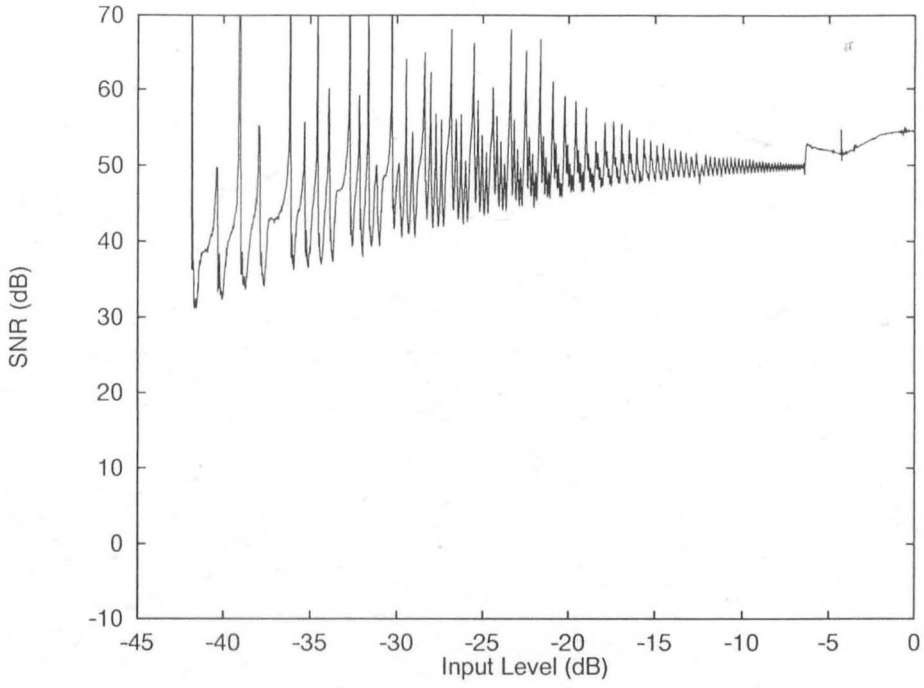


Figure 4-14 Simulated SNR of expander.

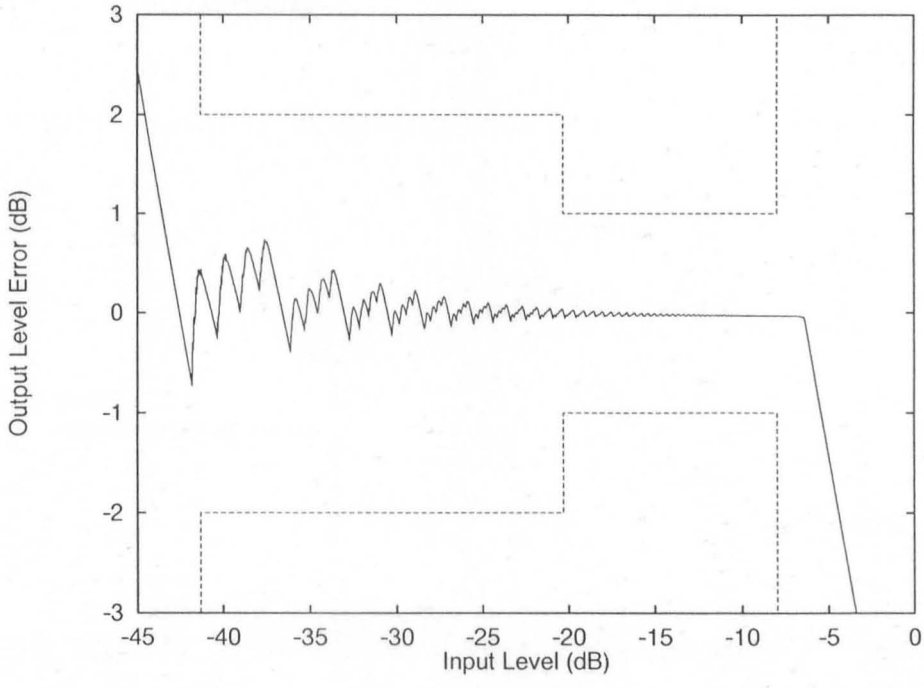


Figure 4-15 Simulated tracking error of expander.

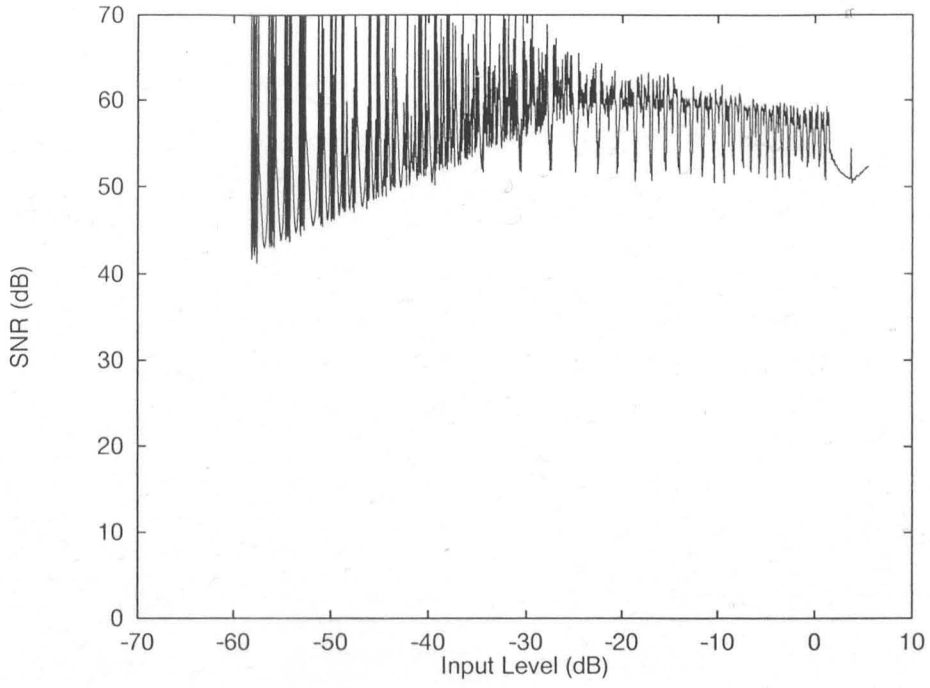


Figure 4-16 Simulated SNR of compressor.

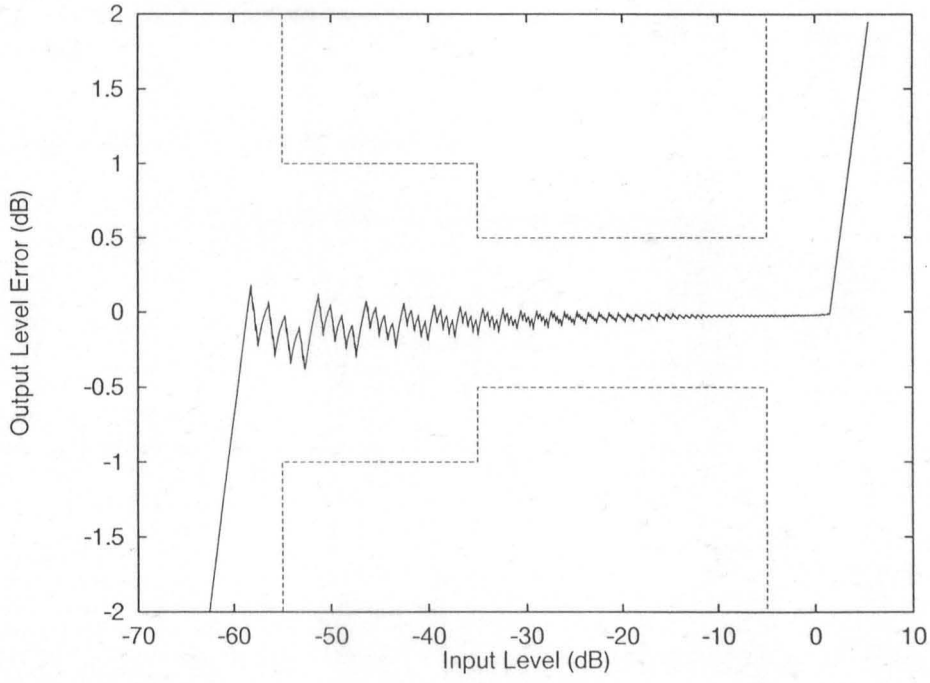


Figure 4-17 Simulated tracking error of the compressor.

The C code included modelling of the truncation effects in the digital filters. The saw-tooth patterns in the tracking errors are due to truncation effects in the digital implementation of the filters. If ideal digital filters are used the plots shown in Figure 4-18 and Figure 4-19 are obtained. The effects on SNR of the 1st order sigma-

delta modulator noise characteristics as shown in Figure 4-10 are clearly apparent. However the SNR is greater than 50dB for all input levels and this is more than sufficient to meet the system requirements. The tracking error shown in Figure 4-19 shows near perfect tracking over the operational input range, with only a slight deviation at low input levels which is due to the quantisation noise of the second order sigma-delta modulator used to digitise the compressor output.

Both the expander and compressor were designed for an uncompressed signal range of 60dB. At high signal input levels, the expander cannot apply more gain as the modulating switch is always switched on. At low signal levels the gain is held constant at -30dB to prevent tone effects. Likewise the compressor at low input amplitudes cannot apply more gain while at high amplitudes limiting occurs in both the analogue circuitry and the digital filters preventing further compressing action. Thus the range of expansion and compression is well defined.

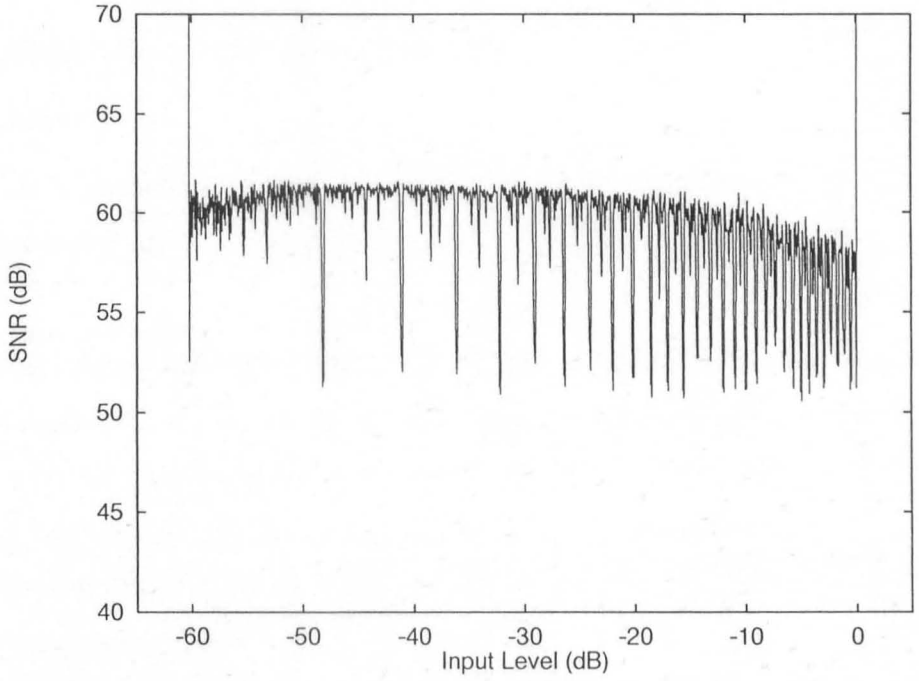


Figure 4-18 SNR of compressor with ideal digital filters.

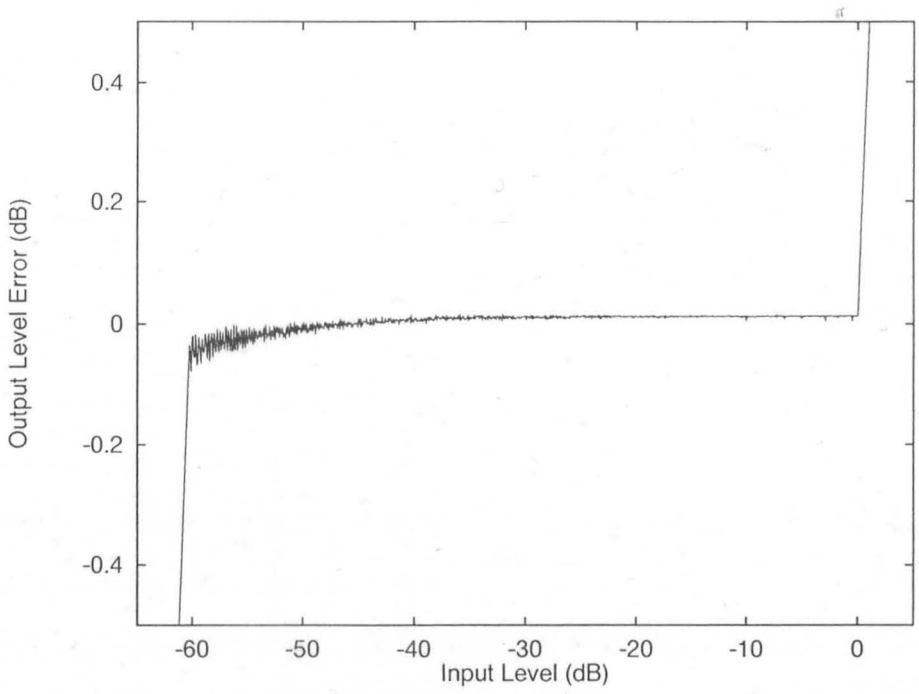


Figure 4-19 Tracking error of compressor with ideal digital filters.

4.6 PRACTICAL IMPLEMENTATIONS

The new sigma-delta based compander has been implemented as a breadboard prototype and has been integrated on silicon.

The breadboard prototype used an existing integrated second order analogue sigma-delta modulator, a discrete one bit multiplier, a continuous time reconstruction filter and a digital Field Programmable Gate Array (FPGA) to implement the logic. This functioned without any problems and met all the specifications required for analogue mobile telephony.

Two completely integrated versions of the compander have been implemented. These were on different 0.8μm pure CMOS processes with double level poly-silicon. These small geometry processes are ideally suited for efficient implementation of the digital logic required. Double level poly-silicon is required to implement good quality capacitors for the SC circuits.

The integrated compander requires no external components and therefore saves the AC coupling capacitor and the capacitor used to determine the attack and decay time-constants and bonding pads / package pins associated with conventional BiCMOS solutions [8].

4.6.1 ADC

The structure of both the compressor and the expander is such that the ADC used to determine the signal level operates on the compressed signal. This considerably reduces the performance requirement of the ADC. Since the input is being digitised solely to provide a digital estimate of the input signal level, the distortion and spurious tone performance is unimportant. Offsets on the input are removed by a digital high-pass filter and any offset from the ADC will be similarly eliminated. The main requirement for the ADC is that it must provide enough resolution to determine the signal level to within 1dB for the smallest input signal, typically -30dB relative to the largest signal. This implies a resolution and noise performance of 10 bits. The simplest method to achieve this is with an analogue sigma-delta modulator, followed by a decimation filter. A successive approximation ADC could also be used, but this is less robust since it relies on element matching and gives no aliasing rejection.

An ideal first order sigma-delta modulator was simulated in the system and found to be adequate. However first order modulators suffer from noise leakage if the integrator gain is finite. A second order modulator is less sensitive to noise leakage since two opamps are used to provide the loop gain. A second order modulator was designed using a multiplexed opamp. The use of a multiplexed opamp does not require an increase in switching frequency and so the area and power consumption are halved. To reap the full benefits of second order modulation, the comb decimation filter requires to be of third order [15]. However for the oversampling ratio used in the compander design, a second order comb filter is sufficient. Thus the cost in terms of area and power consumption of the second order modulator over the first order modulator is insignificant.

The new topology shown in Figure 4-21is derived from the classic two opamp second order SC implementation (Figure 4-20). The switch phasing of the first integrators is changed without altering the response of the circuit. However, in each phase, only one of the two amplifiers is active. This allows a single amplifier to be multiplexed in a similar manner as used in differential SC biquads [16]. The resulting circuit has the advantages of reduced silicon area and power dissipation.

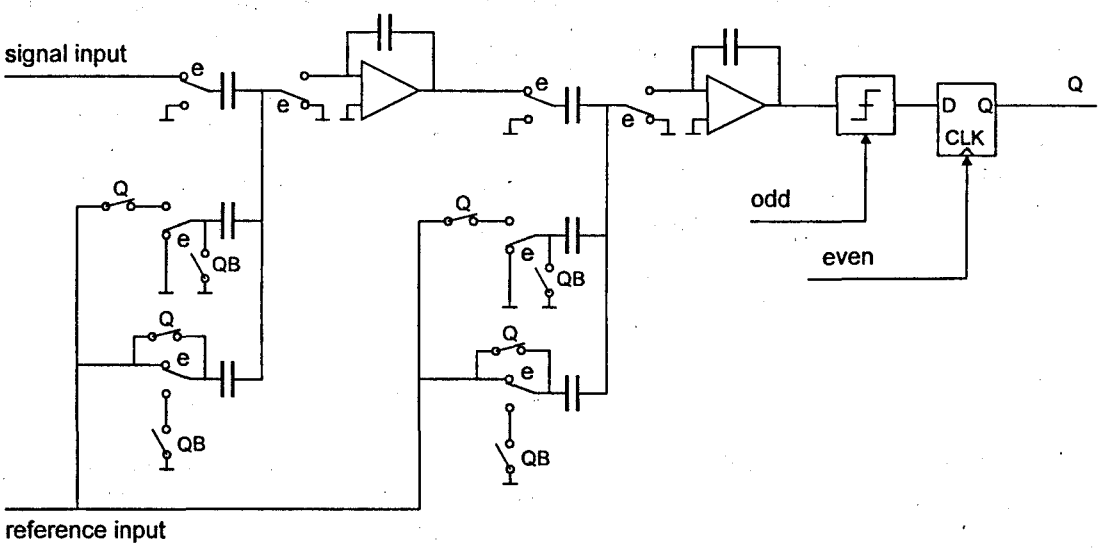


Figure 4-20 Classic second order SC sigma-delta modulator.

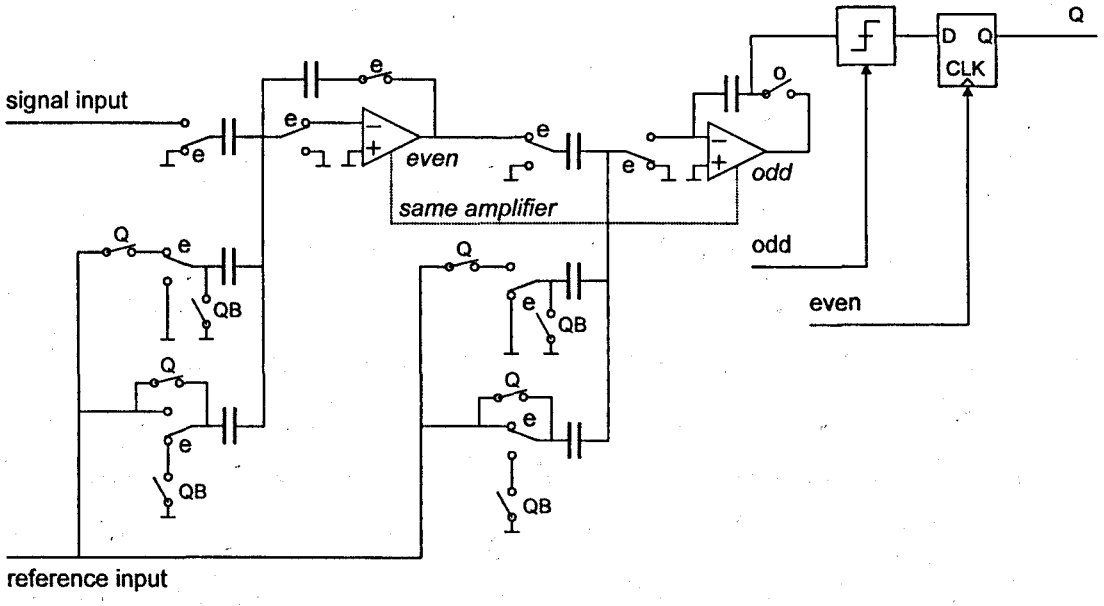


Figure 4-21 Modified second order SC sigma-delta modulator using single opamp.

The schematic for the second order sigma-delta modulator with multiplexed opamp is included in Chapter 5. The circuit is fully differential to reject noise from the digital circuits on the same substrate. The opamp is a single stage folded cascode design using switched capacitor common-mode feedback [17]. A single voltage reference is used. The sigma-delta feedback circuitry could use fewer feedback capacitors but this approach allows the use of a simpler non-overlapping clock generator. The current drawn from the reference, which in this case is the mid-rail voltage is not signal dependent which minimises coupling of the feedback signal to other parts of system using the same reference. It also reduces distortion due to multiplicative effect of the signal modulating the reference, though low distortion is not important in this application. Since good distortion performance is not a requirement, delayed clocking schemes were not adopted [18].

Simulation of this circuit at a high level is necessary to prove that it is indeed a second order modulator. If for example, the comparator is clocked on the wrong phase, it will give an output corresponding to the polarity of the output of the first stage and the resulting system would be a first order sigma-delta modulator. This error would be difficult to notice from a SPICE level simulator over a few tens of clock cycles.

4.6.2 Reconstruction Filter

Because the companding function was required to be an individual component on both IC implementations, the reconstruction filter was not merged with the band-limiting filter specified in the analogue cellular standards. Instead simple second and third order Butterworth filters were implemented. These use fully differential switched capacitor techniques. The second order implementation is shown in Figure 4-22. The one bit multiplier is built into the front end of the filter. Since the differential single stage folded cascode amplifier used cannot drive resistive or external loads, the configuration around the output opamp has been modified to use two single ended two stage amplifiers which are capable of driving resistive and

external loads. The filter is a differential implementation of the F-damped biquadratic structure introduced in [19].

It is important that the analogue input signal contains no energy at high frequencies since high frequency components will mix with the high frequency components in the digital signals to give rise to base-band tones and noise.

4.6.3 Digital circuits

The digital system was implemented using bit-serial techniques except for the first half of the decimator and the output sigma-delta modulator which operate at the oversampled rate and are implemented using parallel logic. Because most of the system comprises of shift registers with relatively few taps used, the logic can be laid out very efficiently even when an auto-routing tool is used. Since the oversampled clock rate of 960kHz is very much lower than the clock frequencies available, it would have been possible to multiplex the digital logic of the compressor and the expander. However since much of the logic are flip-flops holding state information, the savings would not have been particularly great. The clock rate is relatively low for a sub-micron process which allows use of minimum geometry logic cells to be to reduce die area. The digital circuitry is implemented in approximately 2000 gates for each of the compressor and expander.

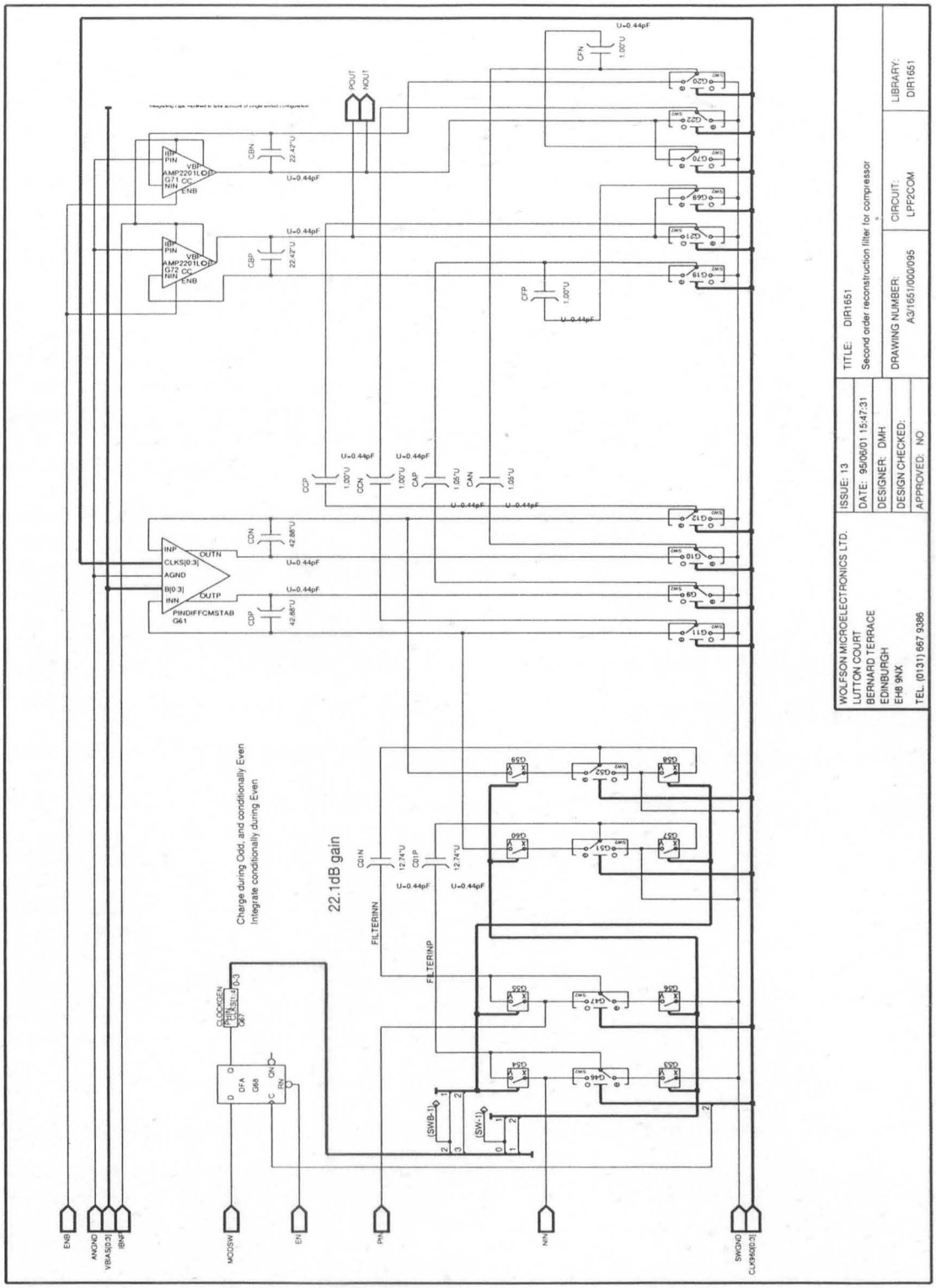


Figure 4-22 Schematic of second order low-pass filter.

The digital section has a single bit sigma-delta modulated input signal and a single bit sigma-delta modulated output. These two lines, together with a clock are the only signals which need to be passed between the analogue and digital parts of the companding system which allows the digital circuits to be placed remotely on the IC die from the noise sensitive analogue circuits.

The only difference between the digital logic required for the compressor and the expander is in the output sigma-delta modulator. The expander sigma-delta modulator has a constant reference, whereas the compressor sigma-delta modulator has a constant input signal and a variable reference level.

4.6.4 Decimation Filter

The decimation filter implements a second order comb filter which reduces the sample rate by 24 whilst increasing the word-length from one bit to 11 bits. The comb filter has a FIR response, but is implemented as a cascade of a recursive section operating at the input sample rate and a non-recursive section operating at the output rate [15].

4.6.5 High Pass Filter

The high pass filter mimics the function of an AC coupling circuit by eliminating any DC component present on the input signal or arising from the analogue sigma-delta modulator, and in the case of the compressor, arising in the SC reconstruction filter. The main requirement is good rejection at DC and little attenuation over the speech band. A first order bit serial high pass filter was implemented with a -3dB frequency of 50Hz.

4.6.6 Absolute Value Circuit

The absolute value circuit performs a one's complement negation on the input signal if the msb (sign bit) is set. This requires a shift register delay line for the bit serial data, since the sign bit in bit-serial arithmetic is the last in the sequence. The output of this circuit is an unsigned quantity with half the range of the input.

4.6.7 Low Pass Filter

The low-pass filter smoothes the rectified signal to provide the indication of input signal level. The time constant of this circuit is derived from the step input level tests specified in the compander system specifications. The time constant required is 20ms corresponding to a -3dB frequency of 9Hz. Truncation effects in this filter give rise to gain tracking errors for low input amplitudes. The word-length used was 18 bits, but higher word-lengths would significantly reduce the low level tracking error with only a slight increase in hardware.

4.6.8 Expander Sigma-Delta Modulator

The expander sigma-delta modulator takes the unsigned output of the low-pass filter and latches it. No further interpolation is required since the images arising from the increase in sample rate are out-with the audio range and will be attenuated by the reconstruction filter.

The sigma-delta modulator produces a single bit stream with a ones density proportionate to the input word. The first order modulator is particularly simple to implement consisting of a single adder and clocked latch.

4.6.9 Compressor Sigma Delta Modulator

The compressor sigma delta modulator is slightly different to the expander sigma-delta modulator since it is required to produce an output bit stream with a one density *inversely* proportionate to the input word. This is again implemented using a first order modulator except that the input level is fixed at 1/32 and the reference is varied between 1/32 and 1. If the input used for the reference signal falls below 1/32 the output sticks at 1 and the compressor is in maximum gain mode. The range of 1/32 to 1 gives a gain range of -30dB to 0dB.

4.7 TEST RESULTS

The specifications required of compressors and expanders typically include the gain tracking error, the signal to noise ratio, distortion, and the attack and recovery times. All the measurements discussed below were taken from the first silicon implementation operating on a 3.0V supply.

4.7.1 Expander

The expander attack time is defined as, the time taken for the envelope of the output to reach 0.75 times the final steady state value, after an increase in the input by +6dB step. Expander recovery time is defined as, the time taken for the envelope of the output to settle to 1.5 times the final steady state value, after a decrease of -6dB at the input. The oscilloscope traces below show the expander output, for the specified ±6dB steps, and a plot showing the complete envelope of the output for a repetitive signal. These are equivalent to the simulated traces shown in Figure 4-11.

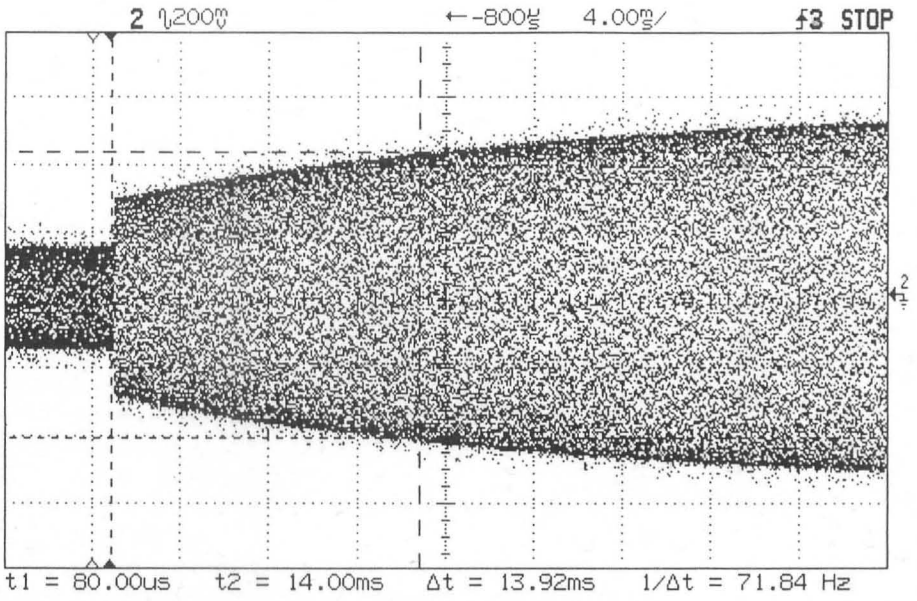


Figure 4-23 Expander attack measurement.

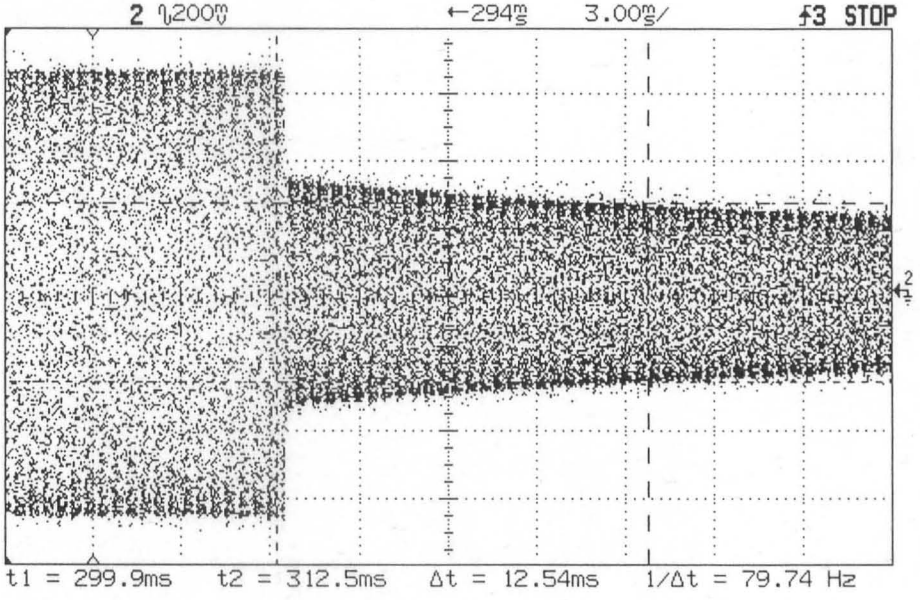


Figure 4-24 Expander recovery time.

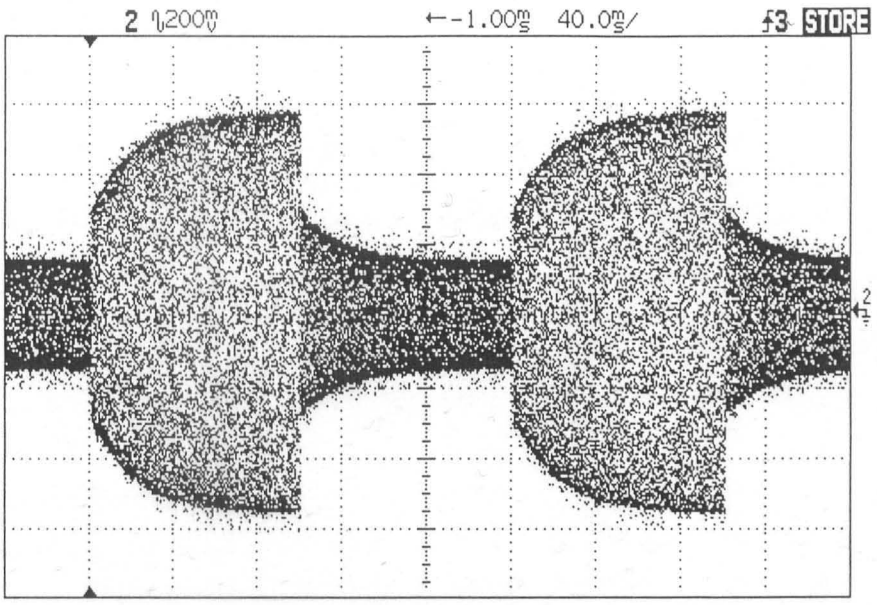


Figure 4-25 Expander response to sinusoid output with a repetitive step change in input amplitude of ±6dB.

The tracking error was measured and gives a curve almost identical to the simulated one (Figure 4-15) confirming that the simulation methods used were valid.

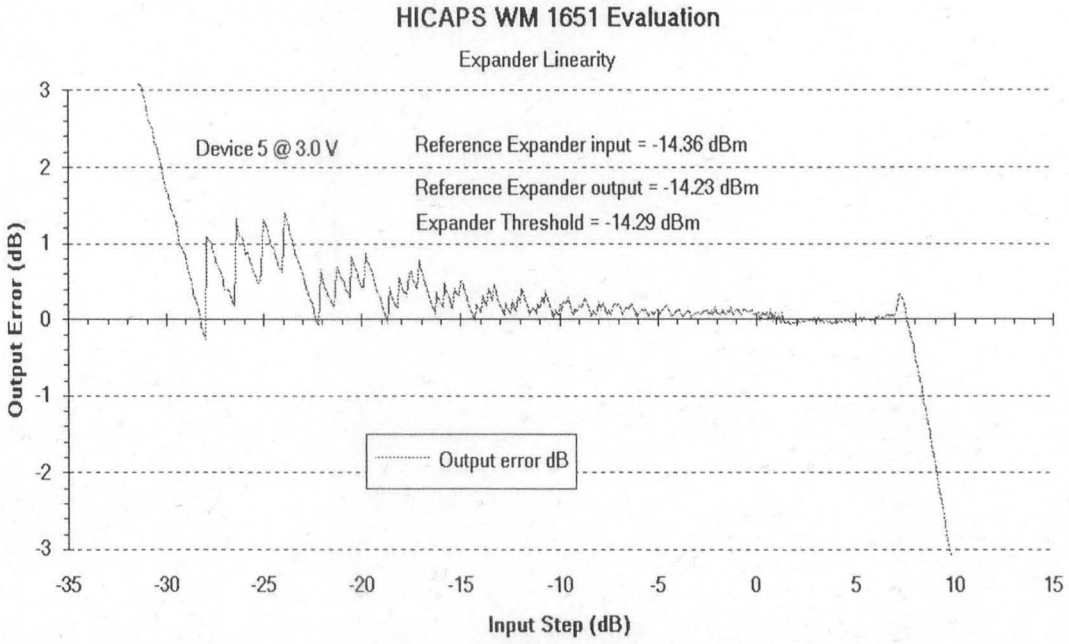


Figure 4-26 Measured gain tracking error of the expander.

4.7.2 Compressor

Compressor attack time, is defined as the time taken for the envelope of the compressor output to decay to 1.5 times the final steady state value, after a +12dB input step. Compressor recovery time, is defined as the time taken for the envelope of the compressor output to reach 0.7 times the steady state value, after a -12dB input step. The following traces are the measured equivalents of the simulated traces which were shown in Figure 4-13.

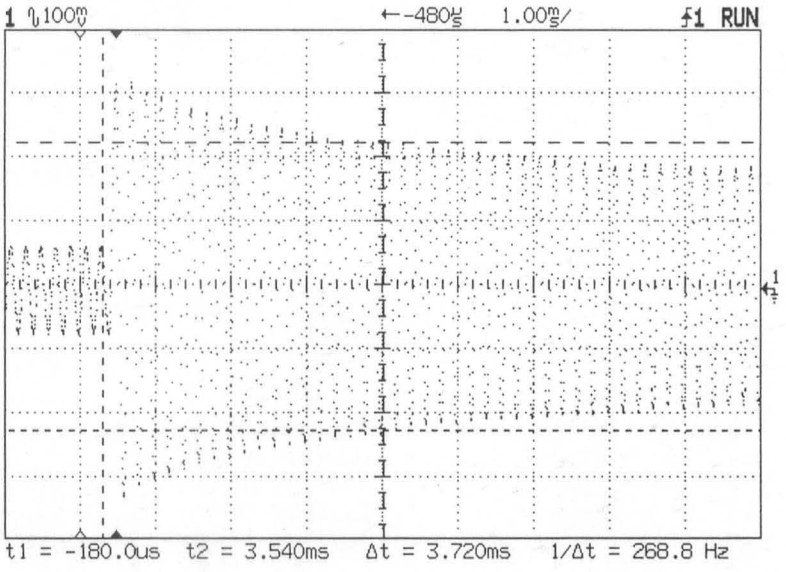


Figure 4-27 Compressor attack.

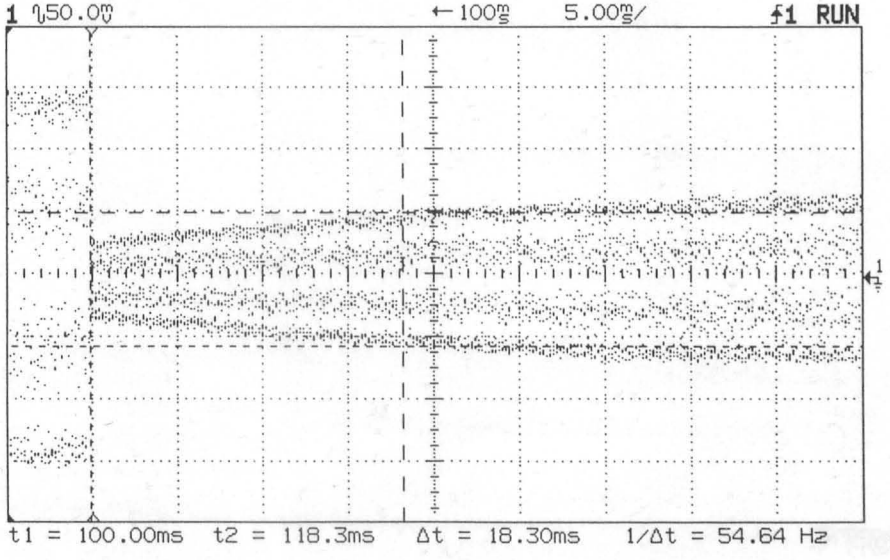


Figure 4-28 Compressor recovery.

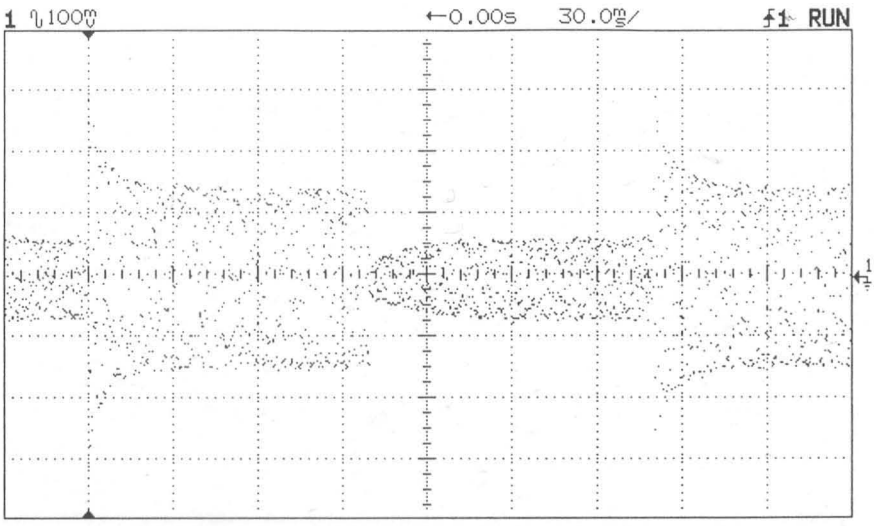


Figure 4-29 Compressor response to input with repetitive change in input amplitude of ±12dB.

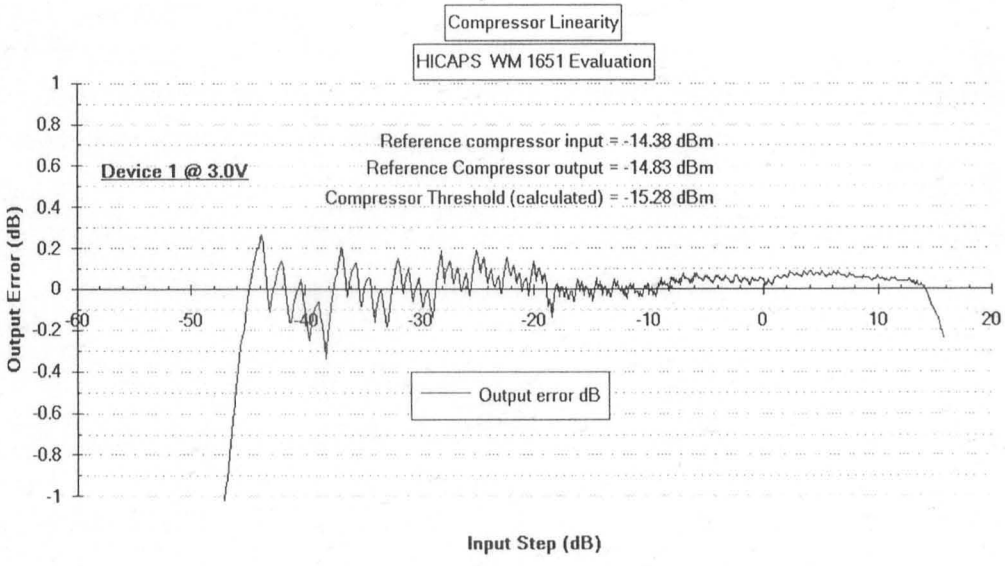


Figure 4-30 Measured compressor tracking error.

4.7.3 Test Issues

The improved sigma-delta based compander structures have several benefits when considering production test. The robustness of the system means that accurate measurement of such parameters as offsets and gains are not necessary. A significant

advantage arises from the fact that the time constants are implemented digitally, and functionality can be checked as a digital system with out waiting for several time constants for the values to settle [20].

4.8 SUMMARY

The compander based on sigma-delta modulation has been designed, simulated, and fabricated with good success. The test results measured were very close to the simulation. This new topology is therefore a good choice for implementing analogue companders on modern CMOS processes.

4.9 REFERENCES

[1] CCITT specification G162, "Characteristics of Compandors for Telephony," Geneva 1964.

[2] "Applications for Compandors," Application Note AN174 in *Linear LSI Data and Applications Manual*, Signetics, California, pp. 9.156-9.164, 1985.

[3] S. Shioda et al, "1.8V compander with 80dB dynamic range paper," *IEEE Custom Integrated Circuits Conference*, pp. 155-159, 1993.

[4] "Compandor NE570/571/SA571" data sheet in *Linear LSI Data and Applications Manual*, Signetics, California, pp. 5.3 - 5.8, 1985.

[5] "Compandor Cookbook," Application Note AN176 in *Linear LSI Data and Applications Manual*, Signetics, California, pp. 9.166-9.173, 1985.

[6] B. Gilbert, "A Precise Four-Quadrant Multiplier with Subnanosecond Response," *IEEE Journal of Solid State Circuits*, vol. 3, pp. 365-373, December 1968.

[7] P.R. Gray and R.G. Meyer, *Analysis and Design of Analog Integrated Circuits* (Third Edition), Wiley, New York, 1993.

[8] "Advanced Analogue Processor for Cellular Radio: TACS, AMPS and NAMPS", Data Sheet for WM8010, Wolfson Microelectronics Limited, Edinburgh, 1994.

[9] Karou Takasuka et al, "A Sigma-Delta Based Square-Law Compandor," *IEEE Custom Integrated Circuits Conference*, Boston, Massachusetts, pp. 12.7.1-12.7.4, 1990.

[10] Huang Q., "Monolithic CMOS Compandors Based on ΣΔ Oversampling," *IEEE International Symposium of Circuits and Systems*, San Diego, California, pp. 2649-2652, May 1992.

[11] P. Van Peteghem and W. Sansen, "T-Cell SC integrator synthesizes very large capacitance ratios," *Electronics Letters*, vol. 19, pp. 541-544, July 1983.

[12] K. Nagaraj, "A Novel Parasitic Insensitive Switched Capacitor Technique for Realizing very Large Time Constants," *IEEE Custom Integrated Circuits Conference*, Rochester, New York, pp. 12.4.1-12.4.4, 1988.

[13] D.M. Hossack (Wolfson Microelectronics), British Patent Application.

[14] J.C. Candy and O.J. Benjamin, "The Structure of Quantisation Noise from Sigma-Delta Modulation," *IEEE Transactions on Communications*, vol. COM-29, pp. 1316-1323, September 1981.

[15] J.C. Candy, "Decimation for Sigma-Delta Modulation," *IEEE Transactions on Communications*, vol. COM-34, pp. 72-76, January 1986.

[16] M. Ismail, T. Fiez, *Analogue VLSI Signal and Information Processing*, Magraw-Hill, pp. 429-432, 1994.

[17] K.R. Laker and W.M.C Sansen, *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, 1994.

[18] B.E. Boser and B.A. Wooley, "The design of Sigma-Delta Modulation Analogue-to Digital Converters," *IEEE Journal of Solid State Circuits*, vol. SC-23, pp. 1298-1308, December 1988.

[19] P. Fleischer, K. Laker, "A Family of Active Switched Capacitor Biquad Building Blocks," *Bell System Technical Journal*, vol. 58, pp. 2235-2269, December 1979.

[20] M. Mahoney, *DSP-Based Testing of Analog and Mixed Signal Circuits*, IEEE Computer Society Press, Washington DC, 1987.

5

SIMULATION METHODS

5.1 INTRODUCTION

This chapter discusses simulation methods for systems incorporating sigma-delta modulators. Analogue implementations of sigma-delta modulators are most often implemented using switch-capacitor (SC) techniques. Digital implementations are more easily simulated, so most of this discussion concentrates on fast SC simulation.

Figure 5-1 shows an outline of the design flow employed during the design of integrated circuits (IC). The first simulation task is at a high level to determine the most suitable and cost-effective topology for the integration of the system. Once the high level design process is complete, a schematic database is constructed. This database incorporates the complete circuit diagram of the IC. The database provides a key role in ensuring the integrity of the design, both that the simulations accurately reflect the circuit and that the layout is correct. The goal is to check this database as thoroughly as possible whilst leaving the least opportunity for errors. This requires much automation of the checking process. Netlists for simulation are automatically generated from the database. The final layout information is verified using a layout versus schematic (LVS) program - again ideally completely automated. A design rule check (DRC) program is used to ensure the components are correctly drawn on the layout. Thus the integrity of the schematic database is key to correct implementation and often simulations are used to verify the correctness of the schematics rather than to determine performance figures.

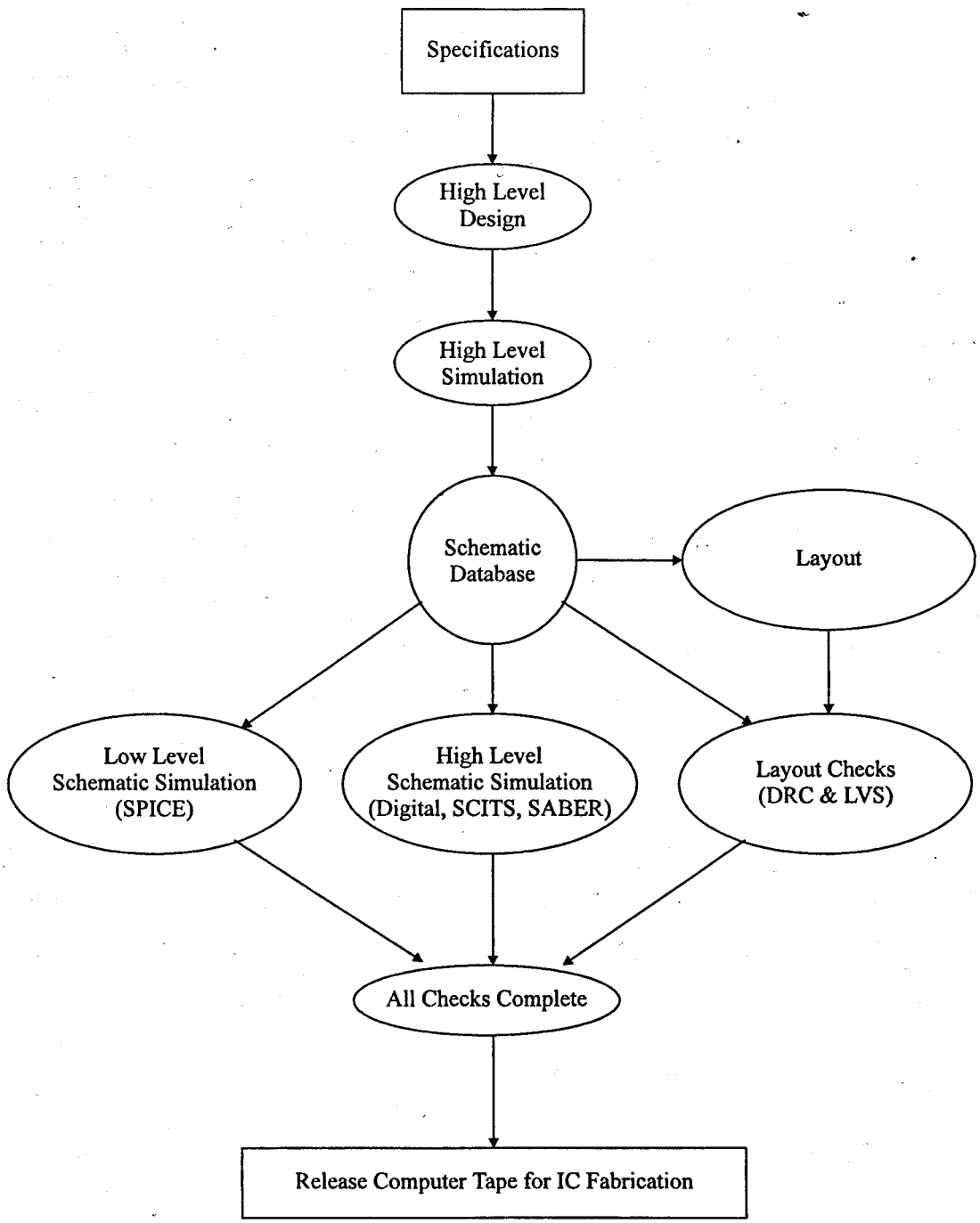


Figure 5-1 Design flow showing importance of the schematic database.

Accurate simulation of switched capacitor circuits is a difficult and computationally intensive task. Switched capacitor sigma-delta modulators are even more computationally intensive since they operate at high sample rates in comparison to the frequencies of the signals of interest. Traditional circuit simulation programs such as SPICE for analogue circuits are inappropriate as complete characterisation of

a sigma-delta modulator based system requires many millions of samples to be computed.

Initially what is required from simulation is an indication of whether a particular design is capable of meeting the required performance. Often this is done by coding a custom simulation program in a standard high level computer language such as C. The main advantage of this technique is the speed at which the resulting compiled code executes. The disadvantage however, is the significant time required initially to write the code. Another problem with this technique is that the resulting code bares no resemblance to the circuit topology. This is a result of the fact that most computer languages are procedural in nature whereas a circuit operates in parallel. Manual conversion of a signal processing block diagram into a procedural step-by-step list of computer instructions is tedious and error prone, especially where closed loops are present and the precise delays around loops are critical.

A more appropriate description of the system can be achieved using a declarative or functional language where each statement describes a rule which is always valid. In a declarative language statements can be considered to be running simultaneously in parallel as would all the individual parts of the real system. This allows the description of discrete-time data-flow systems. A large amount of commercially and non-commercially available software exists for simulating digital signal processing flow diagrams which may be directly used for sigma-delta modulator based systems. Special techniques may be used to speed up simulation for multirate and synchronous systems [1,2].

An object-orientated programming language such as C++ is more appropriate for high level simulation work because of the one-to-one correspondence of the concepts of defining and instantiating a component sub-circuit and the programming concept of defining an object and its associated methods. Furthermore C++ offers the facility for user-defined arithmetic which can allow a very high level (whilst also very efficient) description of digital data-paths. Models of digital filters and digital sigma-delta modulators may be simply coded which include all the effects of word-length

truncation and overflow. It is a simple matter to verify accuracy down to the least significant bit by comparing simulated results with those obtained using a digital simulator. Once the conformance of the model has been proved, the model may be used to determine the performance characteristics such as the signal-to-noise ratio and dynamic range. Verifying the conformance of a C or C++ model to an analogue simulation result is much more difficult. More extensive simulation of analogue circuitry is required before confidence in the equivalence of the models can be reached.

5.2 SC SYSTEM VERIFICATION

Most existing SC analysis programs were developed solely for SC based filters. They are invaluable for verifying the frequency and time-domain response. Analysis programs for SC filters are not applicable for sigma-delta modulators because they require that the switch controlling signals to be repetitive and specified before the simulation commences. With a SC sigma-delta modulator some of the control signals are a function of internal voltages. Since a sigma-delta modulator contains a non-linear component (the quantiser), the concept of a frequency response is not strictly applicable. However correct functionality can be verified through time domain simulation. With modern engineering workstations it is now realistic to simulate SC circuits using transistor level simulators such as SPICE over a small number of sample periods. However it is not realistic to use SPICE to verify that, for example, a third order sigma-delta modulator is not actually a second or first order modulator due to incorrect implementation. Time domain SC simulation closes this gap between low level and high level simulation. This level of simulation is useful for studying internal signal swings since it retains a speed advantage whilst retaining a close correspondence to the actual circuit. The netlist presented to the simulation may be obtained automatically from the schematic data base allowing verification of the integrity of the database. Simulation at the SC level is also invaluable for determining the effects of finite gain, voltage offsets, comparator hysteresis, component mismatch and parasitic capacitance.

5.2.1 Existing SC Simulation Tools

A number of software packages have been developed which are suitable for high level simulation of switched-capacitor sigma-delta modulators. The popular linear SC analysis program SWITCAP [3] was extended to allow transient simulation of systems with internally controlled switches [4]. The approach in [5] is unusual in that graph methods are used to formulate the circuit equations. The result is a model which may be used for chip-level simulation. Another approach has been to extend an analogue simulator to make it more suitable for discrete-time signals and switched-capacitor circuits in particular [6]. An approach specifically developed for sigma-delta modulators uses state transition tables to speed up simulations [7]. Large tables are compiled from running transistor level simulations to give mappings from the present integrator voltage to the next discrete time voltage state due to the integration of charge on the input switched capacitors. The switched network simulator, SCNAP5, includes the effects of resistance and finite amplifier bandwidth and has been coupled to a commercial digital simulator [8] to allow mixed SC and digital simulation.

The method discussed here was developed to provide a fast method for verifying the correctness of SC schematics and also to allow fast simulation of systems including limited types of non-ideal effects such as parasitic capacitances between nodes, finite amplifier gain, comparator hysteresis. The method allows a C program to control the switch states so that systems with significant digital circuitry such as the sigma-delta compander could be simulated quickly. Accurate estimation of performance degradation due to resistance and finite amplifier bandwidth and component non-linearity is not attempted.

Since SC circuits are almost always designed to have accurate settling by the end of each clock phase, the assumptions of zero switch resistance, zero amplifier output impedance and infinite amplifier bandwidth are not severe and they allow circuit simulation many orders of magnitude faster than traditional circuit simulators such as SPICE.

There are two principal contributors of non-ideal effects which must be considered during the design of SC circuits. The first is the linear errors due to non-zero switch-on resistance and finite bandwidth amplifiers. These give rise to settling-time related errors which can be detected as a worsening in performance as the sample rate is increased. The second is due to non-linear component characteristics such as non-linear amplifier transfer characteristics, voltage dependant switch resistance and charge injection effects. These are invariant to changes in sample rate.

Unfortunately both of the above error sources interact and both must be considered together if more accurate simulation is required. This mandates the use of programs such as SPICE which solves sets of non-linear stiff partial differential equations. Whilst it is shown in [9] that the settling time constant can almost be as long as the duration of the a clock period, in practice the settling must be much more accurate since any settling-related non-linear effects (eg. amplifier slew) will ruin the performance of the system [9, 10]. Therefore SC circuits must be designed to have much more accurate settling than that allowed by linear settling requirements and thus simulations incorporating only non-ideal switch resistance and amplifier bandwidth are not particularly useful. Recently a formulation has been reported which allows non-linear components, but making the assumption that the SC circuit will settle completely on every clock phase [11]. This can only give the performance for the limiting case of low sample rates. Accurate simulation of SC circuits therefore need to incorporate both types of non-ideality, which require methods similar to that of SPICE. For faster simulation the requirement for either finite resistance / bandwidth effects or non-linear effects must be compromised - neither of which are required if the purpose of the simulation is to prove that the implemented topology is correct. Thus in practice idealised SC simulation is both a useful and an accurate simulation method.

5.2.2 Formulation of Circuit Equations

The SC simulation method is based on the two-graph formulation of the circuit constitutive equations [12]. However restrictions on repetitive controlling clocks

have been eliminated. As a result of not requiring the same equation formulation to be used for AC analysis, some additional steps can be taken to decompose the circuit matrices into several much smaller matrices which are significantly quicker to solve. Instead of solving one large, although sparse matrix, it is possible to solve several much smaller matrices typically of maximum order 2 or 3, the solution of which may be explicitly solved without invoking general sparse matrix solution methods.

The circuit is represented by a network consisting only of ideal elements - switches, capacitances, opamps, voltage-controlled-voltage-sources (VCVSs) and independent voltage sources. Furthermore the presence of a switch with its associated controlling clock is taken to imply the possible existence of two related networks - that obtained when the switch is open and that obtained when the switch is closed. Different sets of equations are formulated and solved for each different set of clock states. Thus the switches and clock states do not enter the circuit equations which must be solved for a particular clock phase.

The components are ideal. The switches are ideal in that when a switch is closed, the two nodes are effectively the one circuit point (zero impedance) and when the switch is open the two nodes are unconnected (infinite impedance). The VCVS has zero output impedance and constant gain over all frequencies. The opamp is a type of VCVS which has infinite gain, which ensures that the amplifiers differential input voltage is identically zero.

The equations are set up using the charge conservation law. If a node is connected to capacitances only, then the charge on the node must remain constant. Nodes connected to independent or dependent voltages sources do not have charge conservation equations written for them since the source can allow arbitrary charge transfer. However for each voltage source a voltage equation must be written.

Rather than constructing circuit matrices as done by most simulation programs, the method which has been adopted is to store the equations as linked lists. This is effectively a sparse matrix method where the rows and columns are addressed

associatively (as in an associative array) rather than numerically. This method avoids a lot of the book keeping required when eliminating and reordering equations.

5.2.3 Minimisation of the SC equations

The order of the equation set can be significantly reduced by removing duplicate variables from the set. When a closed switch connects two nodes, the two nodes are effectively one. Both the charges on the nodes and the voltages on the nodes are equivalent. In some cases only the charges of two nodes are equivalent (e.g. the two nodes of a voltage source), or only the two voltages on the nodes are equivalent (e.g. the input terminals of an ideal opamp). This leads to the concepts of charge equivalent and voltage equivalent nodes.

5.3 CONTRIBUTIONS TO THE CIRCUIT EQUATIONS

A different set of equations is created for each distinct set of clock phases which occur. Each of these sets of equations is essentially a DC solution giving the final node voltages after all charge has settled. The passage of time is represented by a sequence of different clock phases. For the case of a simple two phase switch-capacitor filter, there are two clock phases, even and odd which alternate. Sigma-delta circuits add at least two more clock phases for case of the comparator output high and low.

Each component in the circuit makes some contribution to the circuit equations. Rather than constructing the equations on a per node basis, it is simpler to build the equations up by considering each component in turn. The complete set of equations is the superposition of all the individual equations as described in the following sections. The notation j_v is used to denote the voltage at node j while the charge present is denoted by j_q.

5.3.1 Closed Switch

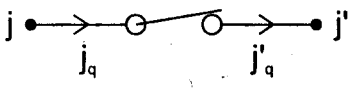


Figure 5-2. Closed Switch

A closed switch connects two nodes to behave as one. Thus:

$$j_v \equiv j'_v \tag{5-30}$$

$$j_q \equiv j'_q \tag{5-31}$$

The equations (5-30) and (5-31) are only valid for clock phases in which the switch is closed. For all other clock phases, the two nodes are disconnected and must be treated as two separate nodes.

5.3.2 Independent Voltage Source

Figure 5-3. Independent voltage source.

The charge on each side of a voltage source (VS) must be equal, allowing two charge equations to merge. The nodes are charge equivalent (5-32). However a voltage equation must be added to represent the voltage constraint (5-33).

$$j_q \equiv j'_q \tag{5-32}$$

$$j_v - j'_v = E \tag{5-33}$$

5.3.3 Voltage controlled voltage source

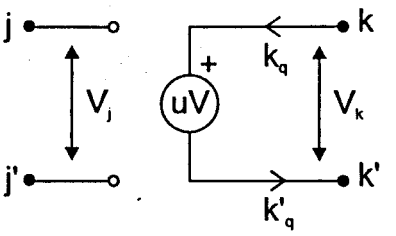


Figure 5-4 Voltage controlled voltage source (VCVS)

The voltage controlled voltage source is similar to a VS except that the voltage constraint is a linear relationship between two other circuit voltages. Thus the equations become:

$$uj_v - uj'_v - k_v + k'_v = 0 \tag{5-34}$$

$$k_q \equiv k'_q \tag{5-35}$$

5.3.4 Operational Amplifier

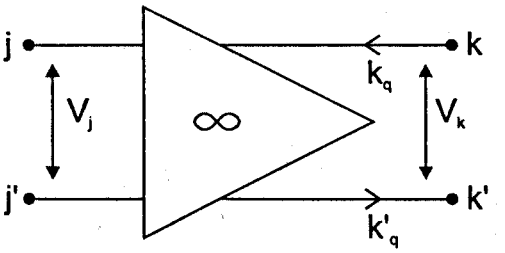


Figure 5-5. Operational Amplifier (Opamp)

The opamp is similar to the VCVS except that it has infinite gain. This cannot enter the circuit equations directly by setting $u = \infty$. Instead the consequence that the input

terminals must be at equal voltages is used (5-36). This allows a further equation to be eliminated.

$$j_v \equiv j'_v \tag{5-36}$$

$$k_q \equiv k'_q \tag{5-37}$$

5.3.5 Buffer

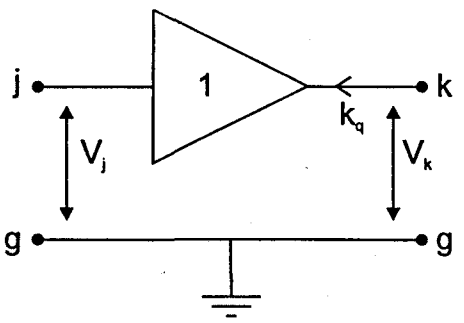


Figure 5-6 Buffer

A buffer is a voltage source which follows the voltage on the input. Thus

$$j_v \equiv k_v \tag{5-38}$$

$$k_q \equiv g_q \tag{5-39}$$

where g_q represents the charge associated with the ground node. Since this is the reference node, no equations are set up for it and it is assumed to have a voltage of zero. Thus node j is made voltage equivalent to node k (5-38) and no charge equations are set up for node k (5-39).

5.3.6 Capacitance

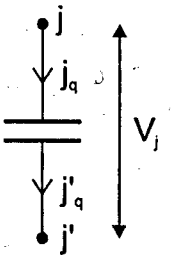


Figure 5-7 Capacitance

Capacitances influence the charge equations for the two nodes connected:

Charges at node j :

$$Cj_v - Cj'_v = Cj_v(t^-) - Cj'_v(t^-) \tag{5-40}$$

Charges at node j' :

$$-Cj_v + Cj'_v = -Cj_v(t^-) + Cj'_v(t^-) \tag{5-41}$$

where $j_v(t^-)$ refers to the voltage on node j at the instance immediately prior to this switch phasing. Capacitors are the only memory in the system from phase to phase. The complete charge equations are given by the superposition of all the contributing components.

5.4 COMPUTER IMPLEMENTATION

The equation set is constructed from the following rules.

Equations are created in the form:

$$LHS = RHS \tag{5-42}$$

where the *RHS* is known.

A charge conservation equation is created for each node other than the ground (reference) node. A single equation is produced for each set of nodes declared as charge equivalent.

Terms are added to the charge equations for each of the capacitors in the circuit. Charge equivalence's are used for both the *LHS* and *RHS*. Voltage equivalence's are used only for the *LHS*. The *RHS* represents the voltages on the capacitances at the end of the previous clock phase which are used to determine the initial charge (before charge transfer) on each node.

Further equations are added for the voltage relationships - independent voltage sources and VCVS. VCVS have no contribution to the RHS.

Unless there is a topological error in the circuit the resulting set of equations should be solvable (n equations and n unknowns). The equations can be written as a matrix equation:

$$A.x = RHS \tag{5-43}$$

and solved by standard LU decomposition techniques. Since the matrix $\boldsymbol{A}$ is sparse, the equation set could be mapped to a standard sparse equation solver [13]. A simple sparse solver has been implemented which has the advantage of retaining the equations in the same data structure as used above.

The solution of the equations yields the voltages for each of the voltage equivalent groups. The only thing left to do is to copy the voltages determined to the other nodes which are voltage equivalent and had been removed from the equation set.

Note that if the equations are stored between clock phases, only the solution of the equation and copying back the voltages to the collapsed voltage nodes need be performed if this clock phasing recurs.

5.4.1 Reduction of the equation set.

The simplification of SC circuit equations relies on the fact that in a particular phase, the SC circuit usually consists of a number of small independent circuits which can be solved independently. The algorithm first looks for single equations which can be immediately solved before looking for pairs (and so on) of equations which can be solved Once nodal voltages become known, references to those voltages in the LHS of the remaining equations may be moved to the RHS (with a corresponding change of sign). Although very simple, this method has been found to very effective. For most differential SC circuits the maximum order of an individual system of equations is three or less. Note that this reduction means that simulation time becomes linear in terms of circuit size. This is an important result since the SC simulation is intended to be able to simulate large systems, such as the entire transmit and receive signal paths of modern telecommunications ICs.

5.5 EXAMPLES

The SC simulation program has been used extensively on a variety of SC circuits, including multirate filters, sigma-delta modulators and other non-linear SC circuits [14].

As an example of the usefulness of SC time domain simulation and as an illustration of how the SC circuit equations are formed and reduced, the single opamp second order sigma delta modulator used in the compander will be discussed. This is only part of the SC circuitry of the compander, but is sufficiently complex in itself that proof of correctness simulation is invaluable. The schematic for this circuit is given in Figure 5-8. The function and operation of this circuit is not obvious from the

diagram, yet to verify correctness requires many thousands of simulation cycles to verify that it is indeed a second order modulator. For example if the clock phasing of the comparator is swapped, the comparator samples the output of the first integrator rather than the second and the result is a first order modulator. This difference would be difficult to see from the few cycles that can be realistically simulated using circuit simulators such as SPICE.

The schematic of Figure 5-8 is used to automatically generate a SPICE compatible netlist, and this is processed to convert the MOS transistor level circuit description in to an SC level netlist. This netlisting process is completely automated and results in a flattened (no hierarchy) netlist using SC level models for the switches and amplifiers (Table 1 and Table 2). Most of the node labels have been marked on the schematic. For this example the amplifier was replaced by an ideal opamp with ideal common-mode performance. The comparator is deleted from the netlist and a small C program is used to set the switch states for each clock phase. The C program monitors the amplifier voltages AMPP and AMPN to determine the comparator digital output and sets the SDQ clock appropriately. For the full compressor simulation, the C program calls a function representing the digital circuits and sets the one-bit multiplier clocks appropriately. Having set all the clocks to their correct states, the C program calls the SC simulation routine which determines the appropriate set of equations to use and solves them to determine the new voltages in the circuit.

A VCVS is used to provide ideal common mode feedback for the ideal opamp. This ensures that the two output voltages swing symmetrically around the analogue ground reference point (AGND). The other VCVS is used as an ideal single-ended to differential converter for the input signal. The voltages at AGND and SWGND are constant and set to be 1.5V (mid-rail on a 3V supply). This sigma-delta modulator derives its feedback signal as the potential between SWGND and VREFN. The capacitance values are in units. So long as all the capacitances are measured in units then only the capacitor ratios are important.

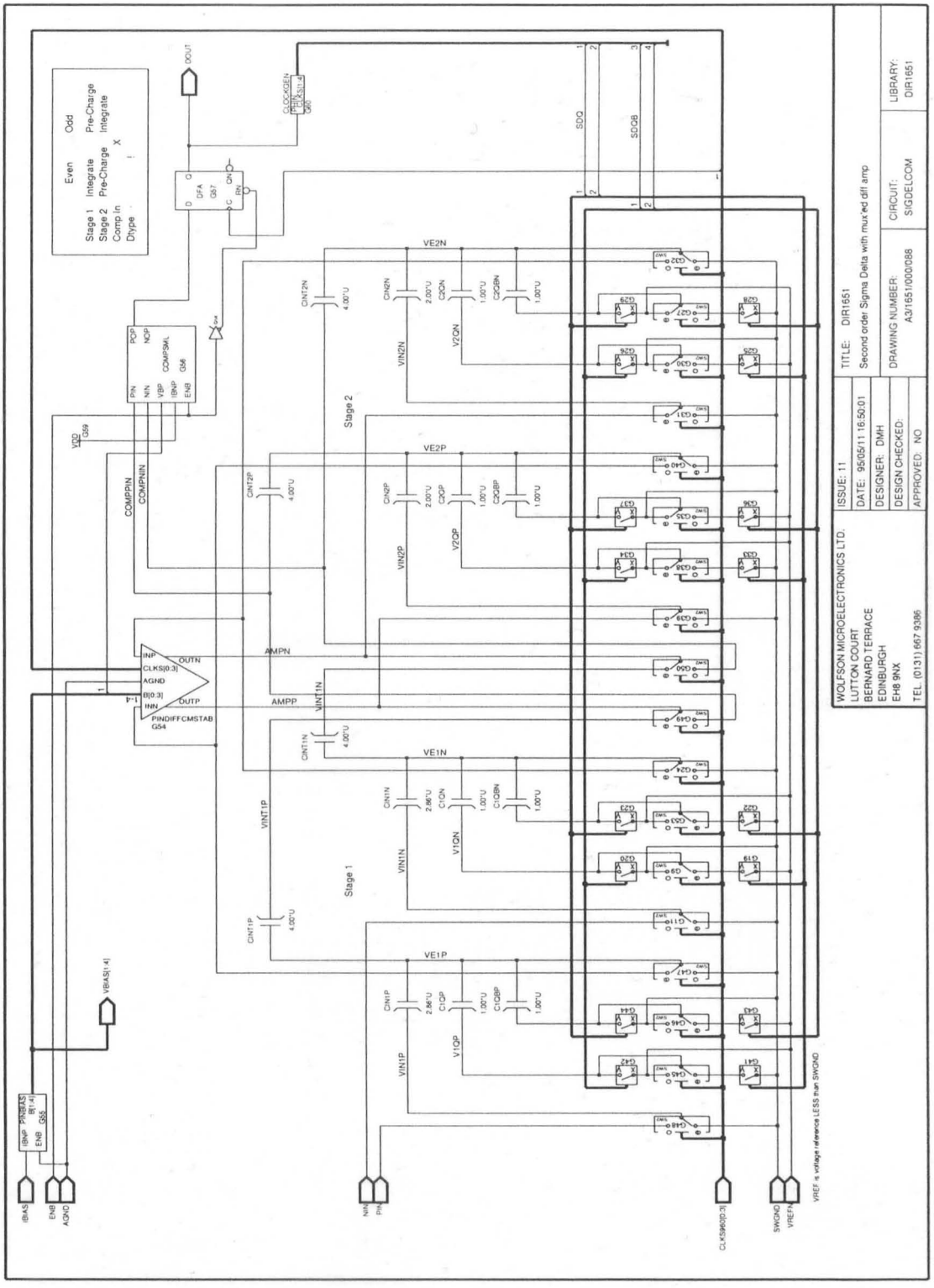


Figure 5-8. Schematic of second order sigma-delta modulator using a single opamp.

```
Nodes:

      0               AGND            PIN             8
      2               AMPNIN          AMPP            10
      4               AMPN            11              5
      AMPPIN          9               VREFN           NIN
      SWGND           3               COMPNIN         COMPPIN
      VINT1P          VINT1N          VIN2N           VIN2P
      VIN1P           VIN1N           V2QBP           V2QBN
      V2QN            VE2N            V2QP            VE2P
      V1QBP           V1QBN           V1QN            VE1N
      V1QP            VE1P

Capacitors:

      c(cint2n)       VE2N        COMPNIN     4
      c(cint2p)       VE2P        COMPPIN     4
      c(cint1p)       VE1P        VINT1P      4
      c(cint1n)       VE1N        VINT1N      4
      c(cin2n)        VE2N        VIN2N       2
      c(cin2p)        VE2P        VIN2P       2
      c(cin1p)        VE1P        VIN1P       2.86
      c(cin1n)        VE1N        VIN1N       2.86
      c(c2qbp)        VE2P        V2QBP       1
      c(c2qbn)        VE2N        V2QBN       1
      c(c2qn)         VE2N        V2QN        1
      c(c2qp)         VE2P        V2QP        1
      c(c1qbp)        VE1P        V1QBP       1
      c(c1qbn)        VE1N        V1QBN       1
      c(c1qn)         VE1N        V1QN        1
      c(c1qp)         VE1P        V1QP        1

Opamps:

      opamp(xg54.opamp) AMPPIN AMPNIN AMPP AMPN

VCVSs:

      vcvs(vcvs) PIN AGND NIN AGND -1
      vcvs(xg54.vcvs) AMPP AGND AMPN AGND -1

Clocks:

      SDQB  SDQ   CLKS9602   CLKS9600

Sources:

      vs(vsSWGND) SWGND 0
      vs(vsAGND) AGND 0
      vs(vsVREFN) VREFN 0
      vs(vsPIN) PIN 0
```

Table 1 Processed Netlist (capacitors, amplifiers, voltage sources).

Switches:

s(xg53.s_odd)	VREFN	V1QBN	CLKS9602
s(xg53.s_even)	9	V1QBN	CLKS9600
s(xg50.s_odd)	COMPNIN	AMPN	CLKS9602
s(xg50.s_even)	VINT1N	AMPN	CLKS9600
s(xg49.s_odd)	COMPPIN	AMPP	CLKS9602
s(xg49.s_even)	VINT1P	AMPP	CLKS9600
s(xg48.s_odd)	PIN	VIN1P	CLKS9602
s(xg48.s_even)	SWGND	VIN1P	CLKS9600
s(xg47.s_odd)	SWGND	VE1P	CLKS9602
s(xg47.s_even)	AMPNIN	VE1P	CLKS9600
s(xg46.s_odd)	SWGND	V1QBP	CLKS9602
s(xg46.s_even)	8	V1QBP	CLKS9600
s(xg45.s_odd)	VREFN	V1QP	CLKS9602
s(xg45.s_even)	2	V1QP	CLKS9600
s(xg44.s)	SWGND	V1QBP	SDQ
s(xg43.s)	VREFN	8	SDQB
s(xg42.s)	VREFN	V1QP	SDQB
s(xg41.s)	SWGND	2	SDQ
s(xg40.s_odd)	AMPNIN	VE2P	CLKS9602
s(xg40.s_even)	SWGND	VE2P	CLKS9600
s(xg39.s_odd)	SWGND	VIN2P	CLKS9602
s(xg39.s_even)	AMPP	VIN2P	CLKS9600
s(xg38.s_odd)	4	V2QP	CLKS9602
s(xg38.s_even)	VREFN	V2QP	CLKS9600
s(xg37.s)	SWGND	V2QBP	SDQ
s(xg36.s)	VREFN	10	SDQB
s(xg35.s_odd)	10	V2QBP	CLKS9602
s(xg35.s_even)	SWGND	V2QBP	CLKS9600
s(xg34.s)	VREFN	V2QP	SDQB
s(xg33.s)	SWGND	4	SDQ
s(xg32.s_odd)	AMPPIN	VE2N	CLKS9602
s(xg32.s_even)	SWGND	VE2N	CLKS9600
s(xg31.s_odd)	SWGND	VIN2N	CLKS9602
s(xg31.s_even)	AMPN	VIN2N	CLKS9600
s(xg30.s_odd)	5	V2QN	CLKS9602
s(xg30.s_even)	SWGND	V2QN	CLKS9600
s(xg29.s)	VREFN	V2QBN	SDQ
s(xg28.s)	SWGND	11	SDQB
s(xg27.s_odd)	11	V2QBN	CLKS9602
s(xg27.s_even)	VREFN	V2QBN	CLKS9600
s(xg26.s)	SWGND	V2QN	SDQB
s(xg25.s)	VREFN	5	SDQ
s(xg24.s_odd)	SWGND	VE1N	CLKS9602
s(xg24.s_even)	AMPPIN	VE1N	CLKS9600
s(xg23.s)	VREFN	V1QBN	SDQ
s(xg22.s)	SWGND	9	SDQB
s(xg20.s)	SWGND	V1QN	SDQB
s(xg19.s)	VREFN	3	SDQ
s(xg11.s_odd)	NIN	VIN1N	CLKS9602
s(xg11.s_even)	SWGND	VIN1N	CLKS9600
s(xg9.s_odd)	SWGND	V1QN	CLKS9602
s(xg9.s_even)	3	V1QN	CLKS9600

Table 2 Processed Netlist (Switches).

The topology in which these components are connected depends on the switch control signals. Each distinct switch control signal signifies a further two possible topologies that are possible. Thus the number of topologies increases exponentially with the number distinct switch control signals. Fortunately many of these combinations are never occur (e.g. EVEN and ODD both on) and in practice it is possible to store all the equations associated with each topology. This makes the program very much more efficient since the equation formulation needs to be performed only once for each topology. If computer storage is a problem then some performance enhancement may be still obtained by keeping a cache of recently used formulations.

In this example there are four switch control lines - two associated with the SC integrator (CLKS9600 and CLKS9602) and two associated with the one bit DAC (SDQ and SDQB).

For each phase, as it occurs, a set of equations if formed. The case of CLKS9600 (even) and SDQB active will be discussed here. The switches listed in Table 3 are closed.

s(xg53.s_even)	s(xg50.s_even)	s(xg49.s_even)
s(xg48.s_even)	s(xg47.s_even)	s(xg46.s_even)
s(xg45.s_even)	s(xg44.s)	s(xg41.s)
s(xg40.s_even)	s(xg39.s_even)	s(xg38.s_even)
s(xg37.s)	s(xg35.s_even)	s(xg33.s)
s(xg32.s_even)	s(xg31.s_even)	s(xg30.s_even)
s(xg29.s)	s(xg27.s_even)	s(xg25.s)
s(xg24.s_even)	s(xg23.s)	s(xg19.s)
s(xg11.s_even)	s(xg9.s_even)	

Table 3 List of closed switches during EVEN and SQB active.

The first task is to determine the charge and voltage equivalence's. These are listed in Table 4.

Nodes:	Charge Equivalent	Voltage Equivalent
0		
AGND	Q == 0	
PIN	Q == 0	
8	Q == 0	V == SWGND
2	Q == 0	V == SWGND
AMPNIN		V == AMPPIN
AMPP	Q == 0	V == VINT1P
10		
4	Q == 0	V == SWGND
AMPN	Q == 0	V == VINT1N
11		
5	Q == 0	V == VREFN
AMPPIN		
9	Q == 0	V == VREFN
VREFN	Q == 0	
NIN	Q == 0	
SWGND	Q == 0	
3	Q == 0	V == VREFN
COMPNIN		
COMPPIN		
VINT1P	Q == 0	
VINT1N	Q == 0	
VIN2N	Q == 0	V == VINT1N
VIN2P	Q == 0	V == VINT1P
VIN1P	Q == 0	V == SWGND
VIN1N	Q == 0	V == SWGND
V2QBP	Q == 0	V == SWGND
V2QBN	Q == 0	V == VREFN
V2QN	Q == 0	V == SWGND
VE2N	Q == 0	V == SWGND
V2QP	Q == 0	V == VREFN
VE2P	Q == 0	V == SWGND
V1QBP	Q == 0	V == SWGND
V1QBN	Q == 0	V == VREFN
V1QN	Q == 0	V == VREFN
VE1N	Q == AMPPIN	V == AMPPIN
V1QP	Q == 0	V == SWGND
VE1P	Q == AMPNIN	V == AMPPIN

Table 4 Charge and voltage equivalent nodes

Typically more than two nodes can be in each charge or voltage equivalent group. Rather than store each group as a set, a pointer is used to point to a single member of the set which is then used in equations. These pointers are indicated by the "Q==" and "V==" entries in Table 4. Thus charge equations are only created for the nodes

in Table 4 which do not have a charge equivalent node marked. Likewise only the voltages on the nodes which do not have a voltage equivalent node marked need to be solved. The other voltages will be copied from the solution vector as indicated by the voltage equivalence field. Thus the charge and voltage equivalence entries are a simple and effective means of reducing equation complexity.

Note that in the above table the opamp inputs (AMPNIN and AMPPIN) are defined to have no voltage between then by means of a voltage equivalence.

At this stage, all the equations required have been identified, but are initially empty (they have no terms in either the LHS or RHS). The next step fills the equations by considering each component in turn, together with the voltage and charge equivalence table and superimposing the component equation contributions as detailed in section 5.3 onto the system equations. The resulting set of equations are printed by the program are listed in Table 5.

Equation Name	LHS	RHS
vcvs	1*v(VINT1N)	0
vcvs(xg54.vcvs)	+ 2*v(AGND)	
	+ 1*v(VINT1P)	
vcvs	1*v(NIN)	0
vcvs(vcvs)	+2*v(AGND)	
	+1*v(PIN)	
Source	1*v(PIN)	1*v(source_node)
vs(vsPIN)		
Source	1*v(VREFN)	1*v(source_node)
vs(vsVREFN)		
Source	1*v(AGND)	1*v(source_node)
vs(vsAGND)		
Source	1*v(SWGND)	1*v(source_node)
vs(vsSWGND)		
Chgs @ COMPPIN	4*v(COMPPIN)	4*v(COMPPIN)
	-4*v(SWGND)	-4*v(VE2P)
Chgs @ COMPNIN	4*v(COMPNIN)	4*v(COMPNIN)
	-4*v(SWGND)	-4*v(VE2N)
Chgs @ AMPPIN	-2*v(VREFN)	-1*v(V1QN)
	-2.86*v(SWGND)	-1*v(V1QBN)
	-4*v(VINT1N)	-2.86*v(VIN1N)
	+8.86*v(AMPPIN)	-4*v(VINT1N)
		+8.86*v(VE1N)
Chgs @ 11	0	0
Chgs @ 10	0	0
Chgs @ AMPNIN	-4.86*v(SWGND)	-1*v(V1QP)
	-4*v(VINT1P)	-1*v(V1QBP)
	+8.86*v(AMPPIN)	-2.86*v(VIN1P)
		-4*v(VINT1P)
		+8.86*v(VE1P)

Table 5 Resulting equation set.

These equations are then simplified as much as possible. For example the voltages at the input nodes such as PIN, NIN, VREFP, VREFN, ANGND are all known without having to solve simultaneous equations. Knowing these nodes then simplifies the equations for remaining nodes. In this case all but three of the nodal voltages may be determined by a linear combination of already known nodal voltages. The final three nodal voltages must be determined simultaneously from the set of three remaining equations. The final complete sequence of equations sets are printed out by the program and are listed in Table 6.

```
Order 1:
Source vs(vsPIN):
 1*v(PIN)                          1*v(source_node)
                                    - (  0 )


Order 1:
Source vs(vsVREFN):
 1*v(VREFN)                        1*v(source_node)
                                    - (  0 )


Order 1:
Source vs(vsAGND):
 1*v(AGND)                         1*v(source_node)
                                    - (  0 )


Order 1:
vcvs vcvs(vcvs):
 1*v(NIN)                          0
                                    - ( 1*v(PIN)  -2*v(AGND)  )


Order 1:
Source vs(vsSWGND):
 1*v(SWGND)                        1*v(source_node)
                                    - (  0 )


Order 1:
Chgs @ COMPPIN:
 4*v(COMPPIN)                      4*v(COMPPIN)  -4*v(VE2P)
                                    - (  -4*v(SWGND)  )


Order 1:
Chgs @ COMPNIN:
 4*v(COMPNIN)                      4*v(COMPNIN)  -4*v(VE2N)
                                    - (  -4*v(SWGND)  )


Order 3:
vcvs vcvs(xg54.vcvs):
 1*v(VINT1N)  1*v(VINT1P)          0
                                    - (  -2*v(AGND)  )
Chgs @ AMPPIN:
 -4*v(VINT1N)  8.86*v(AMPPIN)      -1*v(V1QN)  -1*v(V1QBN)
                                   -2.86*v(VIN1N)  -4*v(VINT1N)
                                   +8.86*v(VE1N)
                                    - ( -2.86*v(SWGND)
                                              -2*v(VREFN)  )
Chgs @ AMPNIN:
 -4*v(VINT1P)                      -1*v(V1QP)      -1*v(V1QBP)
+8.86*v(AMPPIN)                    -2.86*v(VIN1P) -4*v(VINT1P)
                                   +8.86*v(VE1P)
                                    - (  -4.86*v(SWGND)  )
```

Table 6 Simplified sequence of equations sets for the sigma-delta modulator.

In this example, only one set of equations needs to be solved; this being a set of three equations governing the voltages on the integrating capacitors and a voltage controlled voltage source providing common mode feedback for the amplifier. The equations above are solved in sequence. The variables on the RHS in plain text refer to the node voltages prior to the sampling instant. The variables of the RHS in *italics* refer to node voltages after the sampling instant - these will have already been determined by the time they are used. This example has shown how a system consisting of 37 nodes, 52 switches and 16 capacitors has been reduced to some first order equations and a single third order system of equations.

5.5.1 Finite Bandwidth and Resistance Effects

It is possible to use ideal SC simulation to model finite bandwidth and switch on resistance. The method involves converting resistances to switched capacitors operating at a rate higher than the highest clock in the system [3]. Whilst crude, this method has the advantages of being quick to implement (no additional simulation capability is required) and fairly quick to run (there is no requirement for the numerical integration of differential equations).

This method was used to study the settling in compander. The operational transconductance amplifiers (OTA) are modelled as a VCVS with a resistive outputs implemented using the SC method. From the schematic (Figure 5-8) it can be seen that the OTA in the sigma-delta ADC drives a larger capacitive load during the EVEN phase giving a longer settling behaviour which can be seen in Figure 5-8 showing the OTA outputs and inputs. This method has been used to confirm the ordering requirements on SC clocks in the compander system since the precise clock timing can be quite critical. For example, the input of the compander may sample the preceding stage before it has finished settling. The signal dependent switching on the input of the compander can disrupt the operation of SC circuits ahead of the compander giving rise to aliasing and tone phenomena.

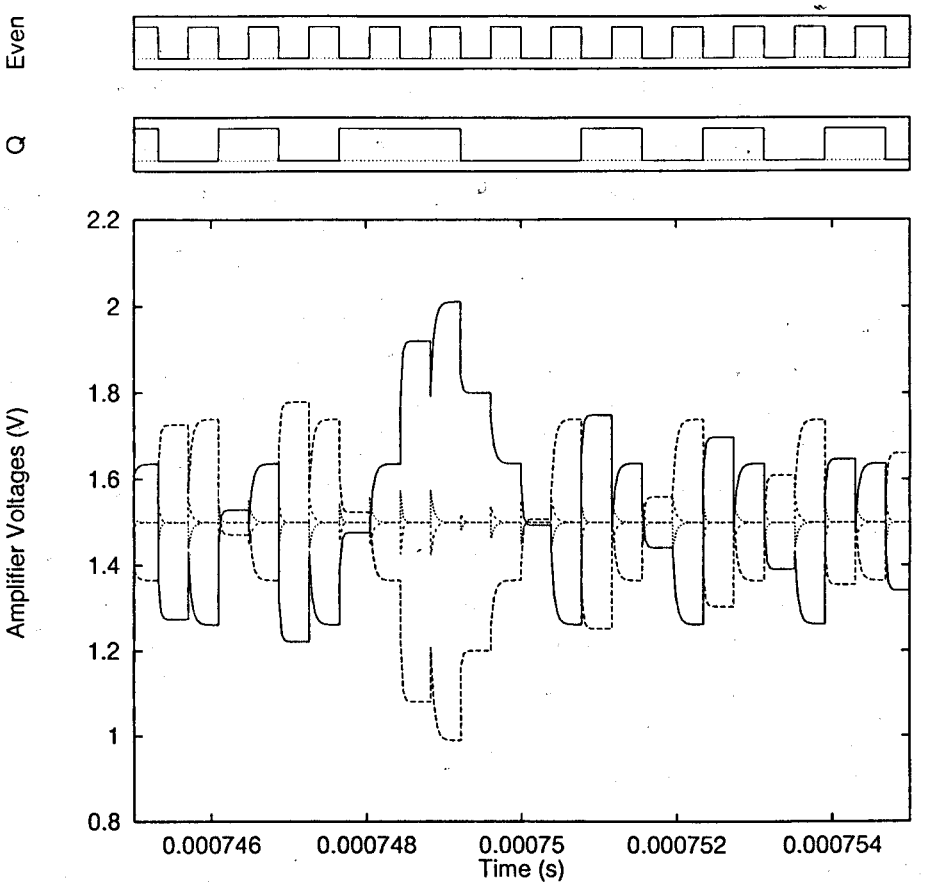


Figure 5-9. Simulation of SC circuit with finite output impedance OTA.

5.5.2 Third Order Sigma-Delta Modulator

The SC simulation program has also been used during the design of a third order cascade sigma-delta modulator. The cascade type of sigma-delta modulator uses a second sigma-delta modulator to digitise the analogue quantisation error of the first. When the two sigma-delta modulator outputs are suitably combined, the result is a higher order noise shaping characteristic. Since this arrangement relies on noise cancellation, component sensitivity and tolerance are important considerations. Also the method of combining the two outputs must be correct and this is very difficult to confirm using a traditional simulator since many samples must be simulated before the performance improvement can be measured. A further use of the simulator for this design problem was to check signal swings for both small and large inputs.

The circuit designed and simulated is shown in Figure 5-10. The first stage has been implemented using a switched-capacitor gain enhancement technique which is used here to increase the effective gain of the first OTA [15]. This reduces the distortion of this stage [16]. The circuit has 37 capacitors and 58 switches.

The simulation program was first used to check that the topology operates as a third order modulator. The equations governing the integrator voltages are more complex than for a simple model as a result of the gain enhancement stage. All the integrators integrate charge on both clock phases (even and odd). The SC simulation provides both even and odd phase voltages used for checking voltage swings. The signal swings are quite different for large and small signal conditions. The signals on the three inputs are plotted for the large signal case in Figure 5-11. This clearly shows that the signal on the first stage is correlated with the input. The capacitance values have been chosen that to ensure that these signals swing by less than half the supply voltage (5V). This is required for the best integrator linearity and consequently the best overall converter linearity.

The internal signal swings for small inputs is quite different (Figure 5-13). Probability density functions for both the large signal and small signal cases have been determined and are shown in Figure 5-12 and Figure 5-14.

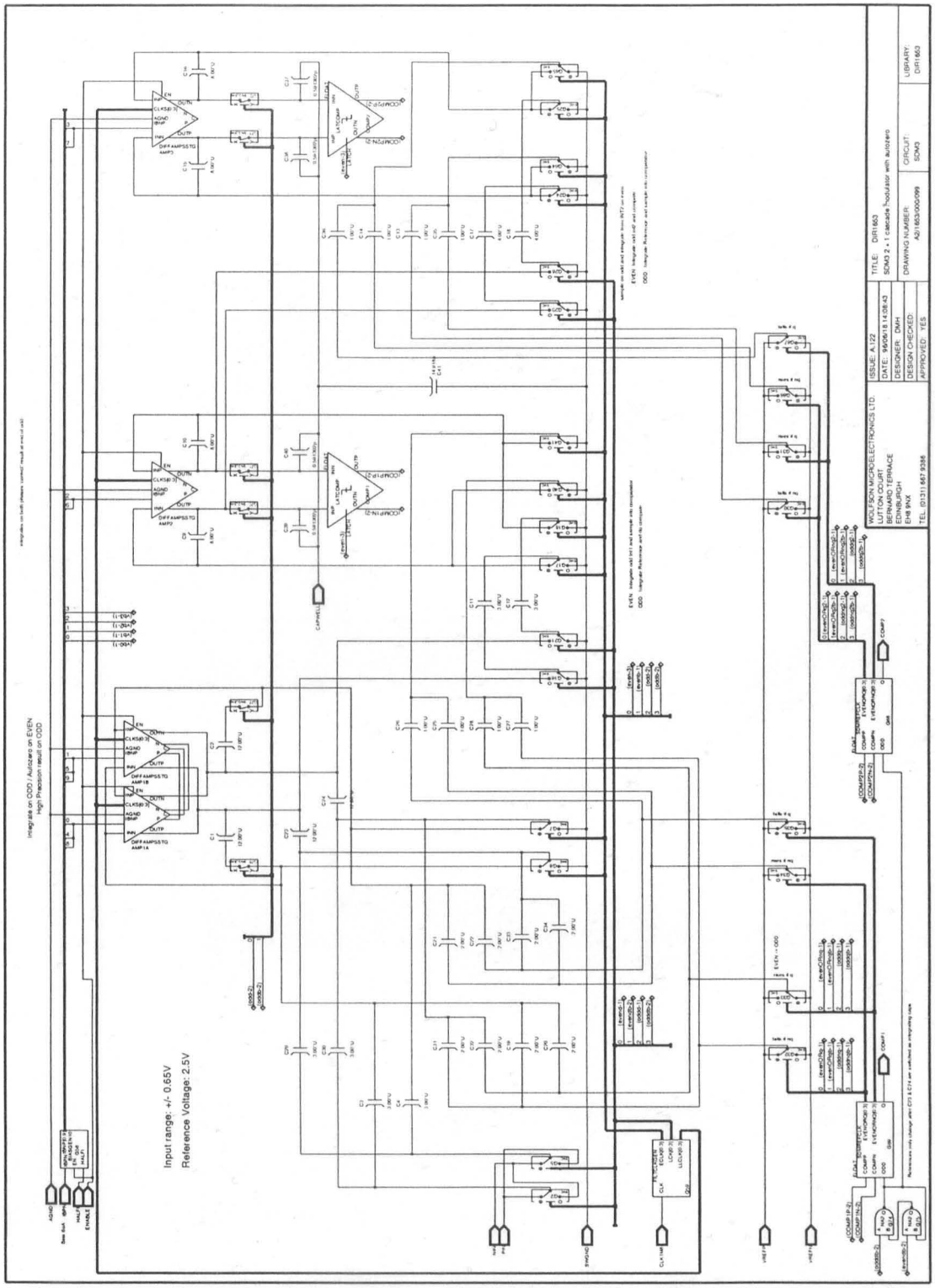


Figure 5-10. Schematic for the third order sigma-delta modulator.

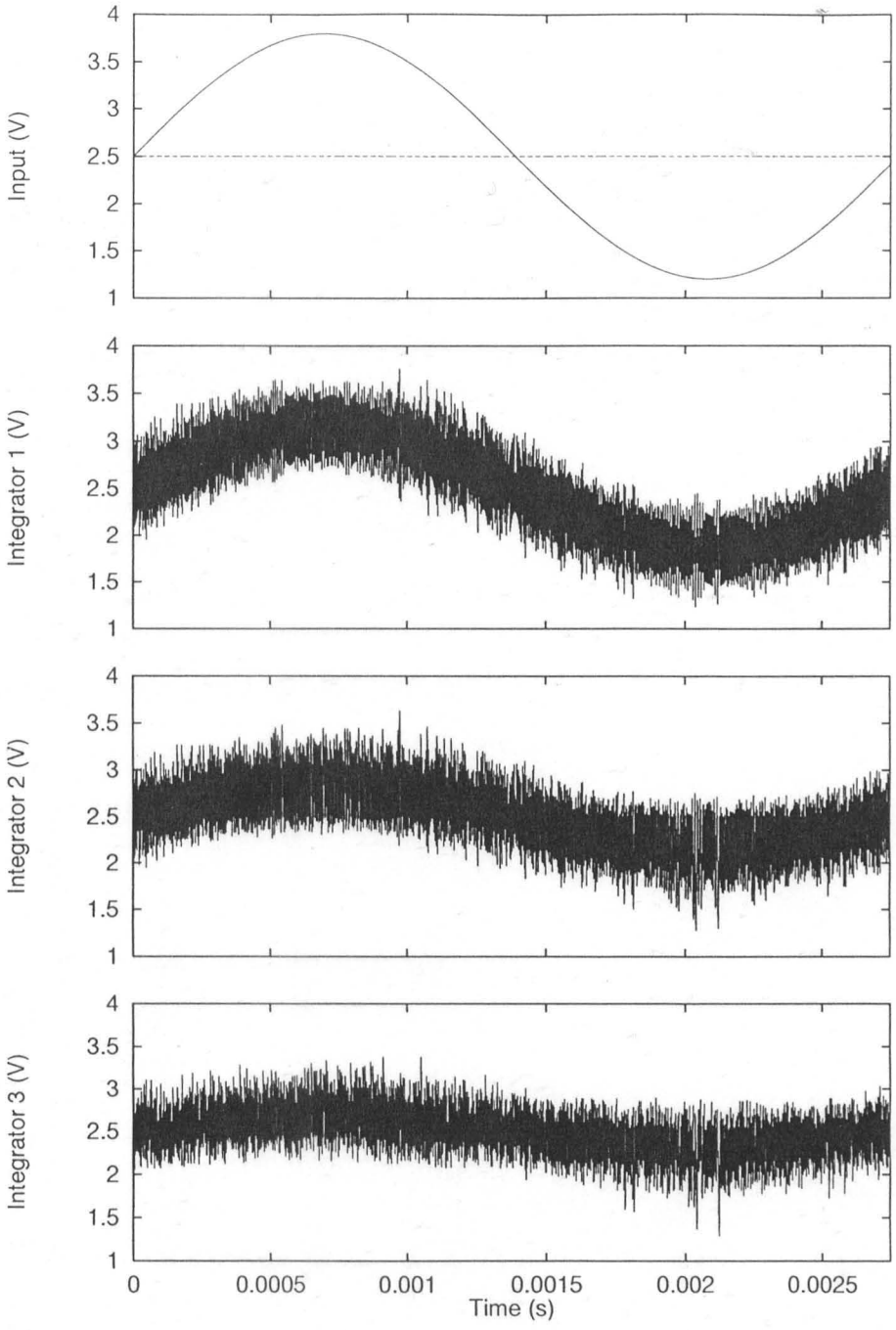


Figure 5-11 Signal swings for large sinusoidal input.

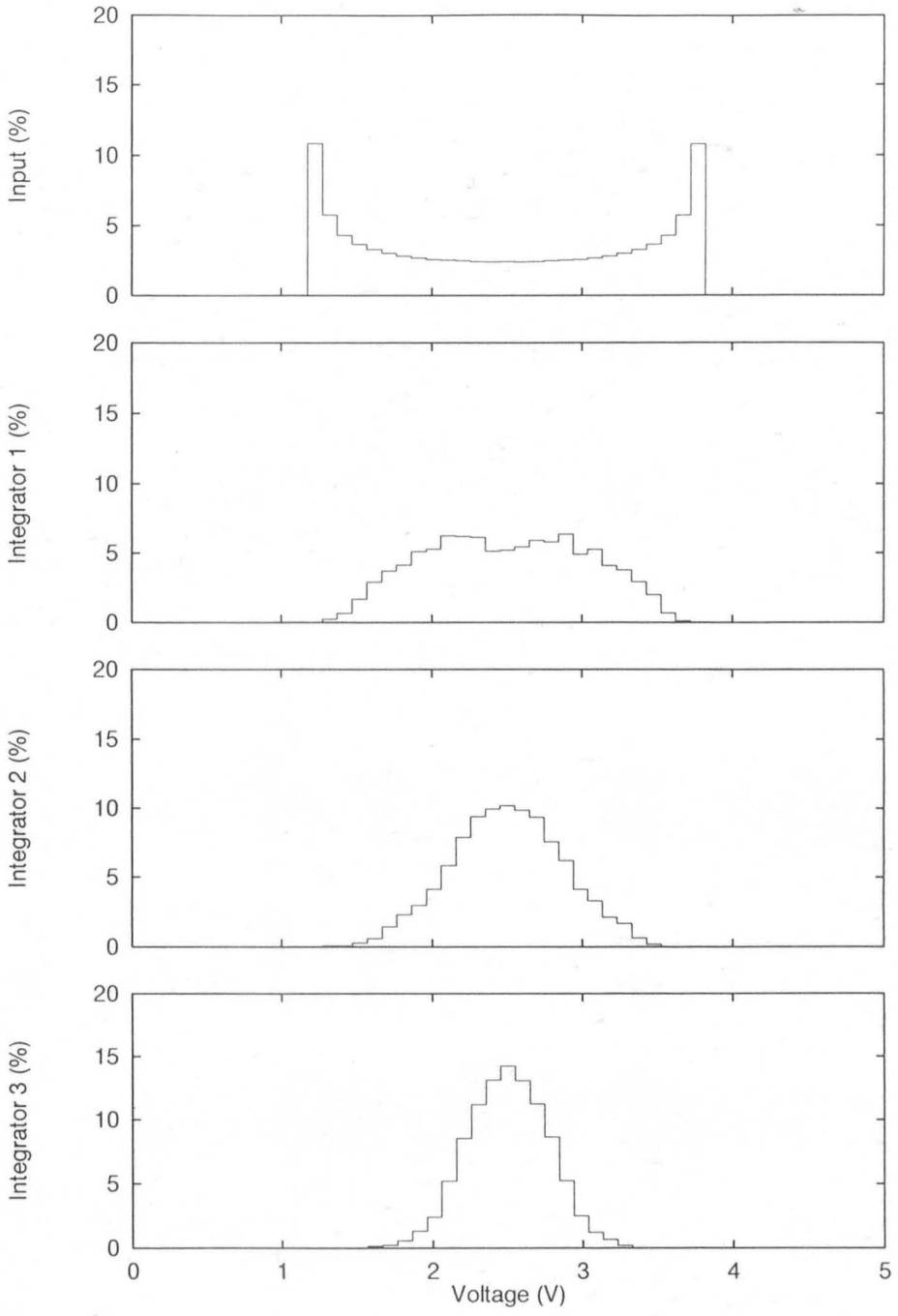


Figure 5-12 Histograms corresponding to Figure 5-11.

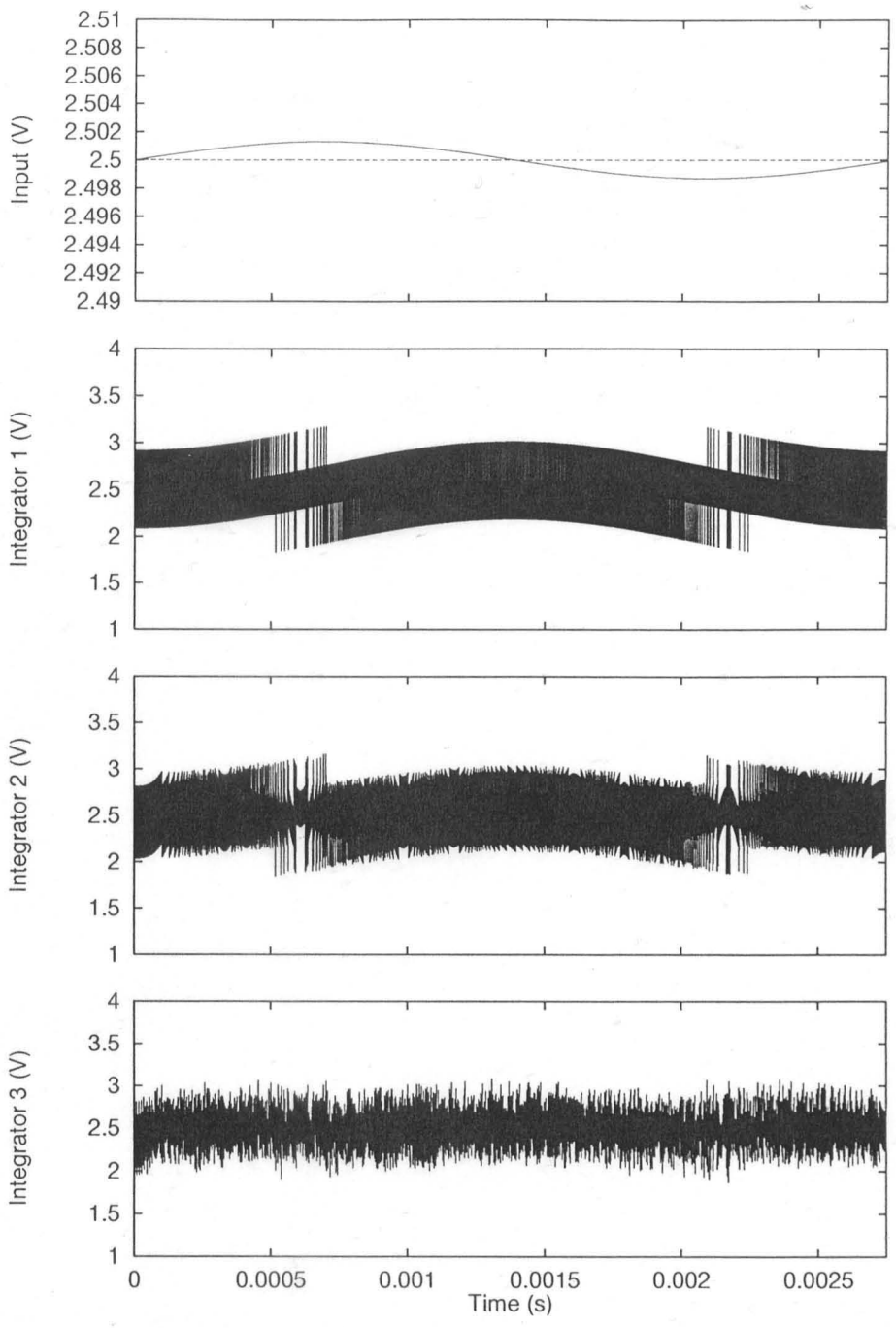


Figure 5-13 Internal signals for small sinusoidal input.

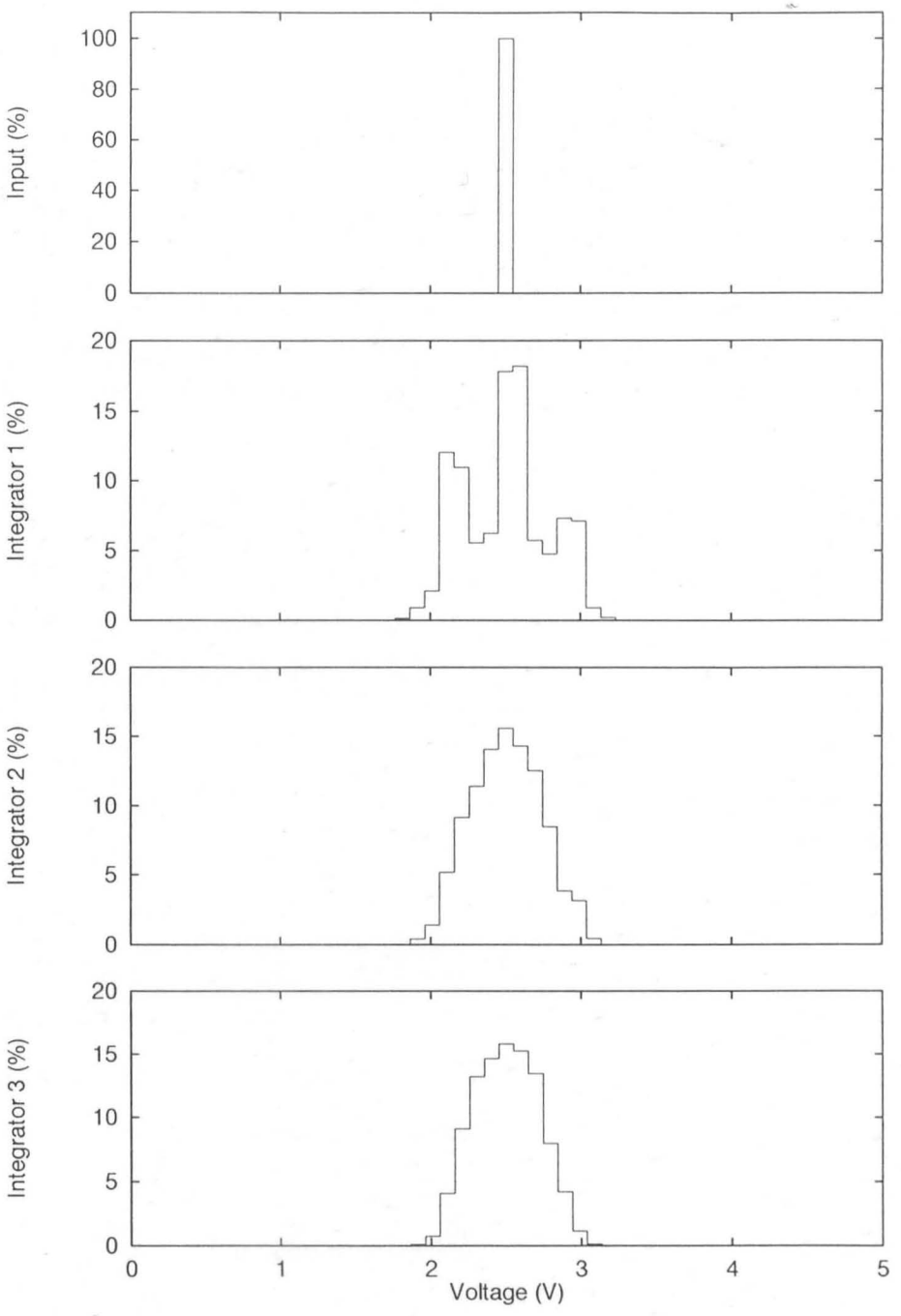


Figure 5-14 Histogram of internal signal swings corresponding to Figure 5-13.

5.6 COMPARISON OF SIMULATION SPEED

The sigma-delta modulator was simulated using commercial simulators. It is difficult to compare simulators because of the speed / accuracy trade-off. The default accuracy settings were used, and only the most important nodal voltages saved since

saving output data can be a bottleneck. The times were measured on a SUN ultraSPARC 1.

Simulator	Speed (samples / second)	Time (seconds / sample)
HSPICE	0.09	11
SABER (transistor level)	0.09	11
SABER (SC level)	1	1
SCITS	800	0.00125
C code	50000	0.00002

Table 7 Speed comparison of various simulation methods for SC circuits.

From Table 7 it can be seen that the ideal SC simulator (SCITS) is four orders of magnitude faster than SPICE level simulation. The version of SPICE used was HSPICE [17]. An order of magnitude speed benefit over HSPICE can be achieved by using the ideal components provided by SABER [18]. For this simulation, the amplifier was replaced by a ideal voltage controlled current source, the MOS switches by ideal digitally controlled switches, the clock generator by a digital equivalent and the comparator was replaced by a very simple ideal model. There were no MOS transistors in this simulation. However this simulation did take switch on resistances and finite transconductances into account, giving non instantaneous charge transfer and so the simulator still had to converge over continuous time signals.

5.7 SIMULATION OF DIGITAL SYSTEMS

Simulation of digital sigma-delta modulator systems is usually much less of a problem than for analogue systems. Digital systems map well to being modelled on digital computers using languages such a C, C++ and VHDL [19]. In particular C++ and VHDL are suitable for emulating digital signal processing operations accurate to

the least significant bit (LSB). Performing fixed-point arithmetic in C, C++ and VHDL is surprisingly tedious, especially if the word-lengths are to be kept variable so that the word-length can be increased until the required noise performance is met. A library of C++ routines has been written which makes coding DSP algorithms very much simpler, whilst still retaining accuracy down to the LSB in fixed word-length arithmetic.

5.7.1 Fixed Point Arithmetic in C++

The fixed point library was written in C++ making use of the facility to define new types and to overload the operators. The performance requirements of digital systems can be determined and checked before the detailed digital design starts. The C++ models can then be compared against a digital logic simulation to check for correct gate level implementation.

Each fixed-point quantity has an associated word length which cannot be altered. To allow for fractional quantities, the number of integer bits (i) and the number of fractional bits (q) are stored and are denoted as (i,q) [20]. Thus the decimal number 1.375 may be held as 1.011 in (1,3) format. Two distinct object types are available to cater for both signed and unsigned numbers. The definition of the fixed-point type and the associated operators were chosen to be as mathematically correct as possible. Thus operators such as "+", "-" and "*" (multiply) return fixed point results with a word length long enough to ensure that the result can be exactly represented. The assignment operator also ensures that no information loss will occur. This differs from the approach of Kim and Sung [21,22] and was chosen to ensure that all information loss was explicit. The "%=" assignment operator was chosen to denote the concept of an approximate or lossy assignment which can be used to assign floating point numbers to a fixed point variable, or to deliberately truncate a longer fixed point quantity and assign to a shorter one. The assignment operators such as "+=" cannot expand the word length of the left hand side, and so operate using modulo arithmetic. These techniques which can be elegantly coded in a standard computer language are likely to supersede the use of special purpose programs such

as Silage [23,24,25]. The following code extract shows how the fixed-point library can be used to describe the operation of the digital filter shown in Figure 5-15.

```
// Filter Coefficients (quantised to 6 bits)
const unsigned_FixPt    c1( 0.470753, 0, 6 ),
                        c2( 0.922338, 0, 6 );

inline void allpass::filter( signed_FixPt x )
{
      signed_FixPt new_internal =
            ((( x - internal ) * c1).truncate_to(15) + input_d)
            .limit( 5, "allpass - 1" );

      output_d = ((( new_internal - output_d ) * c2)
            .truncate_to(15) + internal )
            .limit(5, "allpass - 2");

      internal = new_internal;
      input_d = x;

      set_output( output_d );
      make_valid(1);
      }
```

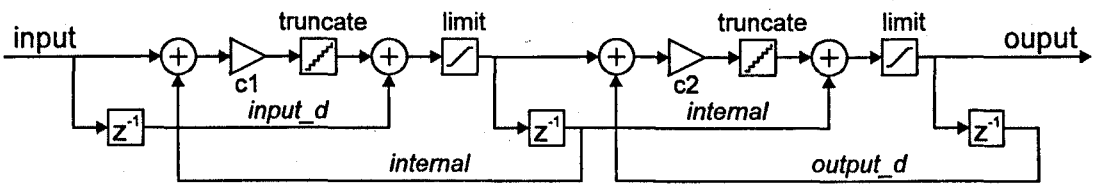


Figure 5-15 Digital All-pass Filter.

The all-pass filter shown in Figure 5-15 can be used to form efficient interpolating and decimating filters [26] which are very suitable for use with sigma-delta modulators. Systems consisting of many of these all-pass sections together with digital comb filters and digital sigma-delta modulators have been successfully modelled with the fixed-point library and verified against digital gate level simulations. The characteristic of the models were shown to be identical to the designed hardware by comparing the responses from various typical and worse case input signals.

The fixed-point library may be combined with the ideal SC simulator to provide complete system level simulation of mixed-signal ICs. The TFN library described in the appendix may also be combined to allow the effects of the continuous time filters and external filters to be included in the simulations.

5.8 SUMMARY

The switched capacitor simulator has been shown to fill a gap in the simulation tools provided to designers, by allowing speedy simulation of SC circuits including internally-controlled switches from a schematic netlist. The simulation has been shown to be four orders of magnitude faster than SPICE and three orders of magnitude faster than a commercially available solution. Furthermore use of the C++ language greatly eases the writing of simulation code to model the digital parts of a system accurately and efficiently.

5.9 REFERENCES

[1] J. Buck, S. Ha, E.A. Lee and D.G. Messerschmitt, "Multirate Signal Processing in Ptolemy," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol 2, New York, NY, pp 1245-1248, 1991.

[2] D.M. Hossack and J.I. Sewell, "A Method for the Evaluation of Multirate Sigma-Delta Systems," *IEEE International Symposium on Circuits and Systems*, San Diego, California, pp. 1328-1331, May 1992.

[3] *User's Guide for SWITCAP Version 5*, Columbia University, New York, NY, August 1987.

[4] K. Suyama, S. Fang and Y. Tsividis, "Simulation of Mixed Switched-Capacitor / Digital Networks with Signal-Driven Switches," *IEEE Journal of Solid-State Circuits*, vol 25, no. 6, pp 1403-1413, December 1990.

[5] T. Chnank and R. Chadha, "Switched-Capacitor Simulation Models for Full-Chips Verification," *IEEE Transactions Computer-Aided Design*, vol. 11, no. 11, pp. 1363-1370, November 1992.

[6] M.A. Copeland, G. Bell and T.A. Kwasniewski, "A Mixed-Mode Sampled-Data Simulation Program," *IEEE Journal of Solid-State Circuits*, vol. SC-22, no. 6, pp. 1098-1105, December 1987.

[7] G. Brauns, M. Steer, S. Ardalan, and J. Paulos, "Table-based modelling of Delta-Sigma Modulators using ZSIM," *IEEE Transactions on CAD for ICs*, vol. 9, no. 2, pp. 142-150, Feburuary1990.

[8] Z.Q. Shang, *Advanced Simulation Methods for Switched Networks*, PhD Thesis, University of Glasgow, Glasgow, March 1995.

[9] B.E. Boser and B.A. Wooley, "The design of Sigma-Delta Modulation Analogue-to Digital Converters," *IEEE Journal of Solid State Circuits*, vol. 23, pp. 1298-1308, December 1988.

[10] R.W. Adams, "Theory and practical implementation of a fifth order sigma-delta A/D converter", *Journal of the Audio Engineering Society*, vol. 39, no.7/8, pp. 1515-528, July/August 1991.

[11] I. Yusim and K. Suyama, "Analysis of Switched-Capacitor and Switched-Current Networks with Complete Settling Assumption," *IEEE International Symposium on Circuits and Systems*, Atlanta, Georgia, vol. IV, pp. 431-434, May 1996.

[12] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand and Reinhold, New York, 1983.

[13] B. N. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole, Pacific Grove, California, 1995.

[14] D.M. Hossack, J.I. Sewell and J.R.C. Reid, "Issues in the Design of Low Oversampling Ration Single Bit Sigma-Delta Modulators," *IEE Colloquium on Oversampling Techniques and Sigma-Delta Modulation*, London, UK, Digest No. 1994/083, March 1994.

[15] L.E. Larson and G.C. Temes, "SC building blocks with reduced sensitivity to finite amplifier gain, bandwidth and offset voltage," *IEEE International Symposium on Circuits and Systems*, Philadelphia, Pennsylvania, pp. 334-338, May 1987.

[16] Y. Huang, G.C. Temes and P.F. Ferguson, "Reduced Non-linear Distortion in Circuits with Correlated Double Sampling," *IEEE International Symposium on Circuits and Systems*, Atanta, Georgia, pp. 159-16, May 1996.

[17] HSPICE analogue circuit simulator, Metasoft Inc., Campbell, California.

[18] SABER simulator, Analogy Inc., Beaverton, Oregon.

[19] V.K. Madisetti, *VLSI Digital Signal Processors - An Introduction to Rapid Prototyping and Design Synthesis*, Butterworth-Heinemann, Newton, Massachusetts, USA 1995.

[20] Jere Babst (editor), *Digital Signal Processing Applications using the APSP-2100 Family*, vol. 1, Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[21] S. Kim and W. Sung, "Fixed-point Simulation Utility for C and C++ Based Digital Signal Processing Programs," *Proceedings of the Twenty-eighth Annual Asilomar Conference on Signals, Systems and Computer*, Pacific Grove, California, pp. 162-166, November 1994.

[22] S. Kim, K. Kum and W. Sung, "Fixed-point Optimization Utility for C and C++ Based Digital Signal Processing Programs," *IEEE Workshop on VLSI Signal Processing*, Osaka, Japan, October 1995.

[23] P.N. Hilfinger, "A High-Level Language and Silicon Compiler for Digital Signal Processing," *IEEE Custom Integrated Circuits Conference*, Portland, Oregon, pp. 213-216, May 1985.

[24] L. Nachtergaele, I. Bolsens, H.D. Man, "A Specification and Simulation Front-End for Hardware Synthesis of Digital Signal Processing Applications," *International Journal of Computer Simulation*, vol. 2, pp. 213-229, 1992.

[25] L. Nachtergaele and C. Scheers, "User Manual for the S2C Silage to C Compiler," IMEC Laboratory, Leuven, Belgium, 1993.

[26] R.A. Valenzula and A.G. Constantinides, "Digital signal processing schemes for efficient interpolation and decimation," *IEE Proceeding*, vol. 130, pt. G, no. 6, pp. 225-235, December 1983.

6

CONCLUSIONS

6.1 DISCUSSION OF RESULTS

The objective of this work was to develop techniques and design methods suitable for exploiting sigma-delta modulation in modern telecommunications applications using modern CMOS IC processes. Such systems are sufficiently varied that it is not possible to conceive of a general purpose design tool to solve most of the design problems, as is the case for filter design [1]. Thus some general purpose techniques and tools were assembled making it easier to describe and evaluate designs such that the designers' creativity is exploited and not inhibited. These tools have been used to develop new implementations of communications functions which have been used as examples throughout this thesis.

IC process technology is advancing at such a rate that new techniques are continually becoming feasible, while existing techniques are being superseded. The now widespread use of sigma-delta based ADCs and DACs for digital audio applications is an example of this. Ten years ago such techniques were not cost effective because of the expense of implementing the extensive digital filtering required. Today sigma-delta modulators are replacing more conventional conversion techniques even where the superior linearity is not important because the digital circuitry continues to shrink and the total cost of mostly digital solutions is always reducing. The compander discussed in Chapter 4 is also an example of this phenomenon - five years ago the mixed-signal solution would not have been appropriate (a BiCMOS solution was cheaper), while today it is a good solution, and in five years time a completely digital solution (with sigma-delta based ADC and DAC) is likely to be more suitable. Contributions to easing and speeding the design process for new techniques have formed the main achievement of this thesis.

Methods for designing and optimising the noise shaping and signal transfer function of high order sigma-delta modulators were given. The application of this to reducing the oversampling ratio required for audio signals was first reported in [2] and this publication also predates most in its discussion of the requirement for a dither signal

even with very high order sigma-delta modulators. The design tools developed for this were also used in the design of a sigma-delta modulator optimised for modem applications, where the sigma-delta modulator also acts as an input filter to prevent information loss due to limiting.

Redundant arithmetic exhibits an aptness for use in digital SDMs, in that it is the most significant digits of the loop filter output which are required quickly in sigma-delta modulation and these digits can be calculated very quickly when using redundant arithmetic. However the continual advance in digital IC processes and the lagging in the development of suitable integrated filters limits the requirement for these special techniques for high speed sigma-delta modulators. At present high speed DACs are most often implemented using multilevel current sources. The combination of redundant arithmetic SDMs and the FIRDAC filter using current source DAC elements, as discussed in Chapter 3, may become a viable technique for SDMs with sample rates greater than 20MHz. Some similarities of the FIRDAC to the emerging techniques for multi-level SDM DACs using dynamic element matching were also discussed.

The compander discussed in Chapter 4 makes good use of relative proficiencies of analogue and digital techniques. Analogue circuitry is used to provide an area-efficient solution while digital circuits are used to implement those functions which are very difficult to manufacture reliably with analogue techniques. In the future this split between analogue and digital will change. However other functions, probably operating at video frequencies (2-20MHz) are likely to benefit from this tightly coupled hybrid approach where digital circuitry controls analogue signal paths without digitising and reconstructing the signal.

The simulation methods for switched capacitor circuits provides several orders of magnitude quicker simulation, yet retains the ability to describe the circuit implementation as a schematic. This has made it possible to design complex SC functions such as the single multiplexed amplifier SDM and the gain-enhanced third order SDM with confidence that the topology is correct. This is important to ensure

functionality in the first implementation on silicon. Both the compander and the third order SDM were fully functional on the first returned silicon.

6.2 PROPOSALS FOR FUTURE WORK

The advance of IC technologies requires continual advances in design tools and methods.

Only a very limited number of topologies have been suggested for the implementation of high order SDM loop filters. This is unlike the case for general filter design. A study into alternate structures may yield systems with important practical advantages such as lower coefficient spread or lower sensitivity.

The choice of SBNR in implementing redundant arithmetic based sigma-delta modulators is probably inefficient. Further research into higher radix redundant arithmetic implementations may yield more efficient designs which still retain unlimited precision with no speed penalty. The associated problem of implementing high performance, high bandwidth analogue reconstruction filters on silicon requires much research since it is this component which limits the bandwidth of sigma-delta DACs today. Some of the techniques used at present for implementing high speed DACs, such as the array of current sources [3], could be extended to be used for FIRDACs. Further work could include a detailed study of the relative merits of the FIRDAC structure and emerging techniques for multi-level SDM DACs.

As mentioned above, future implementations of compander functions are likely to be all digital. Indeed the requirement for new compander designs is limited since most new communications standards are digital. However new compander implementations will be required to provide backwards compatibility for dual-mode cellular telephony, but are likely to implemented using code written for a general purpose digital signal processor already present in the design to handle the more complex modern digital communications standards. Thus future work in specific

area of audio companders is limited. However in the more general area of systems exploiting the single bit output of the sigma-delta modulators, the sensitivity to high frequency input components must be noted. The solution to this problem seems to be to use a simple FIRDAC instead of the single bit DAC multiplier as was discussed in Chapter 3. The requirements and implications of incorporating a simple FIRDAC should be studied for any future SDM compander designs.

It was noted in Chapter 4 that the compressor could also be considered as an compressing analogue to digital converter. A re-evaluation of this system in the context of a compressing ADC may suggest applications where this technique may be useful. The digitisation of telephone speech may be one application. If instead of the internal ADC being a second order SDM, it was a higher order SDM designed to have a low pass signal transfer function (using the methods discussed in Chapter 2) then it may be possible to dispense with or simplify the filter in the compressor topology.

A similar formulation to that used in the SC circuit simulator should be possible for switched-current circuits.

The SC simulation program could be extended to handle non-linear circuits with a complete settling assumption. A formulation for this is discussed in [4], but the formulation has not been developed into a full system simulator. The complete settling requirement could be separately checked using an linear SC simulation methods which includes resistive effects. Unfortunately, one of the main reasons that very good settling is required in SC circuit designs is to be sure that there are no non-linear effects, such as slewing which affects the settling. Thus including either resistive effects *or* non-linear effects individually gives limited additional information. This is one of the reasons why the ideal SC circuit simulator developed in this work has been so useful despite including neither of these effects. Including both resistive effects and non-linearity implies the use of the numerically intensive methods used in simulators such as SPICE and SABER. However, including non-

linearity would give an indication of the performance limit at low clock rates. It would not help determine at what clock frequency the performance begins to drop.

The TFN library described in the appendix could be usefully extended to allow multiple transfer functions to be used to describe the overall transfer function. The consequence of this is that it would allow the structure of the implementation to be represented as well as just the overall transfer function. This could be used to perform internal nodal scaling in both the frequency domain analysis and time-domain simulations. Additionally, the synthesis techniques of [5] could be incorporated. This would allow a specified transfer function equation to be transformed into the set of equations which correspond to a particular topology. The use of the Tcl-Tk library [6] would be appropriate to improving the interface with design tools making use of the polynomial library.

6.3 REFERENCES

[1] R.K. Henderson, *Computer-Aided Design of Switched Capacitor Filters*, PhD Thesis, University of Glasgow, Glasgow, December 1989.

[2] D.M. Hossack and J.I. Sewell, "Design of High Order Audio Sigma-Delta Modulators with Minimum Weighted Noise," *IEEE International Symposium on Circuits and Systems*, Chicago, Illinois, pp. 180-183, May 1993.

[3] Y. Nakamura, T. Miki, A. Maeda, H. Kondoh and N. Yazawa, "A 10-b 70-MS/s CMOS D/A Converter," *IEEE Journal of Solid State Circuits*, vol. 26, no. 4, pp. 637-642, April 1991.

[4] I. Yusim and K. Suyama, "Analysis of Switched-Capacitor and Switched_current Networks with Complete Settling Assumption," *IEEE International Symposium on Circuits and Systems*, Atlanta, Georgia, vol. IV, pp. 428-431, May 1996.

[5] L. Ping, R.K. Henderson and J.I. Sewell, "A Methodology for Integrated Ladder Filter Design," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 8, pp. 853-868, August 1991.

[6] J.K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, Reading MA, 1994.

7

APPENDIX - TRANSFER FUNCTION MANIPULATION LIBRARY (TFN)

7.1 INTRODUCTION

This appendix has been included to describe the syntax of TFN - a transfer function description and manipulation language which has been used to describe transfer functions throughout this thesis.

The transfer function manipulation library was written to allow both z and s domain transfer functions to be easily specified and manipulated. It has been incorporated into numerous programs for filter transfer function design and simulation. The determination of frequency domain response is often used by filter design programs to compensate for other fixed filters in the signal path. The time-domain simulation capability can be used to incorporate linear filter blocks in high level descriptions of systems.

Of specific interest are the facilities for designing and simulating sigma-delta modulators with user definable noise shaping and signal transfer functions.

To the user, the input format appears to be similar to a general mathematical package, highly specific for transfer functions (and consequently highly optimised). The polynomials are always reduced to a fixed but flexible data structure. Each factor may have an associated clock rate which indicates that it is a z domain function, and that when evaluated at a frequency f, the corresponding location on the unit circle is determined, rather than the corresponding location on the imaginary axis for s-domain factors.

A number of general mathematical packages exist, both commercially and non-commercially, which could be used to perform the functions of TFN. However because these packages have not been designed with transfer functions and signal processing in mind, they are less optimal and less easy to use than TFN. For example general symbolic packages such as MAPLE or Mathematica are much slower than TFN because of the generality of the their data structures. The speed at which

transfer functions may be generated and manipulated is not normally important. The speed at which the resulting transfer function can be used to filter sampled data signals is very important since it is common for high level simulations to require many millions of samples to be processed. More numerically orientated packages such as MATLAB require the transfer function to be forced into a fixed form suitable for the underlying matrix and vector data structures. The result is that the MATLAB Signal Processing package stores transfer functions in the coefficient form which can be numerically problematic. Furthermore these packages do not allow concepts such resonant frequencies and quality factors to be directly entered, although functions could be added to do this. The packages mentioned above offer very much more than what is available in TFN - but TFN is superior for transfer function description and manipulation.

7.2 MACHINE REPRESENTATION OF POLYNOMIALS

Rational polynomials are almost always held as a pair of polynomials, one for the numerator (zeros) and one for the denominator (poles). Polynomials may be held by a computer in a number of different forms. The data structure used in TFN has been chosen to general enough to encompass the two most common methods.

A polynomial may be described in coefficient form:

$$H(z) = \sum_{k=0}^{n} a_k z^k = a_k z^k + a_{k-1} z^{k-1} + \ldots a_2 z^2 + a_1 z + a_0$$

The coefficient form is most suitable for finite impulse response (FIR) where the coefficients are simply successive samples of the impulse response.

The alternative method is to hold the polynomial in factored form:

$$H(z) = \prod_{h=0}^{n} (z - z_h)$$

For transfer functions with real coefficients the factors occur either singly on the real axis, or in conjugate pairs. This restriction ensures that a real input signal is transformed to another real signal. The factored form has important advantages in numerical sensitivity and also in the design of algorithms to implement the transfer function. It is generally preferred compared to the coefficient form.

TFN holds a rational transfer function as a pair of polynomials, each of which is held as a linked list of factors. The factors are limited to have only real coefficients. Thus a conjugate pair of roots must be described as a single quadratic factor. The factors may be of any order (including zero order which is a constant gain factor). This method has the advantage that poles and zeros may be held in fully factored form to retain the numerically superior behaviour. Where ever possible the transformations available in TFN preserve the polynomials in as factored a form as possible.

7.3 TRANSFER FUNCTION INPUT FORMAT

The transfer function format was designed to be very simple to parse. The parser reads words separated by white-space from the input file. Each keyword read causes a number of further words to be read dependent on the particular keyword. For example the keyword "gain" takes three arguments - a numeric frequency value, a numeric gain value and a polynomial to which the gain term is applied. Typically the polynomial word will itself be a keyword causing the parser to recursively read more words. The "{" keyword is used to group a list of polynomials together until a trailing "}" is encountered. The result is a single polynomial.

For example:

```
gain 1k 1 {
     1:fo 0  / 2:foQ 1k 0.707
     }
```

This describes a 2nd order band-pass filter since it has a single first order zero at DC with a second order denominator. The resonant frequency is 1kHz with a quality factor Q of 0.707. The gain at 1kHz has been set to unity. This simple example has illustrated that the transfer function can be described in the terms that an engineer is most familiar with (resonant frequency, quality factor, gain) rather than as a list of coefficients or pole locations.

7.3.1 Valid Numerical Suffixes

Numerical suffixes which may be used to modify the value of a numerical constant are listed below. In particular the "dB" and the "rad" suffixes are useful in filter transfer function specification. The suffixes "k" and "M" are useful for specifying clock rates. All frequencies in TFN are in Hertz (Hz), but frequencies measured in radians may be entered and converted to Hertz by using the "rad" suffix. The "K" and "KK" suffixes are useful for specifying FFT lengths which must be a power of two.

Suffix:	Effect:
f	NUM * 1e-15
p	NUM * 1e-12
n	NUM * 1e-9
u	NUM * 1e-6
m	NUM * 1e-3
k	NUM * 1e+3
M	NUM * 1e+6
G	NUM * 1e+9
K	NUM * 1024
KK	NUM * 1024*1024
rad	NUM * 0.5/π
dB	$10^{NUM/20.0}$

7.3.2 Valid Transfer Function Keywords

The transfer function input routine reads a single keyword, and performs the corresponding operation, and returns the resulting polynomial. Typically the first keyword is "{" to read a number of factors. A keyword may be used where ever a transfer function is expected.

7.3.3 Simple factor definition

General n^{th} order

The general form of specifying a factor is an integer *n* denoting the order immediately followed by a colon and the *n+1* coefficient values starting with highest power. For example:

```
3:   1 2 3 1
```

corresponds to the factor $s^3 + 2s^2 + 3s + 1$. This is always interpreted as a s-domain polynomial, unless it is converted to z-domain by using a s to z domain transformation or directly converted using "fclk".

Zero Order

Constant gain factors may be expressed using the standard coefficient form notation or by using the "constant" keyword, or just the number itself.

0: x	zero order factor created with constant value x
constant x	zero order factor created with constant value x
x	zero order factor created with constant value x

Constants are by default s-domain although it does not matter if they are interpreted as z-domain.

First Order

First order factors may be expressed using the standard coefficient form or by specifying the root location (on real axis), or as a corresponding frequency or time-constant. When a frequency is specified, the root is assumed to be on the left hand side of the s-plane corresponding to minimum phase zeros or stable poles. Negative frequency values denote non-minimum phase zeros or unstable poles.

1:root x	first order factor corresponding to a root at (x,0)
1:fo f	first order factor corresponding to frequency f in Hertz
1:tc tc	first order factor corresponding to time constant "tc"
1:rc r c	first order factor corresponding to time constant r*c
1: a_1 a_0	general first order factor (a_1*s + a_0)

All of the above are created in the s-domain.

Second Order

Second order factors may be specified either in the standard coefficient form or by specifing the root location or a frequency and quality (Q) factor. Positive values of frequency result in minimum phase zeros or stable poles in the s-plane. If the Q value is not specified then the roots are assumed to be on the imaginary axis. This is useful for specifying zero pairs which correspond to notches in the frequency response.

2:root x y	2nd order factor created with root at (x,y) and (x,-y)
2:fo f	2nd order factor created with root at (0,2πf)
2:foQ f Q	2nd order factor created corresponding to fo and Q values
2: a_2 a_1 a_0	general 2nd order factor

All of the above are created in the s-domain.

Classical Filter Transfer Function Design

Polynomials may also be specified as being a classical transfer function. The resulting polynomial is normalised so that at the frequency 1 rad/s corresponds to the -3dB frequency (Butterworth) or the pass-band edge (Chebyshev type I) or the stop-band edge (Chebyshev type II, also known as inverse Chebyshev). The comb filter facility gives a polynomial with a specified number of unity coefficients. This is useful for the comb filter decimators and interpolators used in sigma-delta modulator systems.

butterworth n	Butterworth filter order n
chebyshev n passband_ripple	Chebyshev type I order n
inv_cheby n stopband_ripple	Chebyshev type II order n
comb n	comb filter length n (order n-1)

All of the above with the exception of "comb" create fully factored polynomials in the s-domain. The "comb" instruction creates a z-domain factor with a unity sample rate.

Transfer Function Transformations

Factors and polynomials may be transformed and combined in a number of ways. Factors are created as s-domain factors and must be converted to z domain if necessary, by an appropriate function such as fclk, matched_z and bilinear.

/ POLY	swap numerator and denominator (used to implement the denominator)
fclk POLY	apply clock rate (coefficients unaltered)
matched_z fc POLY	matched-z transform assuming clock rate of fc
bilinear f1 f2 fc POLY	bilinear transform, mapping f1 to f2 assuming a clock rate of fc
modified_bilinear f1 f2 fc POLY	bilinear transform mapping f1 to f2 assuming a clock rate of fc without mapping s domain zeros at infinity
freq_scale f1 f2 POLY	s domain low pass - to low pass transform, f1 maps to f2
highpass f1 f2 POLY	s domain low pass - to high pass transform, f1 maps to f2
transform POLY1 POLY2	general transformation of POLY1(x) using $x \leftarrow$ POLY2(x)
bandpass f1 f2 f3 POLY	s domain lowpass to bandpass transform, f1 maps to f2 and f3
add POLY1 POLY2	addition, numerators of polynomials will be expanded first
expand POLY	multiply out all factors in POLY
factorise POLY	determine first and second order factors of POLY
cascade n POLY	duplicate POLY n times

7.3.4 Other Manipulations

gain f g POLY	evaluate POLY at f and add zero order factor to make magnitude g
rm_constant POLY	delete all zero order factors
{ POLY1 ... POLYN }	read polynomials until "}" and treat as one entity
quantise n POLY	quantise coefficients of POLY to nearest 2^{-n}
zeros_of POLY	return numerator of POLY, discard denominator
poles_of POLY	return denominator of POLY, discard numerator
NTF_to_H POLY	convert noise transfer fn to loop filter
H_to_NTF POLY	convert sigma-delta loop filter to equivalent noise transfer fn

7.4 STAND-ALONE PROGRAMS

Once a transfer function has been specified, by the far the most common uses of it are:

1. plot the frequency response (ac response)

2. plot the time domain response (transient response)

As an illustration, the following describes a fifth order elliptic filter with quantised coefficients which has been realised as the sum of two parallel paths, one containing a 2nd order all-pass filter, and the other a 3rd order all-pass filter.

```
fclk 500 {
0.5
add
     quantise 8 {
          2: 0.141349 0 1
          / { 2: 1 0 0.141349 }
          }
     quantise 8 {
          2: 0.589995 0 1
          / { 2: 1 0 0.589995
               1: 1 0
               }
          }
     }
```

The resulting transfer function has a fifth order coefficient form numerator (because the addition must be performed on polynomials in coefficient form), and a fifth order factored denominator (because the addition doesn't need to expand the denominator). In general, all the manipulations retain the factored form if this is possible. If a factored form was required in the numerator, this could be explicitly requested by preceding the whole transfer function by the keyword "factorise".

The stand alone program test_poly can be used to observe the above. The usage is:

```
test_poly "tfn"
```

where "tfn" is the file name containing the transfer function description. "test_poly" does nothing more than read the transfer function and print it out, yet it is invaluable for checking transfer functions when they are in their initial development.

To then check the frequency response of the transfer function, the following is used:

```
response "tfn" f_start f_stop num_pts [lin|log]
```

If f_start is zero then the linear option is default, otherwise the log option is default. This evaluates the transfer function at num_pts equally spaced points on a linear or

logarithmic axis between the frequencies f_start and f_stop. The above transfer function was specified as being z-domain by use of the "fclk" keyword, and so the transfer function will be evaluated at the complex locations around the unit circle. A mixture of s and z domain factors is permissible. For example, if the effect of sin x / x distortion is important, then this can be added to the transfer function with:

```
gain 0 0dB {
     fclk 500 { 1: 1 -1 }
     / { 1:fo 0 }
     }
```

This contains a z domain function to implement the "sin x" part, and a s domain part to implement the " / x " part. This cannot be evaluated at zero frequency (DC) due to the singularity. Mixing s and z domain factors is also useful for taking into account continuous time anti-aliasing and smoothing filters when determining overall system responses.

Having checked the frequency response it is often desirable to study the effect on an input signal in the time domain. This is achieved for sampled data systems in a completely general manner by reading a file of input samples and outputting the corresponding filtered sampled data signal. This is currently only applicable for pure z-domain functions which a consistent sample rate for all factors.

The usage of the time domain program is straight forward:

```
time_domain "tfn" < "input" > "output"
```

where "tfn" is the file containing the description of the transfer function and "input" is a file containing the input samples (in ASCII text format) separated by white space.

For example to plot the impulse response the following file might be used:

```
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

and for the step response:

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Other more complex signals can be created, e.g. for studying the effect of a data recovery filter on the resulting eye diagram.

Often the three standalone programs above are all that is required. Sometimes however the transfer function is just a small part of a more complex system and it is useful to use the above facilities in other C programs. A library containing all the facilities described above is available for this purpose.

7.5 EXAMPLE

This example is for a third order sigma-delta modulator with a clock rate of 1.536MHz. The NTF has a notch at 4kHz and the STF has been given a gain of -6dB so that the modulator remains stable for inputs of magnitude up to the reference signal.

tf_ntf:

```
rm_constant matched_z 1M {
     1:fo 0
     2:fo 4k
     1 / poles_of highpass 1rad 55k butterworth 3
     }
```

tf_stf:

```
expand gain 0 -6dB
     fclk 1M 1 { 3: 1 0 0 0
          poles_of { file tf_ntf }
          }
```

The noise transfer function is based on a third order Butterworth high-pass filter with a -3dB point at 55kHz. The zeros associated with this have been discarded by using "1 / poles_of" so that the zeros can be explicitly placed (a pair a 4kHz and the remaining one at dc). To satisfy the realisability constraint of having a delay in the closed loop, the multiplying constant of the poles and zeros has been set to unity using "rm_constant". The STF has been formed of the same poles as the NTF but without the zeros so that a 3rd order low-pass response results. The responses can be checked using:

```
response tf_ntf 100 500e3 1000
response tf_stf 100 500e3 1000
```

The resulting graphs are plotted in Figure A-1.

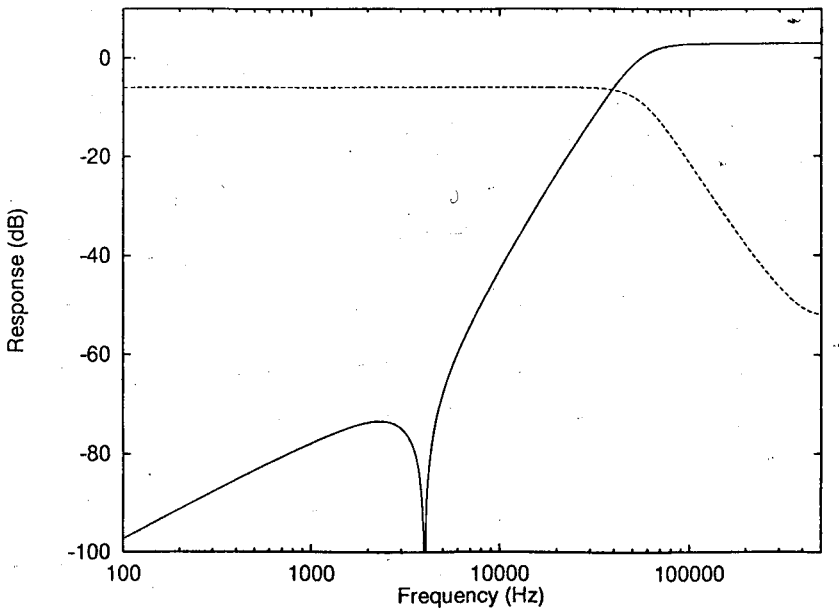


Figure A-1 Response of the NTF and STF.

A sigma-delta modulator simulation may be performed using the command:

```
ac 65536 999 1e6 0.5 | sdm tf_ntf tf_stf 2
```

This generates 65536 samples of a sinusoid of frequency 999Hz with an amplitude of 0.5 as input to the sigma-delta modulator with 2 quantisation levels. The resulting spectrum has been plotted in Figure A-2.

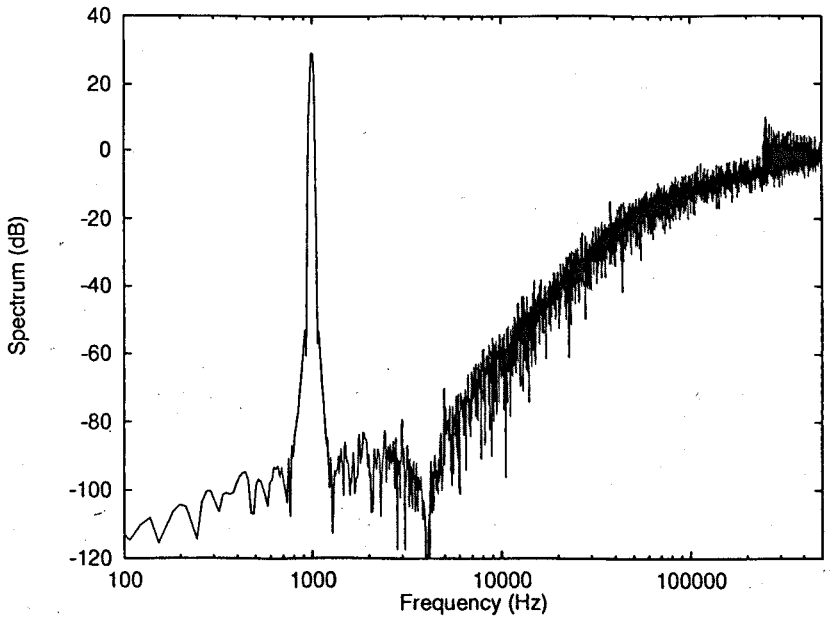


Figure A-2 Spectrum from sigma-delta modulator simulation.