AN ABSTRACT OF THE THESIS OF

Haiqing Lin for the degree of Doctor of Philosophy in Electrical & Computer Engineering presented on May 8th, 1998.

Title: Multi-bit Delta-Sigma Switched-Capacitor DACs Employing Element-Mismatch-Shaping.

Redacted for privacy

Abstract approved: _____
Richard Schreier

Delta-sigma modulators are currently a very popular technique for making high-resolution analog-to-digital and digital-to-analog converters (ADCs and DACs). Most delta-sigma modulators in production today employ single-bit quantization because a 1-bit DAC is inherently linear, whereas a multi-bit DAC is not. Were it not for this drawback, the use of multi-bit quantization would improve a delta-sigma modulator's performance by increasing the modulator's resolution or increasing the modulators's bandwidth, while at the same time whitening the quantization noise and improving modulator stability. This thesis explores the element-mismatch-shaping technique, which attenuates the noise caused by static element mismatch in a multi-level DAC by a method similar to delta-sigma modulation.

Existing element-matching techniques are reviewed and some analytical and architectural work related to the realization of mismatch-shaping logic is presented. A custom switched-capacitor (SC) DAC is used to verify various element mismatch-shaping algorithms. Experiments show that mismatch-shaping can reduce harmonic distortion by up to 30 dB.

Multi-bit Delta-Sigma Switched-Capacitor DACs Employing Element-Mismatch-Shaping

by

Haiqing Lin

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed May 15, 1998
Commencement June 1998

Doctor of Philosophy thesis of Haiqing Lin presented on May 8, 1998

APPROVED:

Redacted for privacy

Major Professor, representing Electrical & Computer Engineering

Redacted for privacy

Chair of Department of Electrical & Computer Engineering

Redacted for privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Haiqing Lin, Author

ACKNOWLEDGMENTS

My respect and gratitude to my advisor Prof. Richard Schreier is beyond my words. I am indebted to him for introducing me to the wonderful world of delta-sigma modulators and mixed-mode IC design. I wish to thank him for his guidance, support, help and encouragement, without which this work would not have been possible.

Many thanks to Prof. Dwight J. Bushnell of the Mechanical Engineering Department for taking time out of his busy schedule to serve as the Graduate Council Representative on my graduate committee.

I would like to thank my graduate committee members: Prof. Un-Ku Moon, Prof. John T. Stonick, Prof. Gabor C. Temes and other ECE Department faculty members such as: Prof. David J. Allstot and Prof. Sayfe Kiaei for valuable suggestions, comments and help. Thanks to Rita Wells, Sarah O'Leary, Kim Rowe, Jodi Nickerson, Sharon Ortiz, Jennifer Heim, Otto Gygax, Roger Traylor, Tim Heldt and other ECE department staff and support members for their support throughout my time here.

Special thanks to Luis Hernandez and Forrest Hudson for helping me to setup the test bed. Thanks also go to my colleague Bo Zhang for helping me to start my study and research at OSU. I will never be able to thank Bo and his wife Bing enough. Without their unselfish help during my early days in this country, my life would have been much harder.

I would also like to thank David Robertson, Todd Brooks, Doug Mercer, Paul Ferguson, Dave Carr, Chris Yaceshyn and other friends at Analog Devices, Inc. for a fruitful summer internship, the test board for my first prototype DAC chip, and the laser surgery of my second prototype chip. I am also grateful to Dru Cabler, Joe Peterson,

Geoffrey Brehmer and other friends at Advanced Micro Devices, Inc. for my second summer internship which was as great as the first.

Thanks to the National Science Foundation (NSF) Center for the Design of Analog/Digital ICs (CDADIC), Westinghouse Electric Corporation and Northrop Grumman Corporation for their generous support, and to Orbit Semiconductor, Inc. for their support in chip fabrication.

Thanks to my Chinese friends Bo Wang, Yunteng Huang, Tao Shui, Min Shen, Yihai Xiang, Yuming Cao, Tao Sun, Xiangping Qiu and their lovely families, with whom I shared many great times in Corvallis. Thanks also go to the OSU international advisors and many other friends whom I am not able to mention one by one due to the limited length of this acknowledgment.

Special thanks to my parents, for their constant support and encouragement in all aspects of life and study, for their love, help, advice, backup and belief in me.

Finally, thanks to my beautiful and lovely bride, Ying, for her deep love and blessings, and for filling my life with happiness.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure | Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF APPENDICES

Page

LIST OF APPENDIX FIGURES

LIST OF APPENDIX FIGURES (Continued)

Figure Page

Multi-bit Delta-Sigma Switched-Capacitor DACs Employing Element-Mismatch-Shaping

Chapter 1. Introduction

By a process similar to delta-sigma modulation [1, 2], the noise caused by static element mismatch in a multi-level digital-to-analog converter (DAC) [3] can be attenuated in a narrow frequency band. The purpose of this thesis is to explore this mismatch-shaping technique for high-resolution wide-band multi-bit delta-sigma switched-capacitor (SC) DACs.

1.1 Motivation

Because of its insensitivity to many analog imperfections, delta-sigma modulation [1, 2] has become one of the most popular techniques for making high-resolution ADCs (analog-to-digital converters) and DACs which are widely used for data conversion at the front-end of mixed-signal ICs.

Most delta-sigma modulators in production today employ single-bit quantization. A key feature of an ADC or DAC based on one-bit delta-sigma modulation is its ability to achieve arbitrarily high linearity. This feature is the result of the inherent linearity of a one-bit DAC. A multi-bit DAC cannot be made perfectly linear since it is impossible to create a transfer characteristic with steps of exactly equal height. Since non-idealities in the DAC are equivalent to noise added directly to the input signal of a delta-sigma ADC (or to the output in delta-sigma DAC), DAC errors can easily be the performance-limiting factor in a delta-sigma converter.

Were it not for this drawback, the use of multi-bit quantization would improve a delta-sigma modulator's performance by increasing the modulator's resolution or increasing the modulators's bandwidth, while at the same time, whitening the quantization noise and improving modulator stability. These powerful incentives have led a number of researchers to develop such schemes as element-randomization [4-6], individual-level averaging [7, 8], and digital calibration [9] to combat the problems caused by imperfect DACs.

More recently, schemes were invented which make the errors caused by mismatch in a multi-bit unit-element DAC appear as first-order shaped noise [10-15]. One of the methods has since been generalized to arbitrary-order lowpass and bandpass shaping in [16]. This thesis explores the implementation of the mismatch-shaping technique, presenting new schemes which are more hardware-efficient, and designing prototype ICs to corroborate the theory.

1.2 Thesis Outline

The fundamentals of delta-sigma modulation are reviewed in Chapter 2, along with brief descriptions of some existing element-matching techniques. Chapter 3 presents some analytical and architectural work related to the realization of mismatch-shaping logic. A generalized butterfly shuffler scheme capable of implementing various noise transfer functions (NTFs) is also presented in Chapter 3. Chapter 4 explores the possibility of applying mismatch-shaping to non-unit-element DACs by designing a 2-segmented DAC with mismatch-shaping only on the MSB part. A 16-element SC DAC which has been implemented on a prototype IC, along with some high-order delta-sigma modulator designs is given in Chapter 5. The experimental results are presented in Chapter 6. Chapter 7 summarizes the work and gives a few suggestions for future work.

Chapter 2. Background: Delta-Sigma Modulation and Mismatch-Shaping

This chapter presents the material necessary to understand ΔΣ modulators in general and this thesis in particular. Starting with a brief review of quantization, the discussion progresses to 1st-order, then high-order and bandpass ΔΣ modulators. Multi-bit modulators are introduced thereafter. At the end of the chapter, the effect of DAC non-linearity in ΔΣ modulators is discussed along with some means for dealing with this important issue.

2.1 Quantization

2.1.1 Quantization and Quantization Noise

Quantization refers to the process of subdividing a quantity into a small but measurable increment. In electrical engineering, such a quantity can be the magnitude of an analog signal, such as voltage or current. It can even be a digital signal if quantization involves reducing the bit-width. A clock is needed to sample the signal and to initiate the quantization. Periodically sampling a continuous signal at a rate more than twice the signal bandwidth does not introduce distortion, but quantization does.

The most commonly used quantization is uniform quantization, which has equal quantization steps denoted as Δ. An example is shown in Figure 2.1 (a), where the continuous amplitude signal x is rounded to the nearest of 8 discrete levels. It is evident that the quantized signal y can be represented by

$$y = Gx + e, \tag{2.1}$$

where the gain G is the slope of the straight line that passes through the center of the quantization characteristic, and e is the quantization error shown in the same picture.

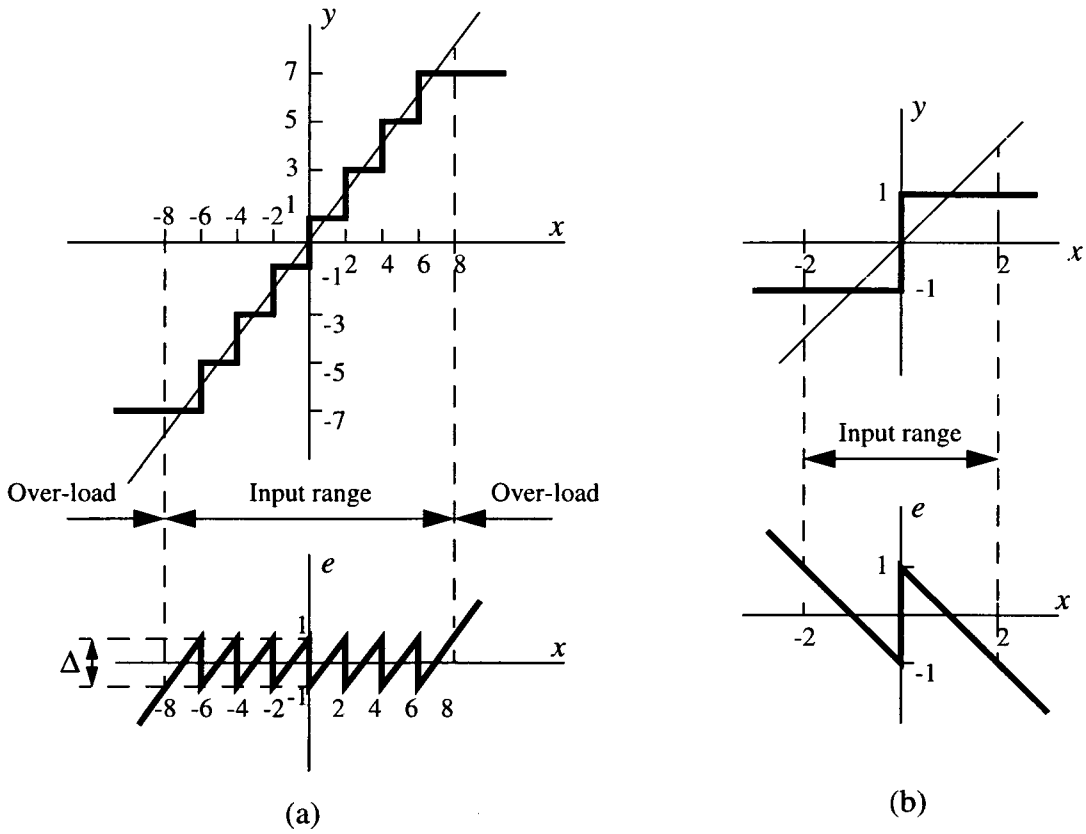


Figure 2.1: (a) An example of uniform multi-level quantization characteristic. (b) Single-bit quantization.

Other non-ideal effects, such as dc offset and non-linear gain error, can be added into Eq. (2.1) when necessary.

If there are a total of 2^n quantization levels, we call it an n-bit quantization. For the example shown in Figure 2.1 (a) the quantizer is a 3-bit quantizer. If the input range is normalized to ± 1, the quantization step, or the value of the least-significant-bit (LSB) is then given by $\Delta = 2/(2^n - 1)$. The error e is completely defined by the input, but if the input changes randomly between samples by amounts comparable with or greater than Δ, and without saturating the quantizer, then e is largely a random variable uniformly distributed in the range $\pm\Delta/2$. Under these assumptions the mean-square value of e, i.e. the power of the quantization error is given by

$$\sigma_e^2 = \frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 de = \frac{\Delta^2}{12}. \qquad (2.2)$$

Through simple calculations, the peak signal-to-noise ratio (SNR) of an ideal n-bit linear quantizer with a full-scale sine-wave input can be found to be approximately $6n$ dB.

Notice that the above consideration remains applicable to a 2-level (single-bit) quantizer, as illustrated in Figure 2.1 (b), but in this case the choice of the gain G is arbitrary.

In conventional Nyquist rate ADCs or DACs. higher resolution is achieved by using smaller step sizes (i.e. increasing n), which requires precisely matched analog components. As a result, the practical limit with current (untrimmed) circuit techniques is about 14 bits of resolution. Trimmed circuits can achieve 16 or more bits of resolution, but are expensive. In a Nyquist rate ADC or DAC, precision analog circuits, such as high-gain op amps, linear integrators, etc., have no opportunity to exert their power because a complete conversion must be performed in every clock period.

2.1.2 Oversampling in Quantization

Oversampling [1, 2] is simply the process of sampling faster than the Nyquist criterion requires. If the signal occupies the band from DC to f_B and the sampling rate is f_s, the oversampling ratio (OSR) R is defined as $R = f_s/(2f_B)$. Oversampling eases the anti-alias filter design since a wide transition band is created by the increased separation between the signal band and its first alias. For quantizers with broadband quantization noise, oversampling also reduces the amount of in-band quantization noise. This allows the conversion to be more accurate than the resolution of the quantizer. Specifically, each

octave increase in the OSR results an increase in resolution of 3 dB (0.5 bit). As the next section will show, ΔΣ modulation improves significantly on this trade-off.

2.2 First-Order Delta-Sigma Modulator

A 1st-order lowpass ΔΣ modulator with a single-bit quantizer is shown in Figure 2.2. Since the loop contains two operations: subtraction ("Δ") of the fed back output signal from the input signal and integration ("Σ") of the differences, it is called a "delta-sigma" modulator. The modulator works in the following way: if an error is introduced at the quantizer at one time step, it will be subtracted from the next output so that the output tracks the input at low frequencies. From a system level point of view, the negative feedback forces the DC value of the modulator output, v, to be the same as the input signal, u, so that the integrator, which otherwise would be unstable, sees an input with zero DC content. Therefore, a 1st-order ΔΣ modulator has ideally unlimited resolution for DC signals.

The z-domain description of the system is

$$V(z) = z^{-1}U(z) + (1 - z^{-1})E(z), \tag{2.3}$$

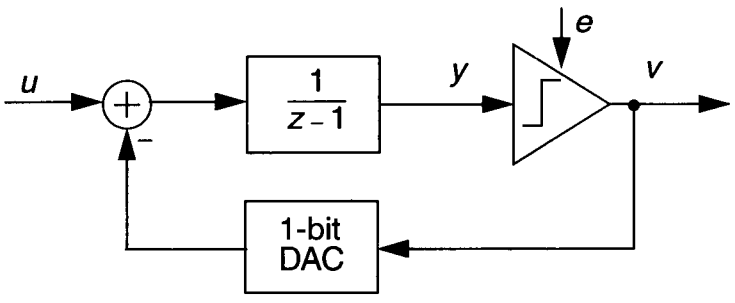


Figure 2.2: A first-order single-bit ΔΣ modulator.

where $v(n)$ is a discrete-time 2-level signal, and E is the quantization error [17]. According to Eq. (2.3), the quantization error is frequency-shaped by the function $H(z) = 1 - z^{-1}$. $H(z)$ is called the noise transfer function (NTF) which, in this example, has a zero at DC and thus suppresses the quantization noise in the vicinity of DC. Likewise, the input signal U is also frequency-shaped by a function called signal transfer function (STF) $G(z)$ which is simply z^{-1} in this example. An STF can be viewed as the signal gain of the modulator and should be unity or a constant within the-band-of-interest. The spectrum of the output of MOD1 in Figure 2.3 clearly shows the shaping of the quantization noise.

Assuming that E is white with power σ_e^2, the in-band noise power is given as

$$N_0^2 = \frac{\sigma_e^2}{\pi}\int_0^{\frac{\pi}{R}} |H(e^{j\omega})|^2 d\omega \approx \frac{\sigma_e^2}{\pi}\int_0^{\frac{\pi}{R}} \omega^2 d\omega = \frac{\sigma_e^2 \pi^2}{3R^3}, \qquad (2.4)$$

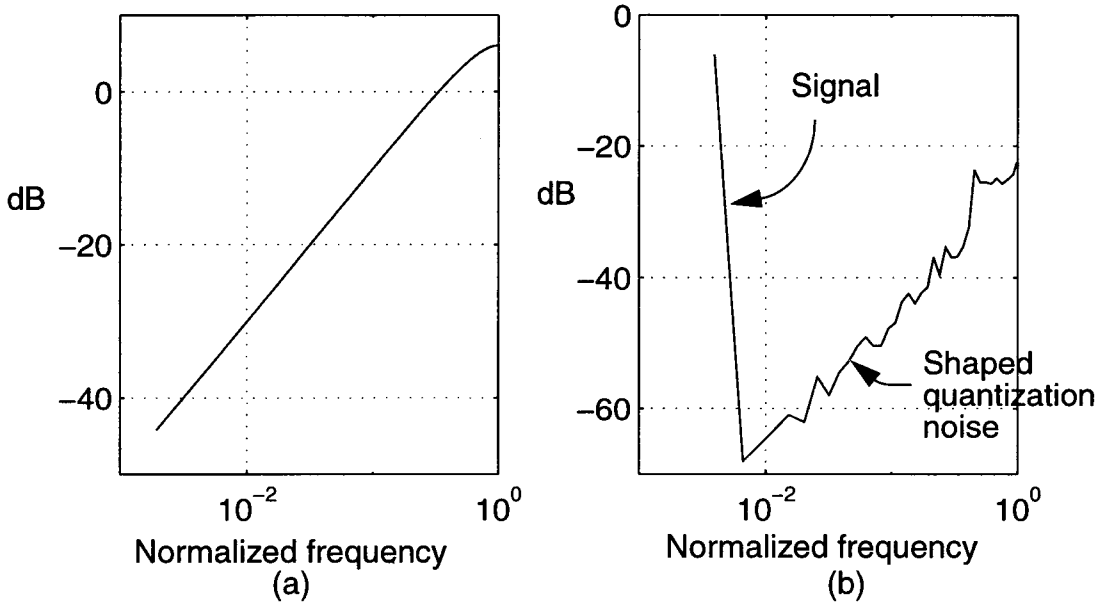


Figure 2.3: (a) The frequency response of $1 - z^{-1}$, and (b) the simulated spectrum of first-order $\Delta\Sigma$ modulator with sine-wave input shows the shaping of quantization noise.

where R is the OSR. If E is uniformly distributed in $[-1, 1]$, according to Eq. (2.2), $\sigma_e^2 = \frac{\Delta^2}{12} = \frac{1}{3}$.

Eq. (2.4) predicts that an octave (factor of two) increase in R will increase the SNR by 9 dB. In principle, the in-band noise can be made as small as desired, simply by making R large enough. Thus, 1st-order $\Delta\Sigma$ modulators have potentially unlimited accuracy, independent of component matching and many other idealities. In general, the resolution of a $\Delta\Sigma$ converter is improved by clocking faster (which is easy) and not by making larger, more sensitive analog circuitry (which is hard).

An important property of single-bit modulators is what is often referred to as "inherent linearity" [18]. This property comes from the fact that the input-output transfer curve of any static two-level DAC can be modeled exactly by a straight line joining the two points on the curve. A binary DAC is therefore ideal and cannot introduce errors other than simple offset and gain errors. These errors do not introduce distortion and the conversion is "linear."

The primary disadvantage of 1st-order $\Delta\Sigma$ modulators is that a high oversampling ratio is needed to achieve high resolution. For example, if we want 16-bit resolution, the oversampling ratio must be about 1500. Except for very low-frequency applications, a high oversampling ratio leads to a high sampling frequency and thus difficulties in implementation. As the next section will show, the oversampling ratio required to achieve a given resolution can be made smaller if higher-order $\Delta\Sigma$ modulators are used.

2.3 High Order Delta-Sigma Modulators

Figure 2.4 shows a general model of a ΔΣ modulator followed by a decimation filter. The modulator consists of three parts: a loop filter, a quantizer and a feedback DAC. Modeling the quantizer with $V = Y + E$, the output of the modulator is

$$V(z) = G(z)U(z) + H(z)E(z), \tag{2.5}$$

where $G(z)$ and $H(z)$ are the STF and the NTF, respectively, of the modulator. To achieve spectral separation between signal and noise, the STF must be approximately one in the band of interest whereas the NTF must be approximately zero. The decimation filter eliminates the out-of-band noise and downsamples the signal to the Nyquist rate. Decimation is beyond the scope of this thesis, and will not be discussed further.

The NTF of the 1st-order ΔΣ modulator discussed in the previous section has a zero at DC and thus suppresses the quantization noise in the vicinity of DC by a first-order

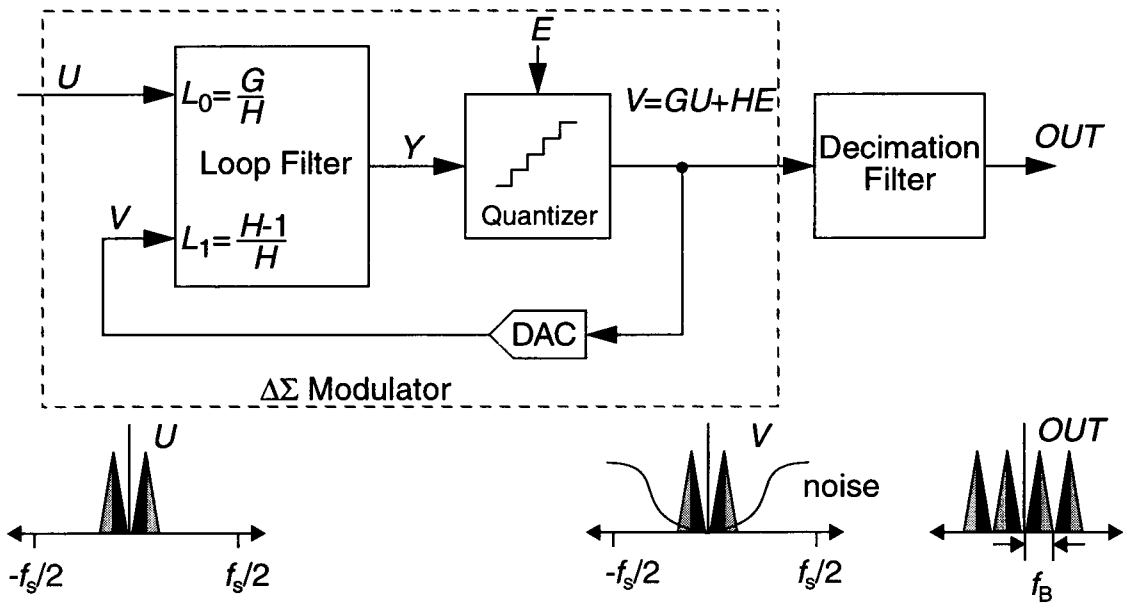


Figure 2.4: A general ΔΣ modulator.

filtering. More aggressive suppression of the in-band quantization noise can be achieved by constructing higher order NTFs which place more than one zero at DC. Figure 2.5 shows a simple 2nd-order ΔΣ modulator. The loop filter consists of two integrators in cascade. The feedback DAC is omitted for the sake of simplicity. The output of the modulator is

$$V(z) = z^{-1}U(z) + (1 - z^{-1})^2 E(z), \tag{2.6}$$

so the NTF is $H(z) = (1 - z^{-1})^2$; $H(z)$ has two zeros at DC. Assuming that E is white with power σ_e^2, the in-band noise power is then given as

$$N_0^2 = \frac{\sigma_e^2}{\pi}\int_0^{\frac{\pi}{R}} |H(e^{j\omega})|^2 d\omega \approx \frac{\sigma_e^2 \pi^4}{5R^5}. \tag{2.7}$$

Eq. (2.7) predicts that an octave (factor of two) increase in R will increase the SNR by 15 dB, which, compared with the 1st-order ΔΣ modulator, is a 6 dB/octave improvement.

Even higher order modulators can be built by employing higher order NTFs. Also, the positions of the zeros in high order NTFs do not need to be all at DC. Certain loop filter coefficients can be chosen to split the zeros around DC to optimize the in-band NTF response.

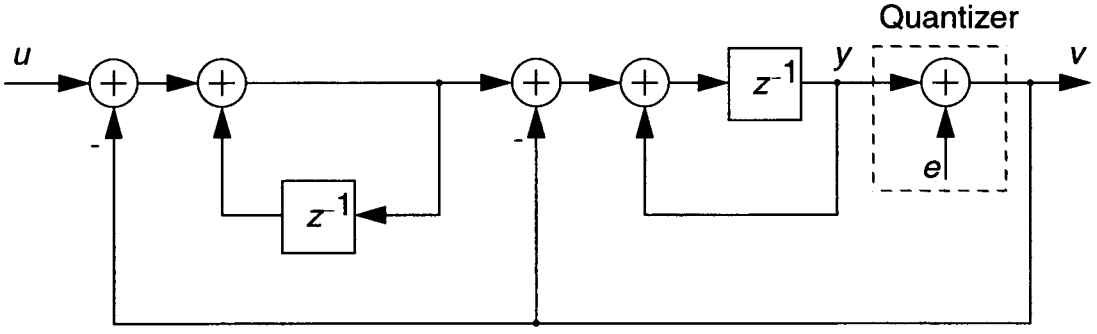


Figure 2.5: Second-order ΔΣ modulator.

In deriving the transfer functions, the quantizer was modeled as $v = y + e$. This allows the quantizer, a nonlinear system, to be treated as a linear system with independent inputs y and e. Note that e and y are in reality functionally related and therefore not independent. For linear systems with stable transfer functions, the output and the internal states are bounded if the input is bounded. However, the same cannot be said for $\Delta\Sigma$ modulators with stable signal and noise transfer functions, because e is not a real input—e is derived from y, in a nonlinear way. It is possible for $G(z)$ and $H(z)$ to be stable and yet result in a modulator with unbounded internal states. For a 1^{st}-order modulator, it is easy to show [18] that if $|u| \leq 1$, then the state y is bounded with $|e| \leq 1$. But for higher-order modulators, the stability issue is much more complicated [19-27].

2.4 Bandpass Delta-Sigma Modulators

A lowpass $\Delta\Sigma$ modulator places NTF zeros near $\omega = 0$ in order to null quantization noise in a narrow band around DC. Noise-shaping can be extended to the bandpass case simply by placing the NTF zeros at a non-zero frequency ω_0. Quantization noise is then suppressed in a narrow band around ω_0, and the output bit-stream accurately represents the input signal in this narrow band. A system which achieves this result is called a bandpass $\Delta\Sigma$ modulator [28-30]. Figure 2.6 compares the pole/zero locations of a lowpass NTF and a bandpass NTF.

A conceptual diagram of a bandpass $\Delta\Sigma$ ADC is given in Figure 2.7. A narrowband analog input is converted directly to a 1 bit oversampled digital signal with noise-shaping around the passband. The decimation filter suppresses the out-of-band quantization noise and downsamples the filtered signal to a baseband digital signal.

The obvious advantage of a bandpass ΔΣ modulator is that the sampling frequency is only several times $(2\pi/\omega_0)$ greater than the signal center frequency ω_0. A direct implementation of lowpass ΔΣ modulator for an intermediate-frequency (IF) or radio-frequency (RF) narrow band signal would result in a sampling rate which is much higher than that required by a bandpass modulator.

Bandpass ΔΣ modulators are most attractive for the conversion of high-frequency narrowband signals to digital form [28-32]. However, in such applications, discrete-time

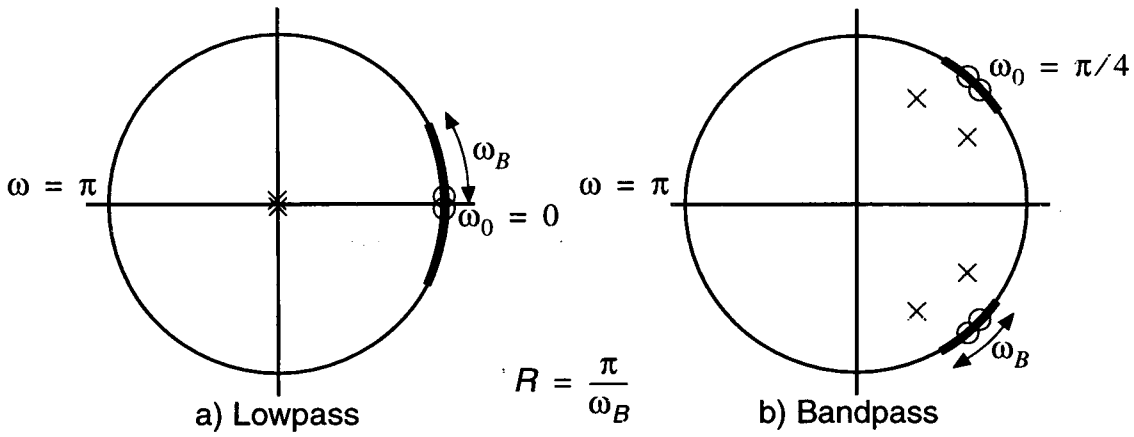


Figure 2.6: The passbands and pole/zero locations of the noise transfer functions for (a) lowpass and (b) bandpass ΔΣ modulation.

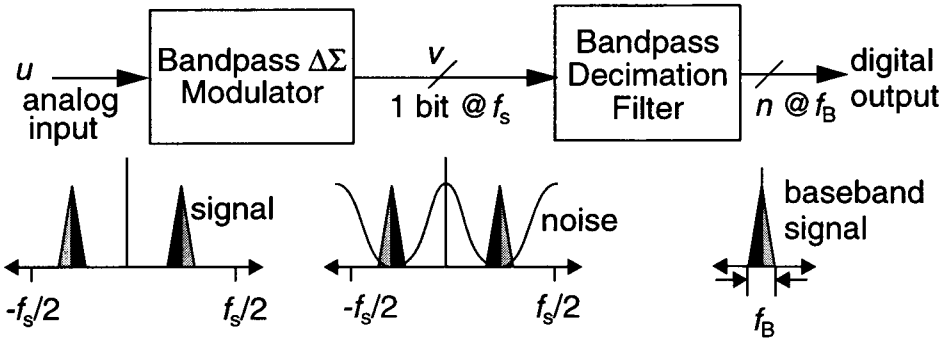


Figure 2.7: A diagram of a bandpass ΔΣ ADC.

modulators must sample fast-slewing signals and will therefore require a sample-and-hold circuit with low aperture error.

2.5 Multi-bit Delta-Sigma Modulator

Although most ΔΣ modulators in production today employ single-bit quantization, the discussions in the previous sections can be applied to systems with multi-bit quantizers. The same 1^{st}-order, high-order, and bandpass modulation structures can be realized with multiple quantization levels. The advantage is obvious: since the quantization step Δ in Eq. (2.2) is now smaller, the overall power of the quantization error σ_e^2 is reduced. Therefore, higher resolution can be achieved. From another point of view, since the noise level is lower, there is now more room for us to expand the signal bandwidth by decreasing the OSR while at same time maintaining the same resolution. The stability is also improved. This improvement in the stability makes higher-order and more aggressive NTFs feasible. Furthermore, the design of the decimation filter after a multi-bit ΔΣ ADC or the analog reconstruction filter after a DAC is eased and can have a lower order since the out-of-band noise floor is lower. In other words, the unfiltered output more closely resembles the desired output in the multi-bit case. Also, in a DAC system the slew rate requirements of the analog output stage can be relaxed and the system has better immunity to edge jitter due to smaller output steps. Lastly, in an ADC system where the loop filter processes the difference between the input and the fed-back output, loop filter linearity is much less of a problem with multi-bit feedback since the difference signal is small. A more subtle advantage of multi-bit quantization is that it helps to whiten the quantization noise so that the modulator suffers less from tone problems.

Despite all these advantages, there is one major obstacle which stands in the way of using multi-bit quantizers in ΔΣ modulators: DAC nonlinearity. Figure 2.8 shows how

DAC non-idealities (D) affect ΔΣ modulators. In a ΔΣ ADC system, D is frequency-shaped by $(H-1)$. Since the NTF H has to be approximately zero within the band-of-interest in order to attenuate the quantization error E, D can not be suppressed at the same frequencies. In a ΔΣ DAC system, non-idealities in the DAC are equivalent to noise added directly to the output. Therefore, DAC mismatch errors can easily be the performance-limiting factors in a ΔΣ ADC or DAC. On the other hand, a binary DAC is "inherently linear" and so does not require precise component matching. That is why single-bit quantizers became so popular that many people misunderstood it as a peculiarity of ΔΣ modulation and proudly printed "1-bit DAC" on their products.

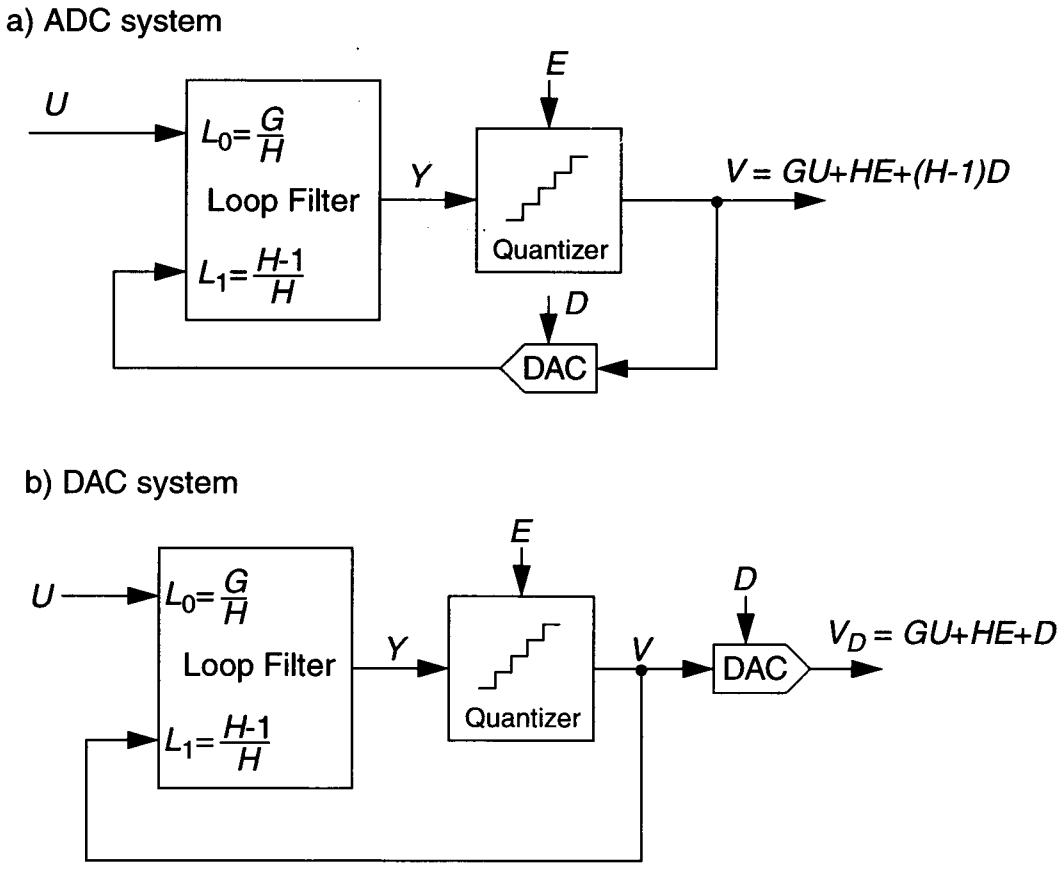


Figure 2.8: A multi-bit ΔΣ modulator.

2.6 Existing Dynamic Matching Schemes

The powerful incentives for using multi-bit ΔΣ modulators have led a number of researchers to employ such technologies as laser trimming, digital calibration [9], and dynamic element-matching [4-8] [10-16] [33-37] to combat the problems caused by component mismatch. This thesis focuses on developing dynamic element-matching schemes. Some of the existing dynamic matching architectures are introduced in the following sections.

2.6.1 Parallel Single-Bit Modulators

The most straight-forward way of making an m-level ΔΣ DAC insensitive to element mismatch is to use m parallel single-bit modulators to drive m one-bit DACs, as shown in Figure 2.9. The output of the system (neglecting DC offsets) is

$$V_D(z) = KU(z) + H(z)(E_1(z) + \ldots + E_m(z)) \tag{2.8}$$

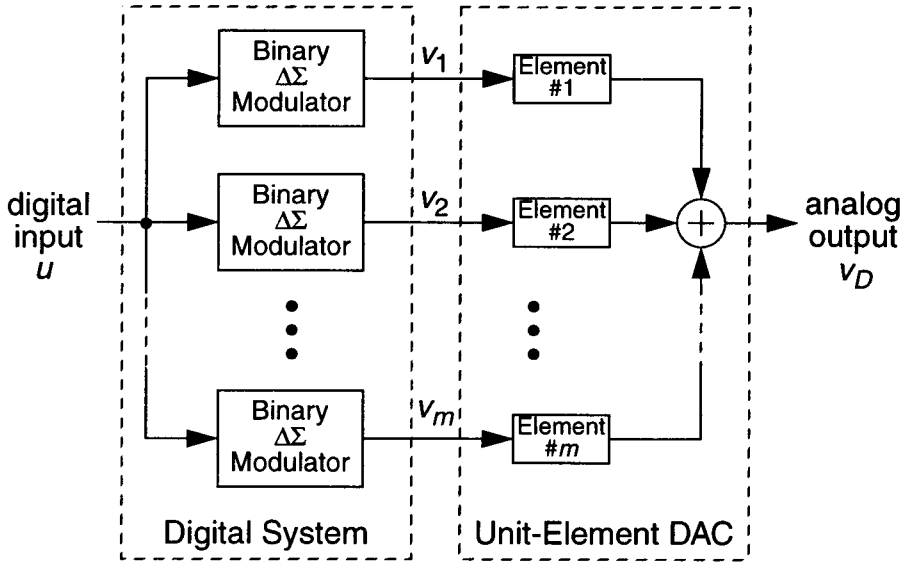


Figure 2.9: Multi-level ΔΣ DAC using parallel 1-bit modulators.

where K is the sum of the gains of the m binary DACs, and the E's are the quantization errors introduced by each modulator. This system shows that DAC errors can be rendered harmless. Other than that, the architecture does not seem to be very useful because it can hardly be called a "multi-bit" modulator. It duplicates the same branches, puts them in parallel, and then sums up their outputs. By doing so, it actually ends up with nothing but higher power consumption and larger chip area. From the frequency domain, it is more clear because all it does is add m 1-bit spectra together. By nature, it is still a single-bit modulator.

2.6.2 Element Randomization [4-6]

There are many element-matching schemes today. Although some may not use the term, they all fall under what is called "dynamic element-matching" [6] in this thesis, because they all try to attenuate element mismatch errors by using some algorithm to select among equivalent elements when constructing the analog representation of a given code. Perhaps the first dynamic matching idea was simply to randomly pick the elements to use [4]. Figure 2.10 shows a unit-element-randomization structure, namely a butterfly shuffler which is constructed with swapper cells [6, 15]. As will be discussed in later sections, the butterfly shuffler can be used to noise-shape the mismatch errors if a certain noise-shaping logic is designed to control the swappers. Random selection whitens the spectrum of the mismatch-induced error, but does not provide spectral shaping of the DAC errors.

2.6.3 Individual Level Averaging [7, 8]

In 1992, Leung proposed the individual level averaging approach [7]. Each time a certain input code appears, the unit-element DAC tries to construct the analog output using a different set of elements. After all the patterns are explored, the operation resets

and iterates. It was found that this approach not only reduced the harmonic distortion introduced by the DAC mismatch errors but also moved some of the error energy from low frequencies to high frequencies. Although the technique proved to be less effective than later schemes, it is one of the earliest demonstrations that the mismatch noise can be spectrally shaped. An important feature of a unit-element DAC, namely that the average of the analog output over all element permutations is error-free, was the key to this algorithm.

2.6.4 First-Order Mismatch-Shaping Architectures [10-15]

Two efficient mismatch-shaping schemes will be introduced in this section. They both demonstrate 1st-order lowpass mismatch-shaping.

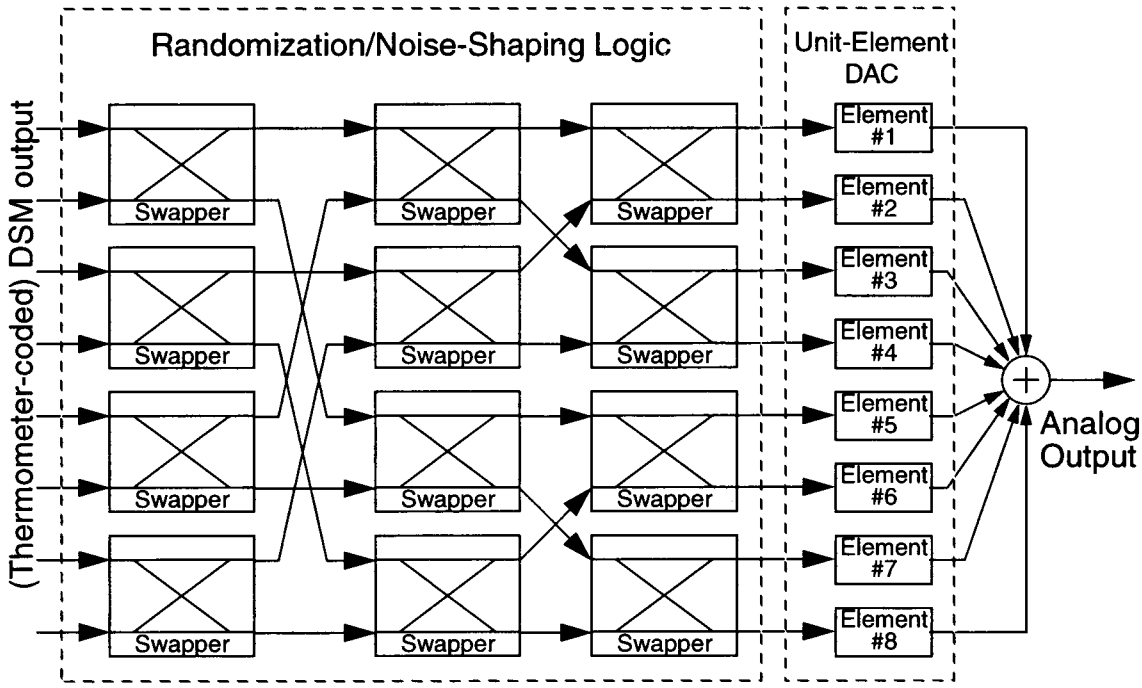


Figure 2.10:An 8-element DAC driven by a butterfly shuffler.

The first scheme is based on the butterfly shuffler introduced in Section 2.6.2. By making the swapper cells into finite-state-machines which seek to equalize the activity of their outputs, Adams and Kwan found that mismatch errors in the DAC become 1st-order-shaped [15]. A 1st-order mismatch-shaping swapper cell is shown in Figure 2.11. The truth table of the swapper is shown below. A detailed analysis will be given in Chapter 3.

Table 2.1: 1st-order mismatch-shaping swapper truth table.

x_1	x_0	CNT0-CNT1	S	New CNT0-CNT1
0	0	0	x	0
0	0	1	x	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	x	0
1	1	1	x	1

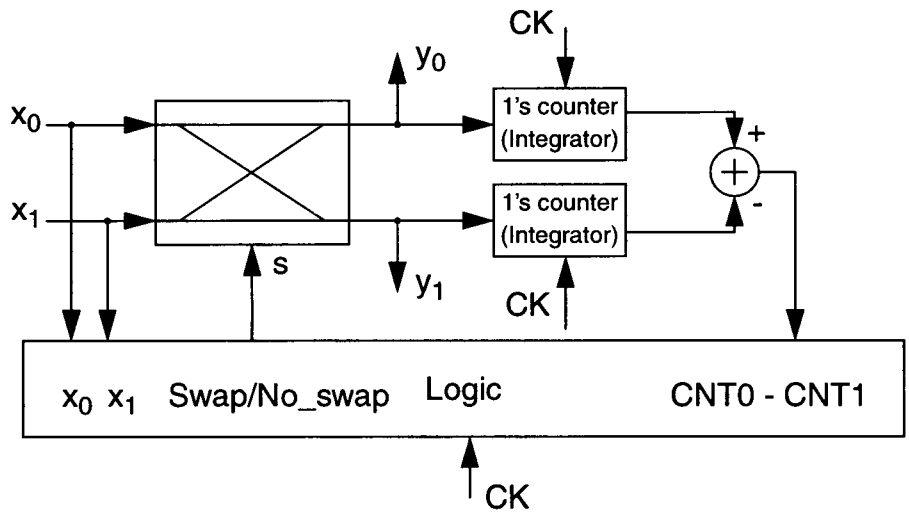


Figure 2.11:A 1st-order mismatch-shaping swapper.

Another highly practical scheme which results in 1st-order shaped mismatch errors is the element rotation scheme [10-14]. To illustrate this algorithm, an 8-element DAC example is shown in Figure 2.12. When choosing the DAC elements to construct an analog output, the algorithm starts from the "next" element instead of always starting from the very first element (say, element 1) as a conventional DAC does. For example, as shown in Figure 2.12, since it stopped at element 4 after finishing with the first input code, it starts from element 5 when the second code comes in. When the end of the element array (i.e. element 8 in the example) is reached, the algorithm wraps around to element 1. A conceptual explanation of the element rotation scheme is shown in the left part of Figure 2.12. Evidently, $V_D(z) = KV(z) + (1 - z^{-1})D(z)$, where K is the DAC gain, so DAC errors are 1st-order shaped. To see how this result emerges, imagine there is a DAC with infinite number of elements. Each time when a digital input code $v(n)$ comes in, the integrator which precedes the DAC will activate $v(n)$ more elements starting from where it stopped previously, while the differentiator will deactivate the previous elements at the same time. Merging these two operations together and stacking a real

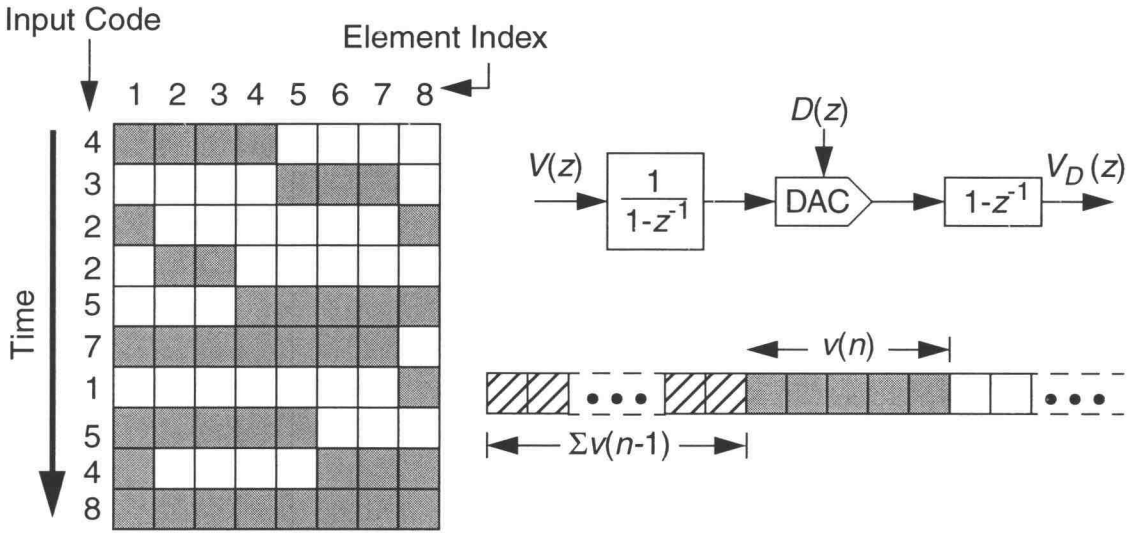


Figure 2.12:1st-order mismatch-shaping with element rotation scheme.

DAC's elements end-to-end, we then end up with the element rotation scheme. This scheme is very hardware-efficient.

Both of the schemes discussed in this section demonstrate 1st-order lowpass mismatch-shaping.

2.6.5 General Mismatch-Shaping Architectures [16, 33, 34, 35]

After 1st-order shaping was demonstrated, the question, "Are there any schemes capable of high-order or bandpass mismatch-shaping?" springs immediately to mind. The answer is "yes." Two general mismatch-shaping DAC architectures are introduced in the following paragraphs.

Figure 2.13 shows the block diagram of a general element-selection logic (ESL) [16, 33]. The bold font and the thick lines in the diagram represent m-dimensional vectors. Each element in ***sv*** is a 1-bit number (i.e. either "1" or "0"), while the word-length of the elements in ***se*** and ***sy*** are in general larger. The "vector quantizer" quantizes ***sy*** to ***sv*** in

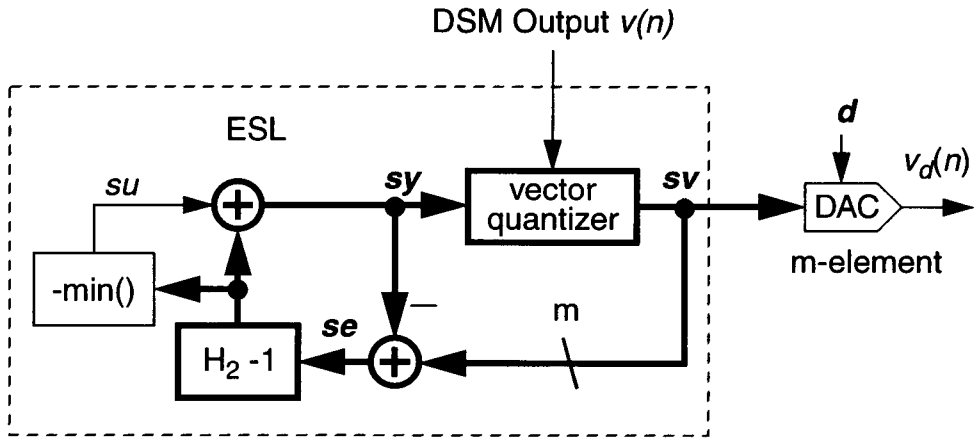


Figure 2.13:A mismatch-shaping DAC with an ESL analogous to ΔΣ modulation.

such a way that the distance between the two vectors is minimized, with the restriction that the sum of the elements in $\boldsymbol{sv}$ equals $v(n)$. The analog output of the DAC is given by

$$V_D(z) = V(z) + H(z)(\boldsymbol{SE}(z) \bullet \boldsymbol{de}). \tag{2.9}$$

where $\boldsymbol{de}$ is an m-element vector containing the difference between the actual value and the ideal value of each element. By properly choosing a mismatch transfer function (MTF) $H(z)$ which yields a bounded $\boldsymbol{se}$ sequence, higher order and bandpass mismatch-shaping can be achieved. If $H(z) = 1 - z^{-1}$, the system is equivalent to the element rotation scheme. More detailed discussions will be given in Chapter 3.

Another scheme capable of arbitrary mismatch-shaping is the tree-structure shuffler presented by Galton [34, 35]. As shown in Figure 2.14, a 2^n-element DAC can be recursively decomposed into pairs of sub-DACs until the sub-DACs are reduced to the binary case. Notice that in the binary case, this structure is equivalent to the swapper described in Section 2.6.4. At each level of recursion the signal splitters are driven by $\Delta\Sigma$ sequences in the way shown in the bottom of the diagram. Assume that the digital input signal $v_0(n)$ to the decomposed system shown in Figure 2.14 can be written as some desired signal $v(n)$ plus a (bounded) error term which is shaped by a transfer function $H(z)$, i.e. $V_0(z) = V(z) + H(z)E_0(z)$. Suppose the shaped sequence which drives the signal splitter can be described as $H(z)E_1(z)$ (the e_1 sequence is assumed to be bounded). Thus, the signals which drive the two sub-DACs are $V_{1,2}(z) = V(z)/2 + H(z)(E_0(z) \pm E_1(z))/2$ and are therefore of the same form as that supposed originally. Applying induction, we conclude that the sequences which drive the unit-elements (i.e. binary DACs) are $\Delta\Sigma$ sequences. Thus the DAC output, neglecting DC offsets, is

$$V_D(z) = (1 + \overline{D})V(z) + H(z)E_D(z), \tag{2.10}$$

where $\overline{D}$ is the average of the element errors and $E_D(z)$ is a linear combination of the $E_i(z)$ signals with weighting factors that are themselves linear combinations of the

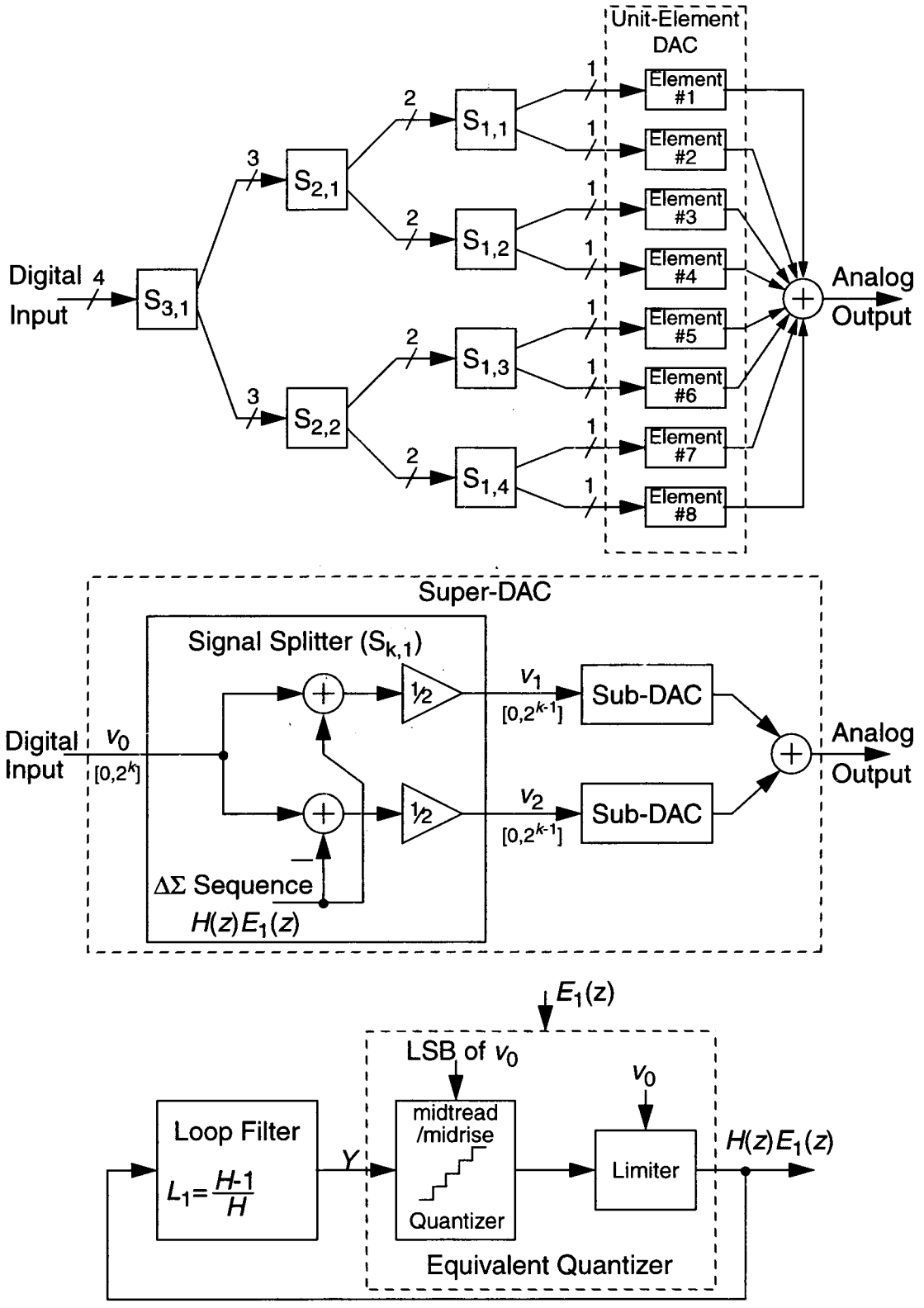


Figure 2.14:Mismatch-shaping DAC using tree-structure shuffler.

element errors. Assuming the e_D sequence is bounded, Eq. (2.10) shows that the DAC output consists of the ideal output (multiplied by a scaling factor) plus a noise-shaped error term.

Also shown in Figure 2.14 is a shaped sequence generator structure presented in [34, 35] but drawn in a more general form here. Similar to the other structure mentioned earlier in this section, a delta-sigma loop with zero input is used to generate the shaped sequence. The equivalent quantizer block in the loop imposes certain constraints which force the signal splitter to split the input signal in such a way that no overflow will occur. These constraints are the sources which introduce the selection errors E_i, and hence are the key factors in determining the loop's stability.

Both of the DAC architectures introduced in this section have demonstrated high-order and bandpass mismatch-shaping in simulation. More details will be discussed in the next chapter.

2.7 Conclusions

$\Delta\Sigma$ modulation basics and some of the existing mismatch-shaping schemes were reviewed and discussed in this chapter. By combining these techniques, high-performance multi-bit $\Delta\Sigma$ modulators can be constructed.

Chapter 3. Element-Mismatch-Shaping Architectures

This chapter takes a closer look at some mismatch-shaping architectures. Comparisons between the delta-sigma ESL and the tree-shuffler are made using simulations and analytical techniques. A design of the vector quantizer and a revised delta-sigma ESL are proposed, and a butterfly shuffler architecture capable of high-order/ bandpass mismatch-shaping is presented.

3.1 Comparison between Delta-Sigma ESL and Tree Shuffler

Simulations were done to compare the delta-sigma ESL and the tree shuffler. A 17-level modulator with a 4^{th}-order lowpass NTF optimized for $OSR = 16$ whose infinity norm is 4 is used in the comparisons. The modulator is driven by a sine wave input situated at 1/4 of the band-of-interest with a magnitude of 3/4 of full-scale. The modulator output v is shuffled by the mismatch-shaping logic and then converted to analog form by a 16-element DAC whose element values have a standard deviation of 0.5%. As shown in Figure 3.1, with an OSR of 16, the SNR is reduced from 104 dB in the ideal case to 71 dB without any dynamic matching effort, and the 2^{nd}- and 3^{rd}-order harmonics are both around 80 dB below the signal tone.

With first-order mismatch-shaping, great improvements are obtained, with both structures yielding similar performance. Figure 3.2 shows that the SNR is improved to 93 dB, and the 2^{nd}- and 3^{rd}-order harmonics are below -100 dB. The 20 dB/decade slope of the noise floor is consistent with the fact that the MTF is first-order.

Shown in Figure 3.3 are the spectra for second-order mismatch-shaping using these two competing architectures. The 2^{nd}-order MTF $H(z) = \frac{z^2 - 2z + 1}{z^2 - 1.25z + 0.5}$ that was simulated with the tree shuffler in [34, 35] is used in both schemes here. The delta-sigma

ESL yields an SNR of 92 dB, while the SNR of the tree shuffler is only 81 dB. The harmonics in both spectra are merged into the noise floor and hence invisible. The number of element permutations for the tree shuffler is much less than 2^m, whereas a delta-sigma ESL can, in principle, select any of the 2^m permutations. Therefore, the selection errors E_1 introduced by the "equivalent quantizer" in the shaped sequence generator of the tree shuffler (shown in Figure 2.14) are relatively large with high-order MTFs. On the other hand, the delta-sigma ESL is open to all the possible element selection patterns, and the selection error ***SE*** (refer to Figure 2.13 and Eq. (2.9)) is minimized by the vector quantizer which takes the entire set of elements into account simultaneously. Therefore, the delta-sigma ESL tends to be more stable than the tree shuffler because there are less selection errors introduced. More aggressive MTFs can be tolerated by the delta-sigma ESL.

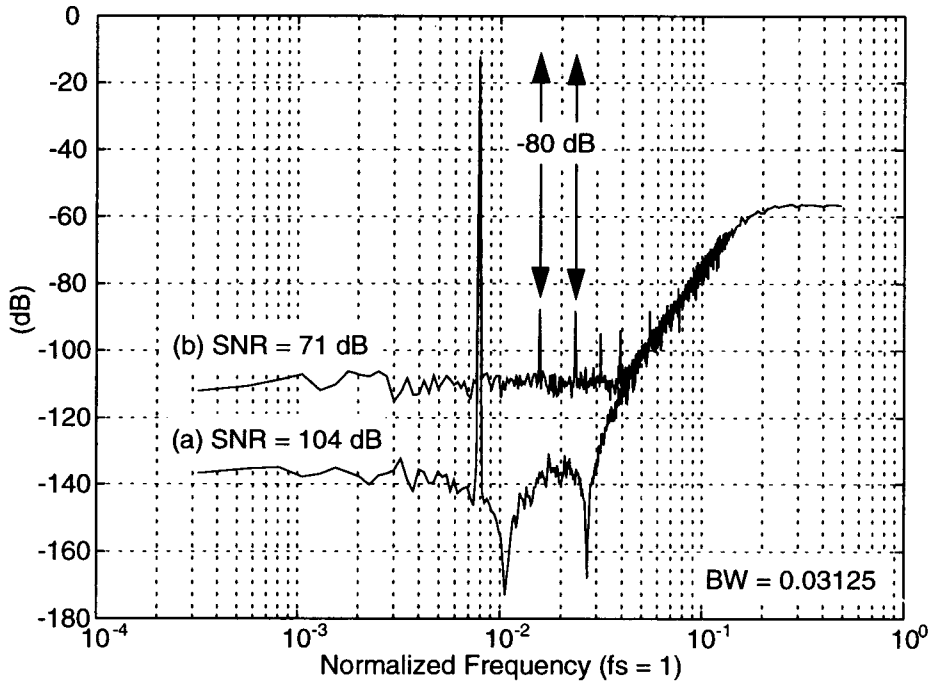


Figure 3.1: SNR and spectra of a 4th-order 17-level delta-sigma DAC with (a) an ideal DAC and (b) a 16-element DAC possessing 0.5% mismatch.

To demonstrate the theory, the delta-sigma ESL is simulated with an MTF optimized at $OSR = 16$ whose infinity norm is 1.8, and the SNR improves to 97 dB, outperforming the first-order mismatch-shaping schemes even at this low OSR, while the tree shuffler is not stable with this optimized MTF and does not demonstrate any mismatch-shaping.

In terms of implementation complexity, the delta-sigma ESL is very hardware-efficient in the first-order case because it can be reduced to the element rotation scheme. For higher-order mismatch-shaping, both the swapper and delta-sigma ESL architectures are very complicated because multiple delta-sigma loop filters are needed to implement ESLs and "signal splitters." Although the number of these delta-sigma loops are nearly the same in both cases (m parallel loops in a delta-sigma ESL and (m-1) in the tree shuffler

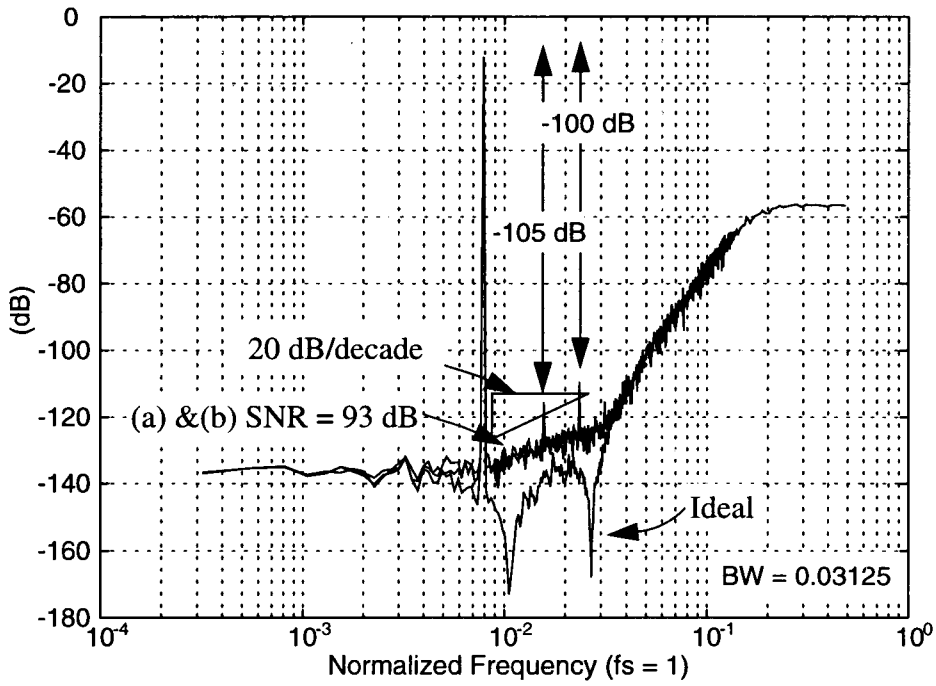


Figure 3.2: SNR and spectra of a 4th-order 17-level delta-sigma DAC with 1st-order mismatch-shaping using (a) tree-structure shuffler, (b) delta-sigma-like ESL.

to drive an m-element DAC), the loop filters in the early stages of a tree shuffler tend to require longer word-lengths. Also, extra logic is needed to implement the "equivalent quantizers" in a tree shuffler. However, it is hard to claim that high order delta-sigma ESLs are any simpler because complicated sorting circuits are needed to realize the vector quantizer. The hardware costs for the vector quantizer can be reduced by employing incomplete sorting algorithms [37, 38]. This, however will increase selection errors and degrade performance somewhat. Another minor advantage of the delta-sigma ESL is that it can drive any number of elements, while the tree structure requires the number of DAC elements to be a power of 2.

The discussion in this section concludes that, by producing less selection errors, delta-sigma ESLs tolerate more aggressive MTFs and can provide better performance in

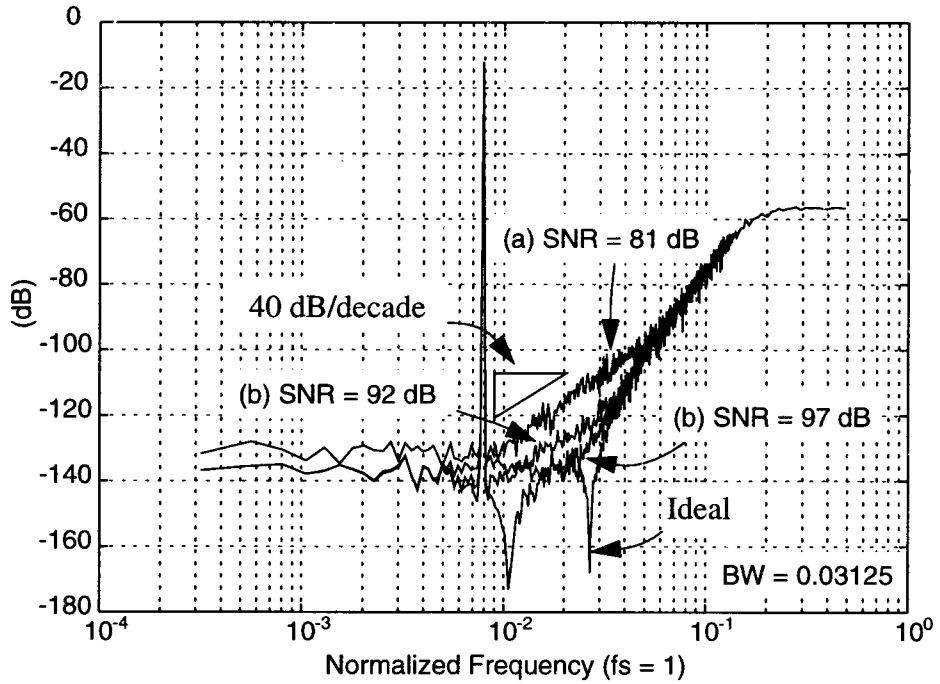


Figure 3.3: SNR and spectra of a 4^{th}-order 17-level delta-sigma DAC with 2^{nd}-order mismatch-shaping using (a) tree-structure shuffler, (b) delta-sigma-like ESL and (c) optimized delta-sigma-like ESL.

high order mismatch-shaping than tree shufflers. However, both architectures are very complicated when high order MTFs are used.

3.2 Delta-Sigma ESL

The delta-sigma ESL architecture is discussed in the following sections. A design of the vector quantizer and a revised delta-sigma ESL are also presented.

3.2.1 A Study on the Delta-Sigma ESL

For easy reference, the system diagram of a mismatch-shaping DAC with a delta-sigma ESL previously shown in Figure 2.13 is re-drawn in Figure 3.4. As shown in the diagram, ***de*** is an m-element vector whose components are the element errors de_i. Since the element errors are defined as the difference between the actual element values and the average of all element values, the sum of the element errors is zero:

$$\sum_{i=1}^{m} de_i = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \bullet \boldsymbol{de} = 0. \tag{3.1}$$

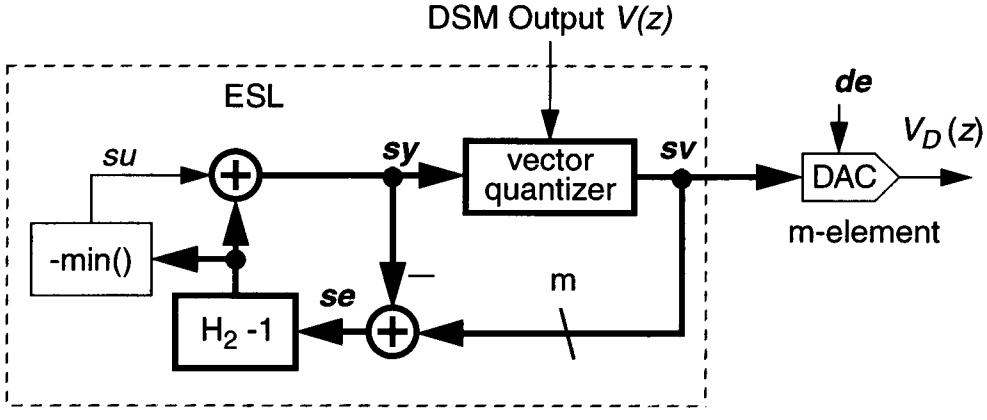


Figure 3.4: A mismatch-shaping DAC with an ESL analogous to ΔΣ modulation.

Since the sum of all the elements in the selection vector $\boldsymbol{sv}$ is equal to the value of the DAC input v, the analog output of the DAC can be described as:

$$V_D(z) = \boldsymbol{SV}(z) \bullet \left(\left[A \; A \; \ldots \; A \right]' + \boldsymbol{de} \right) = AV(z) + \boldsymbol{SV}(z) \bullet \boldsymbol{de} \qquad (3.2)$$

where A is the DAC gain (i.e. the average of all element values). The first term in Eq. (3.2) represents the signal and the shaped quantization errors, while the second term represents the DAC mismatch error, which is the focus of the discussion here.

By noticing that the ESL in Figure 3.4 is simply a vector version of the error feedback structure of a delta-sigma modulator [2], we immediately write that:

$$\boldsymbol{SV}(z) = SU(z) \left[1 \; \ldots \; 1 \right] + H_2(z) \boldsymbol{SE}(z) . \qquad (3.3)$$

Therefore, the DAC error at the output can be written as:

$$E_D(z) = \boldsymbol{SV}(z) \bullet \boldsymbol{de} = SU(z) \left[1 \; \ldots \; 1 \right] \bullet \boldsymbol{de} + H_2(z) \boldsymbol{SE}(z) \bullet \boldsymbol{de} . \qquad (3.4)$$

Substituting Eq. (3.1) into Eq. (3.4), we have:

$$E_D(z) = H_2(z) \boldsymbol{SE}(z) \bullet \boldsymbol{de} , \qquad (3.5)$$

where $H_2(z)$ is known, and $\boldsymbol{de}$ is an unknown vector with constant components for a given DAC. However, an analytical representation of $\boldsymbol{se}$ is somewhat hard to derive. A spectrum of one of the $\boldsymbol{se}$ components obtained from the simulation of the best performing system described in Section 3.1 (i.e. 4^{th}-order modulator with 2^{nd}-order optimized delta-sigma ESL) is shown in Figure 3.5. The spectra of the other $\boldsymbol{se}$ components are almost the same. Except for the signal tone and the 2^{nd}-order harmonic, the rest of the spectrum is nearly white. The tone at the signal frequency usually does no harm to the DAC (it gives rise to a gain error), whereas the 2^{nd}-order harmonic must be dealt with separately. Taking these two bins out of the spectrum, we derive a new vector $\boldsymbol{se}_N$ whose components can be treated as white noise which are the causes of the mismatch noise

described by Eq. (3.5). Assuming that the components of $\boldsymbol{se}_N$ are uncorrelated and white with rms values of σ_{se}, the corresponding noise spectral density is:

$$S_{DN}(\omega) = \frac{|H_2(e^{j\omega})|^2\sigma_{se}^2}{\pi}\sum_{i=1}^{m} de_i^{\,2} = \frac{|H_2(e^{j\omega})|^2\sigma_{se}^2 m\sigma_{de}^2}{\pi}, \tag{3.6}$$

where σ_{de}^2 is the mean-square value of the m element errors. The in-band noise power at the DAC output can then be written as:

$$\sigma_{dv}^2 = \frac{1}{OSR}[(\sigma_1\sigma_e)^2 + m(\sigma_2\sigma_{se}\sigma_{de})^2] , \tag{3.7}$$

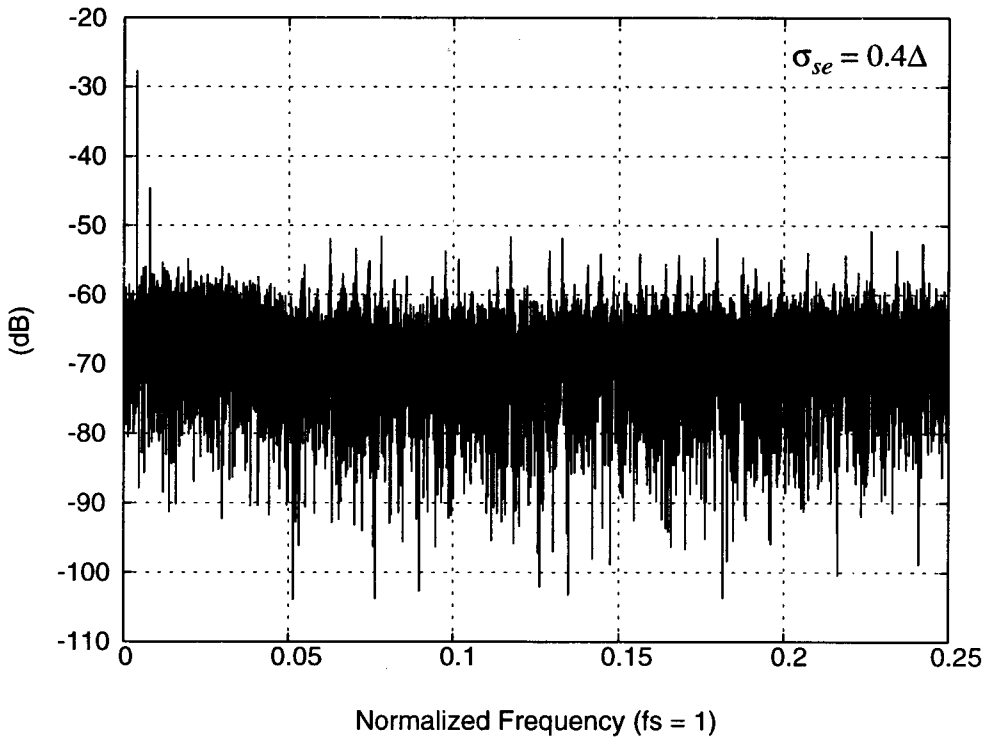


Figure 3.5: Spectrum of one of the ***se*** components.

where σ_1 and σ_2 are the rms values of H_1 (NTF) and H_2 (MTF) in the band of interest, respectively. Due to difficulties in calculating σ_{se_N} analytically, simulation is the main approach to estimate the SNR and to verify the stability of the system. Simulations with second-order MTFs appear to yield bounded state variables, while it is likewise easy to show that with higher order algorithms, the element selection logic is unstable.

Eq. (3.7) shows that the resolution of the DAC can be improved by reducing σ_{se}, the rms value of the selection error, as well as by the more obvious methods of reducing the element mismatch and optimizing the MTF. Enabling the DAC elements which correspond to the $v(n)$ largest components of $\boldsymbol{sy}(n)$ minimizes the norm of the $\boldsymbol{se}(n)$ vector, and thus helps to keep signals finite. Since the priorities of the elements are obtained by sorting the components of $\boldsymbol{sy}(n)$, no difference will be made by alternative choices for su as long as it is a scalar. Setting $su(n)$ to the negative of the minimum value of the output of the $H_2(z)$-1 block removes commonality in the $\boldsymbol{sy}(n)$ vector and helps to keep the components of $\boldsymbol{sy}$ finite and positive. According to Eq. (3.4), it is unlikely that E_D can be further reduced by generating a vector $\boldsymbol{su}(n)$ instead of using a scalar $su(n)$ because it will make the first term non-zero, unless doing so can somehow cause the $\boldsymbol{se}(n)$ sequence to have a smaller σ_{se}. Vector quantizers with other algorithms producing smaller σ_{se} might also exist.

3.2.2 A Design of a Vector Quantizer

All blocks in the delta-sigma ESL shown in Figure 3.4 can be implemented with simple digital logic and normal digital filter structures, except for the vector quantizer. In the software version, the vector quantizer completely sorts the components of $\boldsymbol{sy}(n)$ in order to choose which elements to enable. Performing this operation in hardware requires complicated circuits, one of which is presented in this section.

A vector quantizer capable of sorting a 16-component $\boldsymbol{sy}(n)$ vector within a single clock cycle is shown in Figure 3.6. Each $\boldsymbol{sy}(n)$ component is assumed to be 3 bits wide. According to simulations, 3 bits should be enough to prevent overflow. Also notice that, since $su(n)$ is set to the negative of the minimum value of the output of the $H_2(z)$-1 block, all the $\boldsymbol{sy}(n)$ components are positive at the input of the vector quantizer. The output of the quantizer is the $\boldsymbol{sv}(n)$ vector containing 16 single-bit components that are used to drive a 16-element DAC which is not shown.

The first stage in the quantizer is a layer of 16 1-of-8 decoders. Each takes a 3-bit component of $\boldsymbol{sy}(n)$ and translates it to an 8-bit word with only one of the bits equal to "1" based on the truth table shown below. Each component of $\boldsymbol{sy}$ is assigned to one of 8

Table 3.1:1-of-8 decoder truth table.

sy_i	$sy_{i,8\sim1}$
000	10000000
001	01000000
010	00100000
011	00010000
100	00001000
101	00000100
110	00000010
111	00000001

priorities with $sy_{i,1} = 1$ standing for the first priority and $sy_{i,8} = 1$ standing for the last priority.

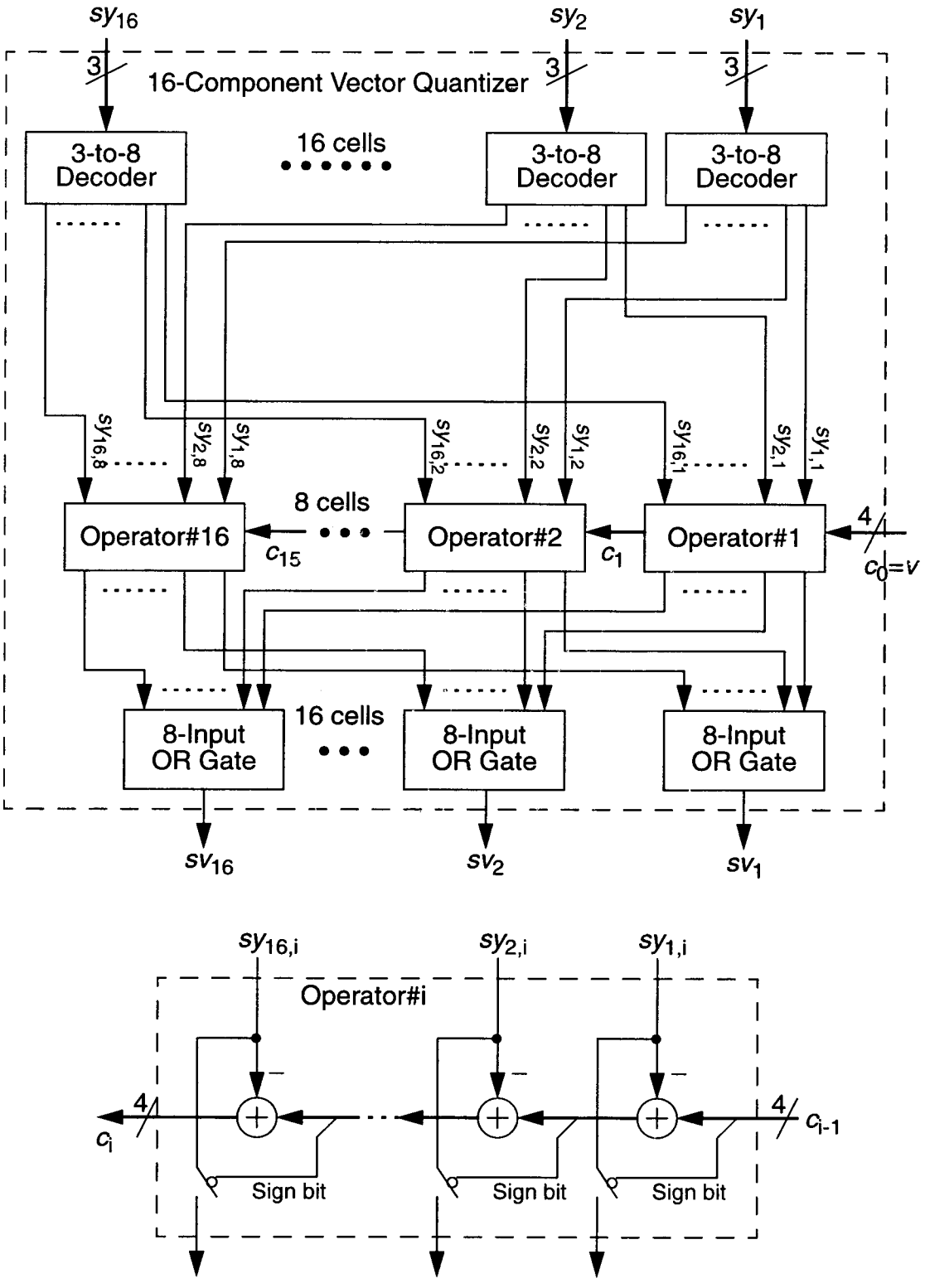


Figure 3.6: A 16-component vector quantizer.

The second stage is a layer of 8 "operator" boxes. Each boxes handles a certain priority level. The 16 "first priority bits" from the output of the decoder stage are wired to the right-most box, "operator #1", and the "second priority bits" are connected to "operator #2", etc. The structure of one of the "operators" is shown at the bottom of Figure 3.6. The name "operator" was chosen because each box contains a set of switches and makes connections between lines in a manner similar to what a telephone operator does. Carried in from the right side of the ith box is a variable c_i with the same word-length as that of the delta-sigma modulator output (namely, the DAC input) v. For "operator #1", $c_i = v$. This variable ripples through the box from right to left. Each time a "priority" bit whose value has been set to "1" is switched to the next level by the box, the variable decrements by 1 until the count hits zero. A "priority" bit will be switched to the next level only when the corresponding carry variable is positive.

Several simplifications to the "operators" are possible. First, if the output bits of the "operators" are tied to "0" by default, then switching "priority" bits with zero values is no longer necessary. Therefore, simple control logic can be add to each cell inside an "operator" box to bypass the cell whenever the "priority" bit is zero so that the ripple can be accelerated. Also, the carry to each of the "operators" can be pre-calculated so that the waiting time among the "operators" is reduced. Secondly, the "operator" chain can be broken to smaller segments and mapping tables can be used to substitute the adders and, at the same time, to integrate the ideas above mentioned, so that both the speed and the hardware efficiency can be improved.

The last layer in the proposed vector quantizer are 16 8-input OR gates. Each gate drives a DAC element. If one of the input signals to an OR gate is high, the corresponding DAC element will be turned on.

The hierarchical structure of the vector quantizer makes it perfect for VLSI implementation. Unfortunately, it is also complicated and slow. The longest path contains $(2^{k+n}+2)$ gate delays, where k is the word-length of the ***sy*** components and n is the word-length of the DAC input v. Great simplification can be done by reducing the word-length of the input ***sy*** components. Every reduction of 1 bit cuts the hardware by 50%. This, however, might result in incomplete sorting and introduce additional errors that must be counted as part of the ***se*** errors.

3.2.3 A Revised Delta-Sigma ESL

As mention earlier, the output ***sv*** vector generated by a delta-sigma ESL such as that shown in Figure 3.4 contains m 1-bit components. Each component is used to drive an individual unit DAC element. When ***sy***(n) is quantized to ***sv***(n), no matter how big a ***sy*** component may be, the best the vector quantizer can do is to quantize it to "1" because there is only one corresponding component can be activated. Should each ***sv*** component control more than one element, or if each element can be turned on more than once within one clock cycle, multi-level quantization of ***sy*** would then be possible and the selection error ***se*** could be reduced.

One example DAC which tries this idea is shown in Figure 3.7. The elements in the m-element DAC are grouped in pairs with each pair driven by a ***sv*** component. The ESL that generates ***sv*** has the same structure as that shown in Figure 3.4, except that the sizes of all the vectors are reduced by half. ***sv*** now has three values: "0", "1" and "2". When choosing the elements inside a pair, the element rotation strategy (which gives 1st-order mismatch-shaping inside the pair) is used. Simulations with the same DAC used in Section 3.1 show that the SNR with a 1st-order MTF is 84 dB, and 83 dB with the optimized 2nd-order MTF. The improvements are not as big as those of the original delta-

sigma ESLs. The reason is that, inside each pair, the two elements are not perfectly matched.

To fight against the mismatch inside each element sub-group, careful layout is needed to match the elements inside a group. However, less effort is needed to match the whole element array because the mismatch errors among the sub-groups are taken care of by the ESL, and matching on the layout of a small number of local cells is relatively easy. Instead of dividing elements into small groups, another way to overcome this "internal" error term is to use sub-clocks, i.e. to divide the main clock cycle to multiple sub-periods, so that each element can be turned on more than one time within a main clock cycle. The trade-offs are more delicate clock circuits and a slower maximum sampling frequency. To prove these ideas, the elements in the DAC previously used are again grouped in pairs, then the average value of each pair is calculated and re-assigned to the both element inside the pair. With the optimized 2^{nd}-order MTF, an SNR of 96 dB is achieved. Some

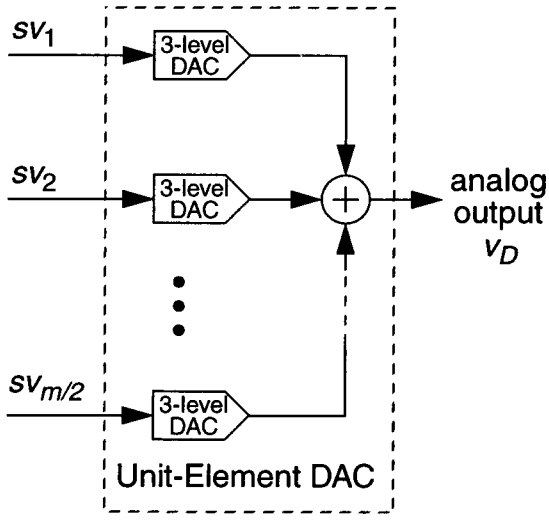


Figure 3.7: Unit-element DAC with the elements grouped in pairs.

simulation results are shown in Figure 3.8. The harmonics are part of the unshaped curve. Both mismatch-shaping schemes shown in the picture produce no visible harmonics.

This revised delta-sigma ESL also reduces the number of delta-sigma loops and the size of the vector quantizer by half if *sv* controls 3-level DAC arrays, and can save more if *sv* drives sub-DACs with even more levels. However, the control logic inside the vector quantizer gets more complicated with a increased number of sub-DAC levels.

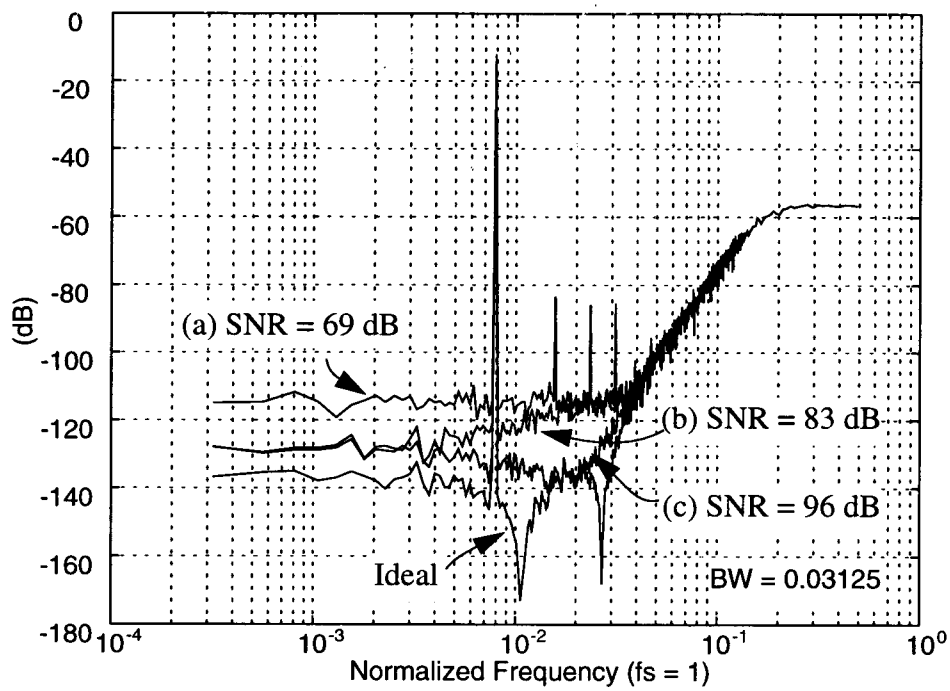


Figure 3.8: SNR and spectra of a 4th-order 17-level delta-sigma DAC using (a) no mismatch-shaping with internally matched element pairs, (b) revised 2nd-order optimized delta-sigma ESL driving internally mismatched element pairs and (c) revised 2nd-order optimized delta-sigma ESL driving internally matched element pairs.

3.3 Butterfly Shuffler

In the following sections, a new scheme capable of "arbitrary" mismatch shaping based on the existing butterfly shufflers is presented. The modeling and analysis method combines partly the features of Adams' [15], Schreier's [16], [33] and Galton's [34], [35] work. A 2^{nd}-order bandpass noise-shaping swapper is given as an example along with a 1^{st}-order lowpass example which has been proven to be equivalent to Adams and Kwan's scheme. Comparisons with the delta-sigma ESL architecture are given last.

3.3.1 Modeling a Swapper

Figure 3.9 shows a swapper cell whose functionality is defined as:

$$\begin{aligned} y_0 &= x_0 \text{ and } y_1 = x_1 \text{ if } S = 1, \\ y_0 &= x_1 \text{ and } y_1 = x_0 \text{ if } S = -1. \end{aligned} \tag{3.8}$$

Eq. (3.8) can be written in a mathematical form:

$$\begin{aligned} y_0 &= \frac{x_0 + x_1}{2} + S\frac{x_0 - x_1}{2} \\ y_1 &= \frac{x_0 + x_1}{2} - S\frac{x_0 - x_1}{2} \end{aligned} \tag{3.9}$$

From Eq. (3.9), it is easy to derive that:

$$(y_0 - y_1) = S(x_0 - x_1). \tag{3.10}$$

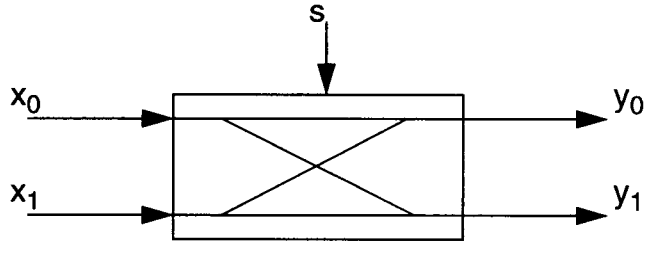


Figure 3.9: A swapper cell.

A swapper cell is a single stage butterfly shuffler which can be used to drive a 2-element (1.5-bit) DAC. Assume the DAC elements deviate from the mean value A of the two elements by e_0 and e_1, respectively. (Note that by this definition, $e_0 + e_1 = 0$.) The DAC output can then be written as:

$$a_{out} = A(y_0 + y_1) + e_0 y_0 + e_1 y_1 . \tag{3.11}$$

Substituting Eq. (3.9) into Eq. (3.11), we have:

$$a_{out} = A(x_0 + x_1) + S(x_0 - x_1)\frac{e_0 - e_1}{2} . \tag{3.12}$$

Eq. (3.12) shows that the mismatch in the DAC elements introduces an error term $S(x_0 - x_1)\frac{e_0 - e_1}{2}$, where $\frac{e_0 - e_1}{2}$ is constant for a given DAC.

According to Eq. (3.10), if $(y_0 - y_1)$ is a n^{th}-order shaped sequence, the error term of Eq. (3.12) will be n^{th}-order shaped, and the DAC output can be written in the form:

$$a_{out} = (Gain \times Input\_Signal) + (n^{th}\text{-}Order\_Shaped\_Sequence) .$$

The conclusion is that, if the two output signals of a swapper are such that their difference is a n^{th}-order shaped sequence, then the error caused by the element mismatch of a 2-element DAC which is driven by the swapper is n^{th}-order shaped. In this sense, the 3-level butterfly shuffler is very similar to tree shuffler discussed earlier.

3.3.2 Modeling a Butterfly Shuffler

A 2-stage butterfly shuffler is illustrated in Figure 3.10 as an example for modeling the butterfly shuffler structure based on the swapper model presented above.

The shuffler in Figure 3.10 has two layers, Layer 0 and Layer 1. The first subscript in the notation denotes the layer number. Using Eq. (3.9), it is easy to write

$$
\begin{aligned}
x_{1,0} &= \frac{x_{0,0}+x_{0,1}}{2} + S_{0,0}\frac{x_{0,0}-x_{0,1}}{2} \\
x_{1,1} &= \frac{x_{0,0}+x_{0,1}}{2} - S_{0,0}\frac{x_{0,0}-x_{0,1}}{2} \\
x_{1,2} &= \frac{x_{0,2}+x_{0,3}}{2} + S_{0,1}\frac{x_{0,2}-x_{0,3}}{2} \\
x_{1,3} &= \frac{x_{0,2}+x_{0,3}}{2} - S_{0,1}\frac{x_{0,2}-x_{0,3}}{2}
\end{aligned}
\quad , \tag{3.13}
$$

$$
\begin{aligned}
y_0 &= \frac{x_{1,0}+x_{1,2}}{2} + S_{1,0}\frac{x_{1,0}-x_{1,2}}{2} \\
y_1 &= \frac{x_{1,0}+x_{1,2}}{2} - S_{1,0}\frac{x_{1,0}-x_{1,2}}{2} \\
y_2 &= \frac{x_{1,1}+x_{1,3}}{2} + S_{1,1}\frac{x_{1,1}-x_{1,3}}{2} \\
y_3 &= \frac{x_{1,1}+x_{1,3}}{2} - S_{1,1}\frac{x_{1,1}-x_{1,3}}{2}
\end{aligned}
\quad . \tag{3.14}
$$

Eq. (3.14) can be further written as

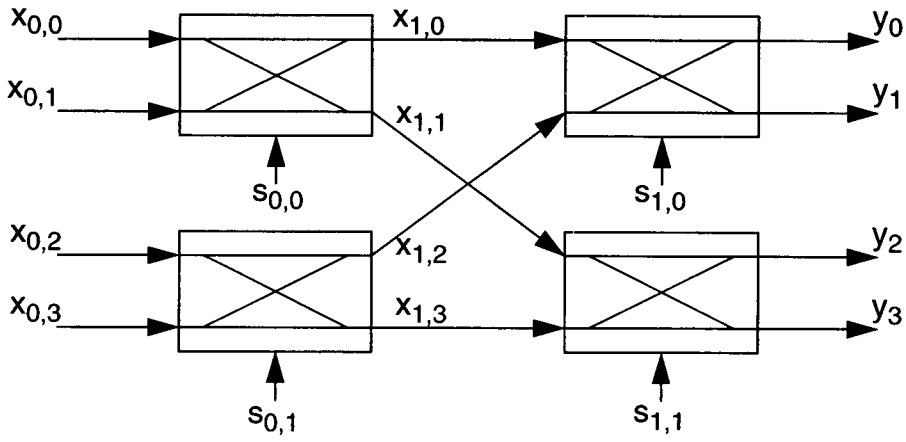


Figure 3.10: A 2-stage Butterfly Scrambler for driving a 4-element DAC.

$$
\begin{aligned}
y_0 &= \frac{v}{4} + S_{0,0}\frac{x_{0,0}-x_{0,1}}{4} + S_{0,1}\frac{x_{0,2}-x_{0,3}}{4} + S_{1,0}\frac{x_{1,0}-x_{1,2}}{2} \\
y_1 &= \frac{v}{4} + S_{0,0}\frac{x_{0,0}-x_{0,1}}{4} + S_{0,1}\frac{x_{0,2}-x_{0,3}}{4} - S_{1,0}\frac{x_{1,0}-x_{1,2}}{2} \\
y_2 &= \frac{v}{4} - S_{0,0}\frac{x_{0,0}-x_{0,1}}{4} - S_{0,1}\frac{x_{0,2}-x_{0,3}}{4} + S_{1,1}\frac{x_{1,1}-x_{1,3}}{2} \\
y_3 &= \frac{v}{4} - S_{0,0}\frac{x_{0,0}-x_{0,1}}{4} - S_{0,1}\frac{x_{0,2}-x_{0,3}}{4} - S_{1,1}\frac{x_{1,1}-x_{1,3}}{2}
\end{aligned}
\quad , \qquad (3.15)
$$

where v is the DAC input signal which comes from a delta-sigma modulator. From the discussion in the last section, if the difference of the outputs of each swapper is a shaped sequence, then the last three terms of each equation in Eq. (3.15) are all shaped sequences. If y_0, y_1, y_2, y_3 are used to drive a 4-element DAC with element mismatch errors e_0, e_1, e_2 and e_3 $(e_0 + e_1 + e_2 + e_3 = 0)$, simple algebra similar to that of the single swapper shows that the mismatch error at the DAC output is:

$$
\begin{aligned}
e_d = &\left(S_{0,0}\frac{x_{0,0}-x_{0,1}}{4} + S_{0,1}\frac{x_{0,2}-x_{0,3}}{4}\right)(e_0 + e_1 - e_2 - e_3) \\
&+ \left(S_{1,0}\frac{x_{1,0}-x_{1,2}}{4}\right)(e_0 - e_1) + \left(S_{1,1}\frac{x_{1,1}-x_{1,3}}{4}\right)(e_2 - e_3)
\end{aligned}
\quad . \qquad (3.16)
$$

Therefore, as long as the difference of the two output signals of each swapper is a shaped sequence, the spectrum of e_d will have the same shaping.

A similar analysis can be easily carried out on an m-stage butterfly shuffler to show that the mismatch error term at the output of a 2^m-element DAC which is driven by the shuffler is a linear combination of the differences of all the swapper outputs with weighting factors that are themselves linear combinations of the element errors. A general conclusion can be made that, as long as each swapper in the shuffler is controlled in such a way that the spectrum of its output difference is shaped by a MTF, the errors introduced by the static element mismatch in a 2^m-element DAC which is driven by the shuffler will be shaped by the same MTF.

3.3.3 Mismatch-Shaping Swappers and Butterfly Shufflers

After the modeling and discussion of the previous sections, the question to be answered is: how does one shape the difference between the two outputs of a swapper with a desired MTF? A mismatch-shaping swapper using a process similar to delta-sigma modulation provides the answer.

Figure 3.11 illustrates the structure of a mismatch-shaping swapper. At first glance, it is quite different from the swapper cells drawn before. But if it is viewed as a black box from the outside, it behaves exactly like any other swapper, i.e. either passes x_0 to y_0 and x_1 to y_1, or swaps x_0 to y_1 and x_1 to y_0. Inside the box, it is very similar to a delta-sigma modulator with the input signal set to zero. The only difference is that a logic block is used to substitute the normal quantizer in a modulator loop. The logic block takes two 1-bit input signals, x_0 and x_1, and produces two 1-bit outputs, y_0 and y_1, in such a way that the difference signal $sv = y_0 - y_1$ is the best approximation to sy given the restriction

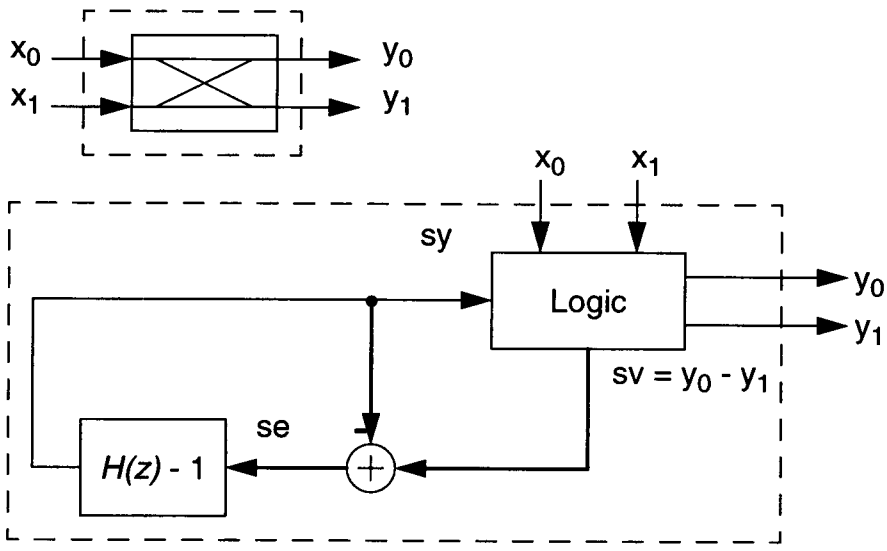


Figure 3.11:A noise-shaping swapper.

that $y_0 + y_1 = x_0 + x_1$. As was the case with the vector quantizer, the logic can be viewed as a quantizer which quantizes *sy* to *sv* introducing a "quantization" error *se*. This quantization error is then manipulated by a digital filter with the transfer function $H(z) - 1$ to generate the next *sy*, where $H(z)$ is the desired MTF. By again noting that this is the "error feedback" structure of a delta-sigma modulator, we immediately write the transfer function:

$$SV(z) = SE(z)H(z). \tag{3.17}$$

Since the input to the modulator is zero, there is no signal term in the expression for *sv*. Assuming *se* is bounded, then *sv* (i.e. $y_0 - y_1$) is a spectrally shaped sequence.

3.3.4 1st-Order Lowpass Swapper

For the 1st-order MTF $H(z) = 1 - z^{-1}$, the number of logic states is so limited that they can be mapped into a simple truth table and implemented with simple digital logic.

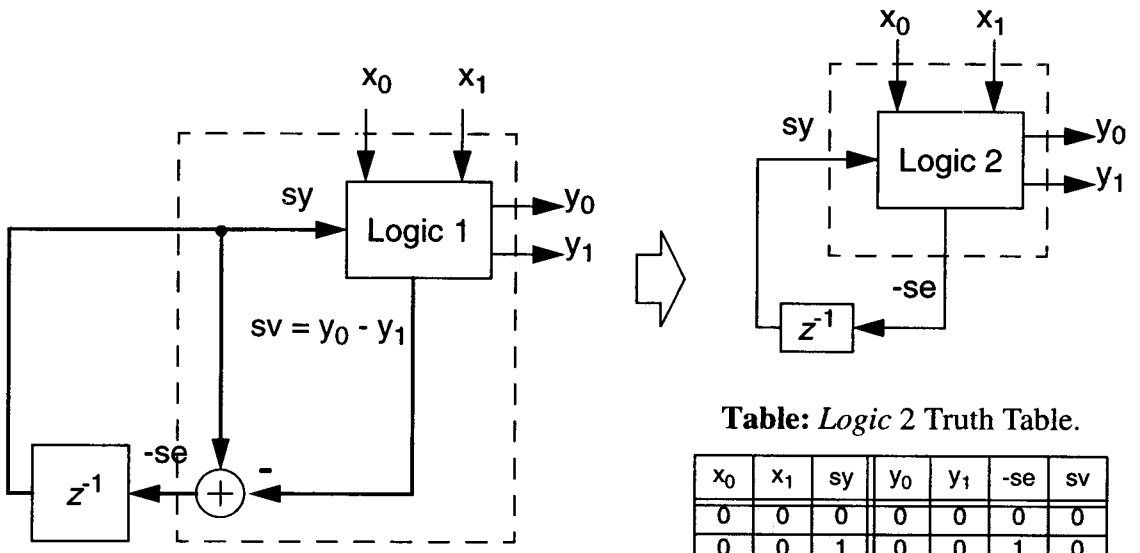
The system diagram and the "quantization" logic truth table is shown in Figure 3.12. Note that the "quantization" logic (*Logic* 1) and the adder are all merged into *Logic* 2. All the variables are single-bit, except *sv* which no longer exists in the real circuit. The system is therefore very simple to implement.

Comparisons between the *Logic* 2 truth table and Table 2.1 which describes the swapper operations of the 1st-order mismatch-shaping butterfly shuffler presented by Adams & Kwan show that the two are exactly the same. Therefore, the idea works in the 1st-order case.

If combined with the special connection of Robertson's butterfly shuffler structure [39] so that each individual swapper always sees thermometer-coded input $[x_0, x_1]$, the swapping logic can be further simplified. The possible states that can occur are shaded in the *Logic* 2 truth table. Among all these states, only the one that is heavily shaded corresponds to a swapping operation; others are non-swapping cases.

3.3.5 Bandpass Swapper

A bandpass Butterfly Shuffler was designed based on the model of the mismatch-shaping swapper presented above. The center frequency is designed at $\frac{\pi}{2}$ (i.e. 1/4 of the sampling frequency). A simple MTF $H_2(z) = (1+z^{-2})$ is used for designing the shuffler and deriving the truth table.



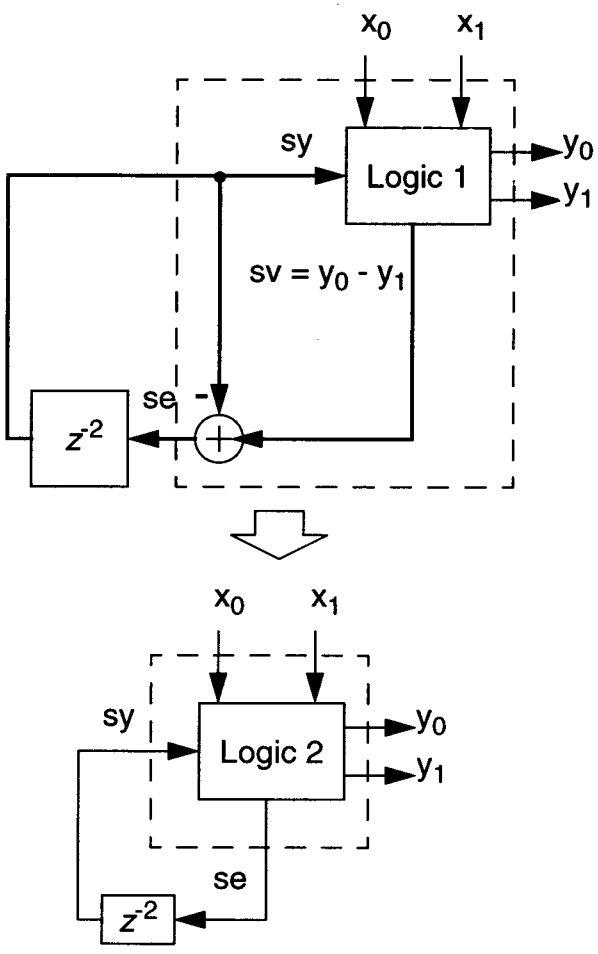
$$H(z) = (1-z^{-1})$$

Table: *Logic* 2 Truth Table.

x_0	x_1	sy	y_0	y_1	-se	sv
0	0	0	0	0	0	0
0	0	1	0	0	1	0
0	1	0	0	1	1	-1
0	1	1	1	0	0	1
1	0	0	0	1	1	-1
1	0	1	1	0	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

Figure 3.12:A 1st-order lowpass noise-shaping swapper.

Figure 3.13 illustrates the 2nd-order bandpass mismatch-shaping swapper along with its truth table. Here, *sy* has three states, and therefore is a 2-bit word. Hence, the truth table is a little larger than that in the single-bit 1st-order lowpass case. Also notice that a double-delay block is needed in the bandpass case. Again, this bandpass swapper can be combined with Robertson's thermometer code shuffler to simplify the logic. The corresponding cases are shaded in the table. As with the lowpass example, the swapper only swaps under one particular condition which is heavily shaded in the truth table.



$$H_2(z) = 1 + z^{-2}$$

Table: *Logic* 2 Truth Table.

x_0	x_1	sy	y_0	y_1	se	sv
0	0	0	0	0	0	0
0	0	1	0	0	-1	0
0	1	0	0	1	-1	-1
0	1	1	1	0	0	1
1	0	0	1	0	1	1
1	0	1	1	0	0	1
1	1	0	1	1	0	0
1	1	1	1	1	-1	0
0	0	-1	0	0	1	0
0	1	-1	0	1	0	-1
1	0	-1	0	1	0	-1
1	1	-1	1	1	1	0

Figure 3.13:A bandpass noise-shaping swapper.

A delta-sigma DAC system which employs the lowpass and bandpass butterfly shufflers will be introduced as a design example in the next chapter. More simulation and performance details can be found there.

3.3.6 Comparisons between the Mismatch-Shaping Butterfly Shuffler and the ΔΣ ESL

Performance comparisons are made between the butterfly shuffler and the delta-sigma ESL using simulations similar to those given Section 3.1. The input v is generated by the same 4^{th}-order modulator with $OSR = 16$ as the one in Section 3.1, and the shufflers' outputs are used to drive the same 16-element DAC.

With the 1^{st}-order MTF, the SNR is improved from 71 dB to 87 dB with the butterfly shuffler. This number, however, is still about 6 dB below what the delta-sigma ESL and tree shuffler can do. However, the harmonics produced by the butterfly shuffler are merged into the noise floor and lower than that of the other two schemes. This is due to the fact that the swappers produce more varied element selection patterns (selected elements need not be adjacent) and so are less likely to produce periodic selection patterns. Some of the spectra from the simulations are shown in Figure 3.14. Note that the harmonic peaks are not part of the spectrum produced by the butterfly-driven DAC; they are from the curve underneath which corresponds to 1^{st}-order mismatch-shaping with delta-sigma ESL.

Using the mismatch-shaping swapper architecture in Figure 3.11, butterfly shufflers with higher order MTFs can be built. However, state variables in higher-order systems can get large, and more complicated designs are inevitable. Stability is another limiting factor in high-order systems.

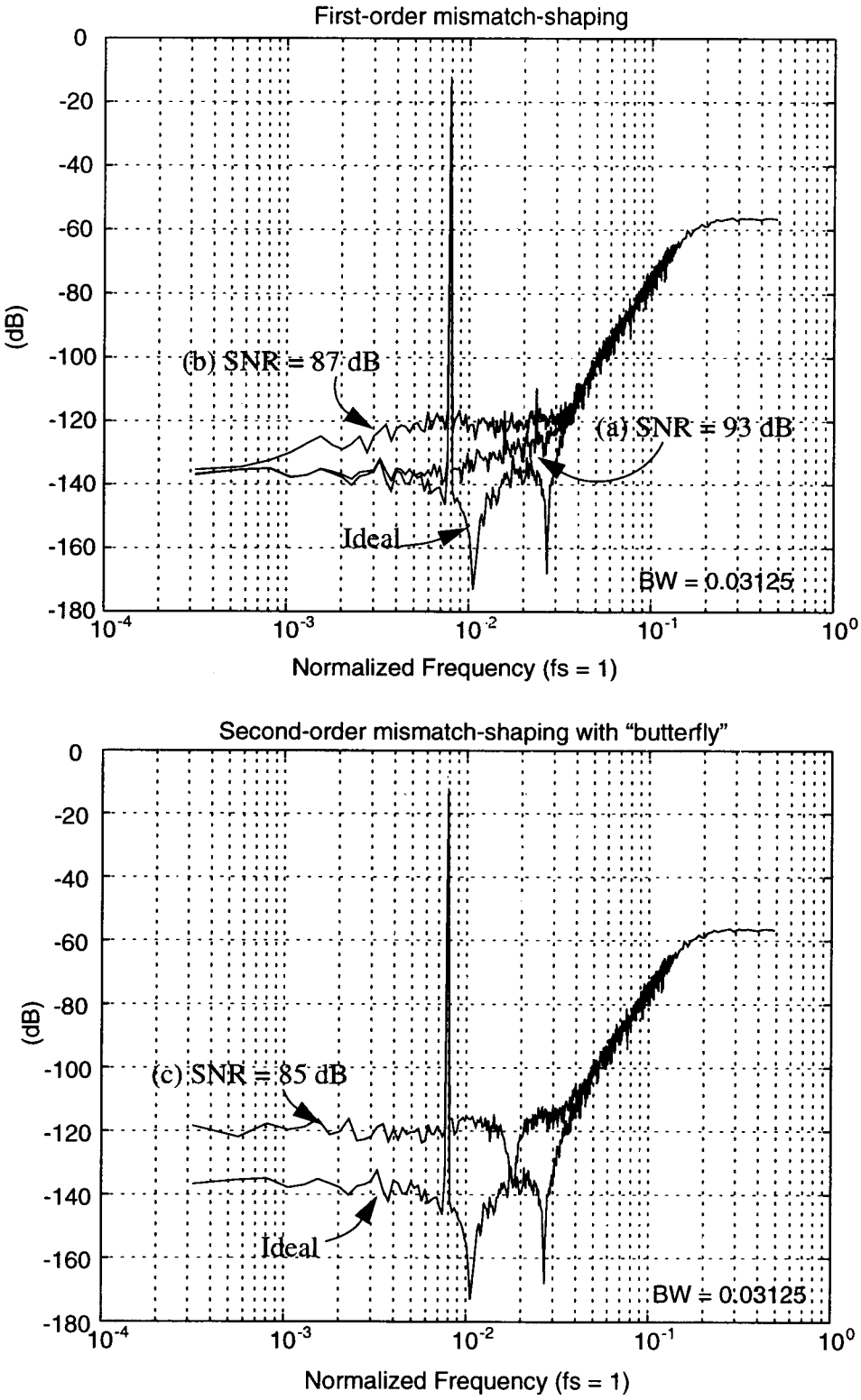


Figure 3.14:SNR and spectra of an 4th-order 17-level delta-sigma DAC with element mismatch-shaping using (a) 1st-order delta-sigma ESL, (b) 1st-order butterfly shuffler and (c) 2nd-order butterfly shuffler.

When the same MTF $H(z) = \frac{z^2 - 2z + 1}{z^2 - 1.25z + 0.5}$ as that used in the comparison between the tree shuffler and the delta-sigma ESL in Section 3.1 is chosen, the DAC driven by the butterfly shuffler demonstrates no mismatch-shaping, indicating that the noise-shaping loops inside the swappers are not stable. A new 2nd-order MTF with an infinity norm of 1.2 was synthesized and optimized with $OSR = 16$. with this second MTF, the butterfly shuffler is stable and achieves an SNR of 85 dB, which is below that achieved by a 1st-order MTF! This indicates that the selection logic is only barely stable even with this carefully designed MTF. The output spectrum is also shown in Figure 3.14.

Butterfly shufflers do not produce SNRs as good as delta-sigma ESLs because the element usage patterns are limited by the particular way in which the butterfly shuffler is wired. This tends to produce larger "selection errors" than that of a competing delta-sigma ESL.

To drive a unit-element DAC containing 2^n elements, a n-stage butterfly shuffler with 2^{n-1} swappers in each stage is needed. Therefore the total number of swappers is $n \cdot 2^{n-1}$. For $n > 2$, more noise-shaping loops are needed in a butterfly shuffler than those in a delta-sigma ESL or a tree shuffler; and as n increases, the difference becomes larger. Also, a butterfly shuffler requires the number of the DAC elements to be a power of 2.

3.4 Conclusions

The delta-sigma ESL has the best performance of the three structures examined, because it generates smaller "selection errors" and so can tolerate more aggressive MTFs. However, it is perhaps the most complex in terms of hardware cost, due to the complicated vector quantizer.

The tree shuffler provides a simpler solution to implement high-order element mismatch-shaping by trading off a small amount off the performance.

The butterfly shuffler introduces less harmonic distortion with a 1^{st}-order MTF. However, with higher-order MTFs, it seems to have the worst performance of the three. It is also hardware inefficient. When the number of the DAC elements is sufficiently large, it can be even more complicated than a delta-sigma ESL.

Chapter 4. A Segmented DAC with Mismatch-Shaped MSBs

A multi-bit delta-sigma CMOS switched-current (SI) DAC was evaluated and designed at the system level during the author's summer internship with Analog Devices, Inc. in 1996. The design was targeted at 90 dB+ SNR with an equivalent Nyquist rate of 10 MHz. A highly linear Nyquist-rate segmented CMOS SI DAC, the AD9760 [40], was chosen as the DAC core, thus the design work was mainly architectural. Based on comparisons with a cascade structure, a choice was made to use a direct form 2^{nd}-order lowpass delta-sigma digital modulator to drive the DAC core. The mismatch error is dominated by the MSB cells and suppressed by a 1^{st}-order butterfly shuffler.

4.1 DAC Core

A DAC has two sets of DAC cells: MSB cells and LSB cells. Each set of cells can be viewed as an independent unit-element DAC. Ideally, the sum of the LSB DAC cells is equal to the value of one of the MSB cells minus the value of one LSB cell. The analog output signal is constructed by summing the chosen cells from both DACs. The smallest analog increment of such a 2-stage DAC equals the value of one element in the LSB DAC.

The 12-bit SI DAC used in the design is the 2-stage DAC core of AD9760 developed by Mercer et al. It contains a 127-element LSB DAC and a 32-element MSB DAC. Each of the 32 current sources in the MSB DAC should be exactly 128 times as large as a unit current source in the LSB DAC. Measurements show that the overall DAC achieves up to 14-bit linearity when clocked at 80 MHz.

4.2 MSB-Mismatch Shaping: Direct Form

A complete delta-sigma DAC includes three major parts: an interpolator, a modulator, and a DAC, as shown in Figure 4.1.

The interpolator oversamples the input Nyquist-rate (f_N) digital data with an over-sampling rate $f_S = Rf_N$ and suppresses the spectral replicas centered at f_N, $2f_N$,... (R-1)f_N, where R is the OSR which is chosen to be 8. In this system, a standard 3-stage half-band filter bank is used to produce 8x oversampled 16-bit data.

Following the interpolator is a delta-sigma modulator which quantizes the data to a shorter word length (12 bits in this design) with most of the quantization noise staying out of the signal band $(0, \frac{f_s}{2R})$. A 2^{nd}-order delta-sigma modulator with $NTF = (1-z^{-1})^2$ is used.

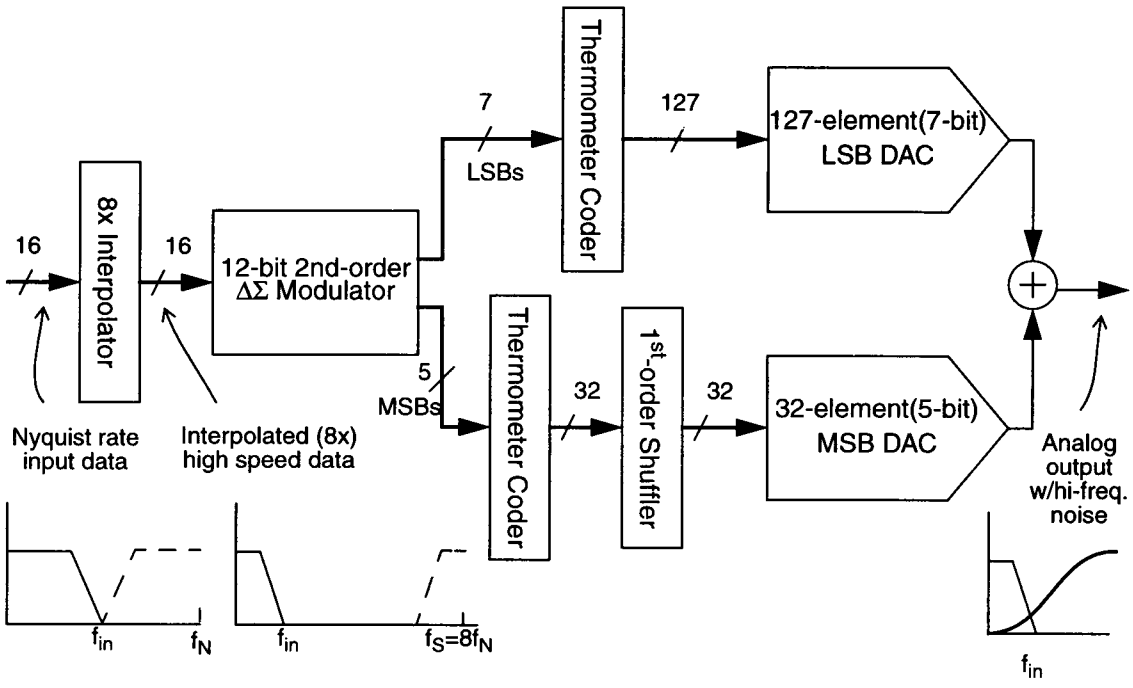


Figure 4.1: System diagram for a 12-bit delta-sigma DAC.

The 12-bit output data of the modulator is then divided into two parts: the lower 7 bits are converted to a 127-bit thermometer code and passed directly to the LSB DAC driving 127 unit current cells. The upper 5-bit data from the modulator is used to drive the MSB DAC. A thermometer decoder converts the MSB data into a 32-bit thermometer code, and a 1^{st}-order butterfly shuffler scrambles the data so that the error introduced by the MSB DAC is 1^{st}-order shaped. The sum of the MSB and the LSB currents, which is the analog representation of the modulator output, is then passed to an off-chip analog filter which attenuates the out-of-band noise and the spectral replicas.

In addition to the direct form modulator described above, the cascade structure has also been evaluated. Comparisons were made between the two structures using simulations.

4.3 MSB-Mismatch Shaping: Cascade Structure

Figure 4.2 shows the system diagram of a cascade structure. The major difference between the structure shown here and a traditional cascade digital modulator [41 - 43] is that, instead of summing the two paths digitally, it takes advantage of the 2-stage DAC structure and uses the two digital paths to drive the LSB DAC and MSB DAC, separately. Once again, a 1^{st}-order butterfly shuffler is used to noise-shape the MSB mismatch error.

In this example, the modulator loop quantizes the 16-bit input signal u to 5 bits, introducing a quantization error e_0 which is spectrally shaped by the NTF H, and passes a 7-bit representation of the quantization error (denoted as e_1) through a digital filter which has the same transfer function, H. Assuming the gains of the LSB DAC and the MSB DAC are A_1 and A_2, respectively, the conceptual DAC output is:

$$DA_{out} = A_2U + A_2H(z)E_0 - A_1H(z)E_1 = A_2U + H(z)(A_2E_0 - A_1E_1) \quad (4.1)$$

If the LSB DAC and the MSB DAC were perfectly matched, the quantization error left (i.e. $e_0 - e_1$) would be equivalent to the error introduced by 12-bit quantization.

Note that the signal term in Eq. (4.1) is only related to the gain of the MSB DAC, while for a direct form modulator such as the one shown in Figure 4.1, the construction of the analog signal depends on both the MSB DAC and the LSB DAC. Furthermore, the mismatch error between A_1 and A_2 is suppressed by $H(z)$ in the band-of-interest. Therefore, the cascade structure could be insensitive to the mismatch between the MSB DAC and the LSB DAC. However, the above discussion is based on one assumption: the feedforward quantization error does not cause the digital filter $H(z)$ to overflow. In the case when overflow happens, the overflow data is carried to the signal path. This carry causes two additional digital error terms with the same value (E_c) but opposite signs to be added into the two paths. The DAC output can therefore be written as:

$$DA_{out} = A_2 U + A_2 H(z) E_0 - A_1 H(z) E_1 + A_2 E_c - A_1 E_c . \qquad (4.2)$$

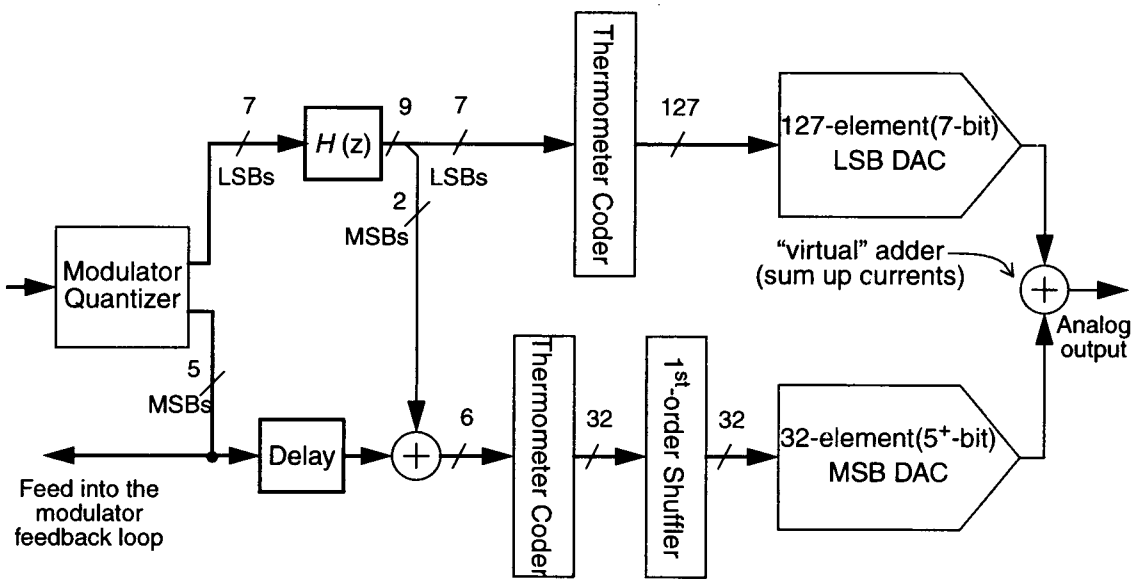


Figure 4.2: A cascade structure with 12-bit equivalent quantization.

If there were no mismatch between the LSB DAC and the MSB DAC, these two error terms would cancel each other. When mismatch exists between the two DAC parts, it introduces an additional error term which is not taken care of by the NTF. This drawback might ruin the advantage of the cascade structure.

One disadvantage of the cascade structure, compared to a direct form modulator having the same number of the quantization levels, is that the stability issue is more critical for the cascade structure, because the modulator loop filer needs to "correct" larger quantization errors.

4.4 Simulation Results

The systems discussed in the previous sections are simulated and compared under different DAC matching conditions. Some of the results are summarized in Table 4.1. The

Table 4.1:SNR of the ΔΣ DAC with different structures and matching conditions.

	Test Condition	Direct form SNR	Cascade Structure SNR
0	Ideal	104.9 dB	105.2 dB
1	7-bit LSB VS. MSB match., everything else ideal	81.2	84.6
2	9-bit LSB VS. MSB match., everything else ideal	93.1	96.1
3	12-bit LSB VS. MSB match., everything else ideal	104.3	104.7
4	3-bit LSB matching, everything else ideal	83.7	83.2
5	7-bit LSB matching, everything else ideal	104.4	104.8
6	11-bit MSB matching, everything else ideal	83.7	80.6

Table 4.1:SNR of the ΔΣ DAC with different structures and matching conditions.

	Test Condition	Direct form SNR	Cascade Structure SNR
7	12-bit MSB matching, everything else ideal	94.2	93.1
8	13-bit MSB matching, everything else ideal	97.5	96.1
9	13-bit MSB match., 7-bit LSB match., 11-bit MSB/LSB match.	98.5	98.4
10	AD9760	96.9	95.6

simulations used the same NTF $H(z) = (1 - z^{-1})^2$ and OSR (8). A single-tone at 1/10 of the band-of-interest (i.e. 0.5 MHz for f_N = 10 MHz) with a magnitude of 7/8 of full scale was applied as the test signal.

Tests 1-3 demonstrate the idea discussed in the previous section that the cascade structure is less sensitive to the gain mismatch between the LSB DAC and the MSB DAC: providing about a 3 dB SNR advantage. However, according to the measurements and the simulations, the matching between the MSBs and the LSBs is not the dominant source of error for the AD9760 DAC core. The tests 4-5 also show that the mismatch error in the LSB cells does not affect the performance significantly: 7-bit matching is sufficient. The dominant error source turns out to be the MSB DAC mismatch. Therefore, only mismatch-shaping on the MSB DAC is needed. Since the DAC with the direct form modulator handles this term better than the cascade structure does, the direct-form structure is chosen to implement the system.

4.5 Bit-True Details

The error feedback structure was used to build the modulator. Bit-true simulations were done with ADICE5 [44], a mixed-level circuit simulator used within Analog

Devices Inc. The bit-true 2nd-order 12-bit lowpass modulator is drawn in Figure 4.3. Although it is known that the error feedback structure is sensitive to coefficient inaccuracy which is a tricky issue in an analog modulator, it is a very simple structure for a digital modulator, especially in the multi-bit case. As illustrated in the second diagram, there are only two adders (ADD1 and ADD2) in the system. ADD1 is a 5-bit adder, while

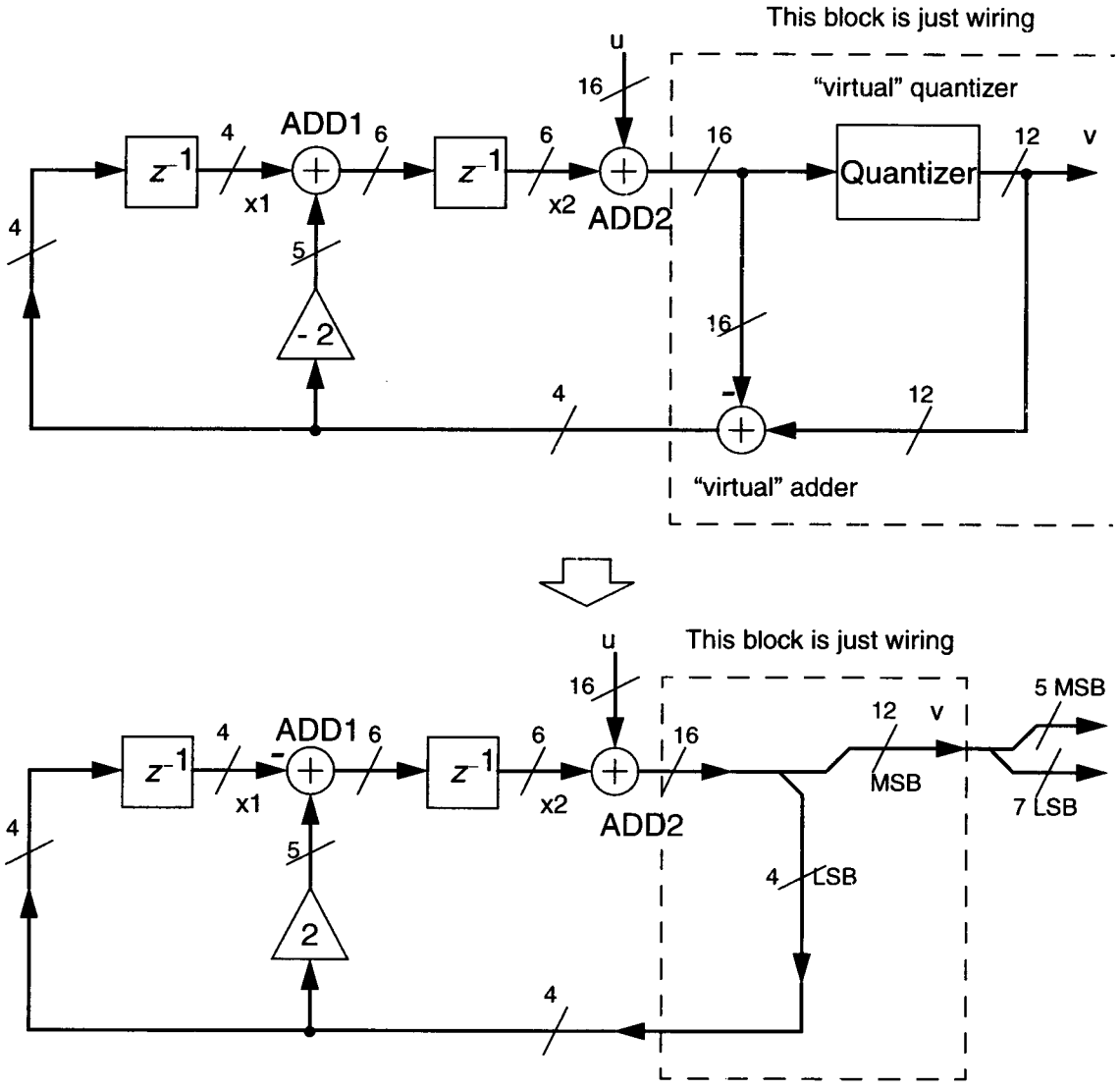


Figure 4.3: The bit-true system diagram of the 2nd-order 12-bit lowpass digital ΔΣ modulator.

ADD2 is a 16-bit adder. However, ADD2 can be simplified by taking advantage of the fact that one of the inputs ($x2$) is only 6 bits wide.

Another way to simplify ADD2 is to combine it with the thermometer decoders which follow the modulator. As illustrated in Figure 4.3, the 12-bit output of the modulator is split into two parts. The 5 MSBs are converted to a thermometer code and passed to the shuffler, and the 7 LSBs are also converted to a thermometer code. It is then possible to make a thermometer decoder which takes the upper 5 bits from the input u directly, and outputs its 31-bit thermometer code to the top 31 bits of the 32-bit input of the shuffler, leaving the bottom bit blank. The remaining 11 bits (on the LSB side) of u are added to the 6-bit $x2$, with the carry output of ADD2 passed directly to the bottom bit of the shuffler input. The idea described above is illustrated in Figure 4.4. However, this structure does not work for Robertson's thermometer code shuffler [39] which only takes true thermometer codes because now the input to the shuffler is not a true thermometer code.

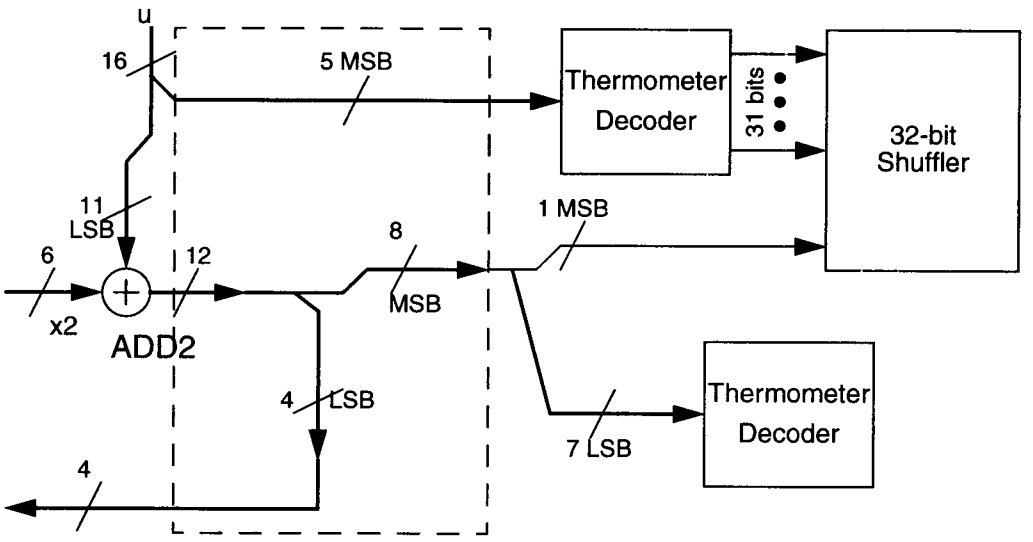


Figure 4.4: An improved structure with the adder A2 cut down to 11-bit.

Also note that, in Figure 4.3, ADD1 and ADD2 are in series, which means that both of the additions need to be done within one clock cycle (which corresponds to 12.5 ns for a clock rate of 80 MHz). However, in order to do its addition, ADD1 only needs to know the lowest 4 bits of the output of ADD2, which means that ADD2 can take its time to do the rest of its own addition as long as it is done before the next cycle starts. In other words, ADD2 has a whole clock cycle to accomplish its job.

Although the output of ADD2 can be as wide as 17 bits, the carry output bit was simply thrown away in the simulations. Simple logic can be added into the system to turn on all the current cells when such an overflow happens, although simulations did not show much improvement by doing so.

4.6 Conclusions

In a 2-stage DAC, the mismatch error is most likely dominated by the MSB cells. Therefore, instead of shuffling all the DAC cells, mismatch-shaping only on the MSB cells suppresses the mismatch error efficiently, and at the same time keeps the hardware cost low. Matching between the MSB DAC and the LSB DAC is also an important issue.

Chapter 5. A 16-Element Switched-Capacitor Mismatch-Shaping DAC

In Chapters 2 and 3, the theory of mismatch-shaping was introduced and several mismatch-shaping schemes were discussed. In this Chapter, a 16-element switched-capacitor (SC) mismatch-shaping DAC is described as a design example. The DAC is driven by 1^{st}-order lowpass and 2^{nd}-order bandpass ESL circuits. Two different schemes discussed in Chapter 3, namely, the general mismatch-shaping scheme (which is equivalent to the element-rotation in the 1^{st}-order lowpass case) and the butterfly shuffler scheme, are used. Three lowpass $\Delta\Sigma$ modulators (two 4^{th}-order, one 8^{th}-order) are designed to drive the mismatch-shaping DAC. An 8^{th}-order bandpass modulator is also derived from a 4^{th}-order lowpass prototype to facilitate the demonstration of bandpass mismatch-shaping. The analog part was designed and fabricated in the Orbit 1.2 μm CMOS process [45, 46]. The digital part, which includes the modulator and the ESL was implemented on a Xilinx 4010 FPGA [47].

5.1 Motivation

A simple block diagram of a $\Delta\Sigma$ SC ADC is shown in Figure 5.1. The purpose of the system is to demonstrate the usefulness of the mismatch-shaping techniques described in the previous chapters. This technology is expected to help to extend the frequency range of $\Delta\Sigma$ ADCs and DACs, which are renowned for their high accuracy but not for high

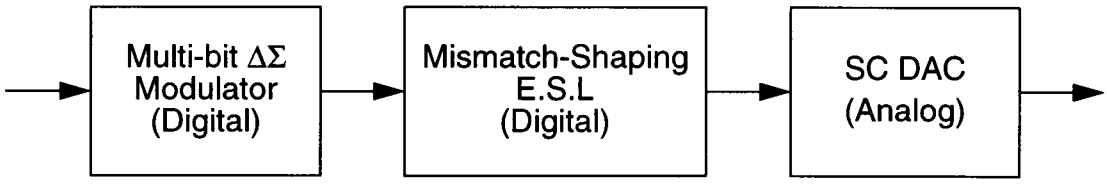


Figure 5.1: Block diagram of a $\Delta\Sigma$ ADC.

bandwidth. The communications field is most likely to be the driver for this technology. Possible application areas also include high quality audio, multi-media network applications, and other systems which require high dynamic ranges and wide bandwidths.

5.2 System-Level Definitions and Specifications of the ΔΣ Modulator

The 16-element SC DAC was designed in the Orbit 1.2 μm CMOS process. Since no particular application is targeted, there are no specific performance requirements. However, in order to optimize the modulator for the purpose described above, a few system level decisions regarding the number of quantization levels, the OSR and the order of the modulator need to be made.

5.2.1 General Design Chart for Multi-Bit ΔΣ Modulators [48]

As mentioned earlier, multi-bit ΔΣ modulators possess two major advantages over modulators employing binary quantization. First, multi-bit quantization makes higher order modulations with more aggressive NTFs feasible. Therefore, the in-band quantization noise can be reduced. Second, the output of a multi-bit ΔΣ modulator is a smoother waveform which tracks the desired signal better than the full-scale, PWM-like square-wave output of a single-bit modulator. Unfortunately, these two advantages of multi-bit quantization compete with one another because a more aggressive NTF means a larger out-of-band gain which results in higher out-of-band noise and thus poor tracking in the time domain.

The design chart in Figure 5.2 [48] illustrates the trade-off between SNR and time-domain tracking. Figure 5.2 plots the SNR (for an oversampling ratio of 8) as a function of the rms error in the output for a variety of modulator designs. To eliminate the possibility of instability due to quantizer overload, the quantizer was given an infinite

number of levels. The SNR was measured using a sine wave input with a peak-to-peak amplitude of 1 LSB. Level curves for the infinity norm of the NTF are also shown. Although the SNR values are computed using OSR = 8, the SNR for other OSRs can be estimated by noting that the SNR of an n^{th}-order modulator increases at roughly $(6n + 3)$ dB per octave of oversampling. Furthermore, an examination of the simulation data underlying Figure 5.2 yields the empirical rule that the maximum peak-to-peak input amplitude is $m - 7e_0$ LSBs, where m is the number of quantization steps and e_0 is the rms output error for a 1 LSB peak-to-peak input (i.e. the horizontal axis in the plot.) Specifically, the peak SNR for an n^{th}-order modulator with $(m + 1)$ quantization levels can be estimated using the following empirical formula

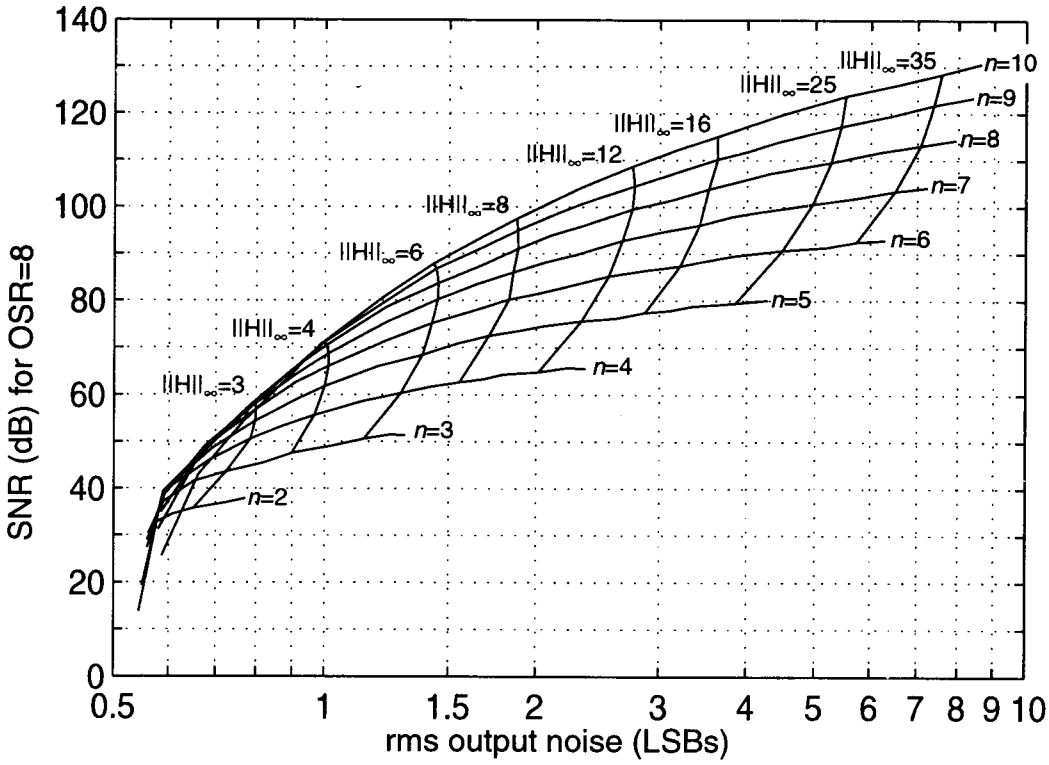


Figure 5.2: Design chart for multi-bit ΔΣ modulators.

$$SNR_{peak} \approx SNR_{Fig4.2} + (6n+3)\log_2(\frac{OSR}{8}) + 20\log_{10}(m - 7e_0) \text{ dB}. \qquad (5.1)$$

For example, given an OSR of 16 and 16 quantization levels, an order of at least 4 is needed to build a modulator which keeps the rms quantization error below 1 LSB (given a 1 LSB peak-to-peak input), while at the same time achieving an SNR of more than 100 dB.

5.2.2 Number of Quantization Levels

Theoretically, the larger the number of quantization levels, the better the performance that can be achieved, because there is less quantization noise generated in the first place. The modulator loop tends to be more stable and larger input signals are allowed, therefore higher peak SNRs can be expected.

However, to make use of more accurate quantization, a more complicated DAC is needed to produce finer analog output steps. From the point of view of circuit implementation, this usually means larger chip area, higher power consumption, and more complicated wiring and layout. Worst of all, a more complicated and larger ESL is required to do mismatch-shaping since there are more DAC elements (and hence, more degrees of freedom) to handle. Therefore, a compromise is needed.

According to Eq. (5.1), the number of quantization steps $(m + 1)$ only affects the last term $20\log_{10}(m - 7e_0)$ which is the maximum peak-to-peak input amplitude from the empirical rule. This term is plotted as a function of m in Figure 5.3. The rms error e_0 is assumed to be 1. From the plot, the maximum input magnitude (in dB) increases rapidly when the number of quantization steps is small. When the number of the quantization steps is large, the slope becomes smaller.

Based on the above discussions and Figure 5.3, $m = 16$ was chosen for the mismatch-shaping DAC in this project. This requires a DAC with 16 unit elements. Note that it is often convenient to choose the number of the unit elements to be in the form of 2^n. In this example, $n = 4$.

5.2.3 Oversampling Ratio

Usually, the application should be the number one issue that determines the value of the OSR. When carrying out the design of such a product, the ideas of the signal bandwidth f_b and maximum clock rate f_s at which we can run the circuit should already be known to the designer. The maximum OSR is then $\frac{f_s}{2f_b}$. Of course, one can always

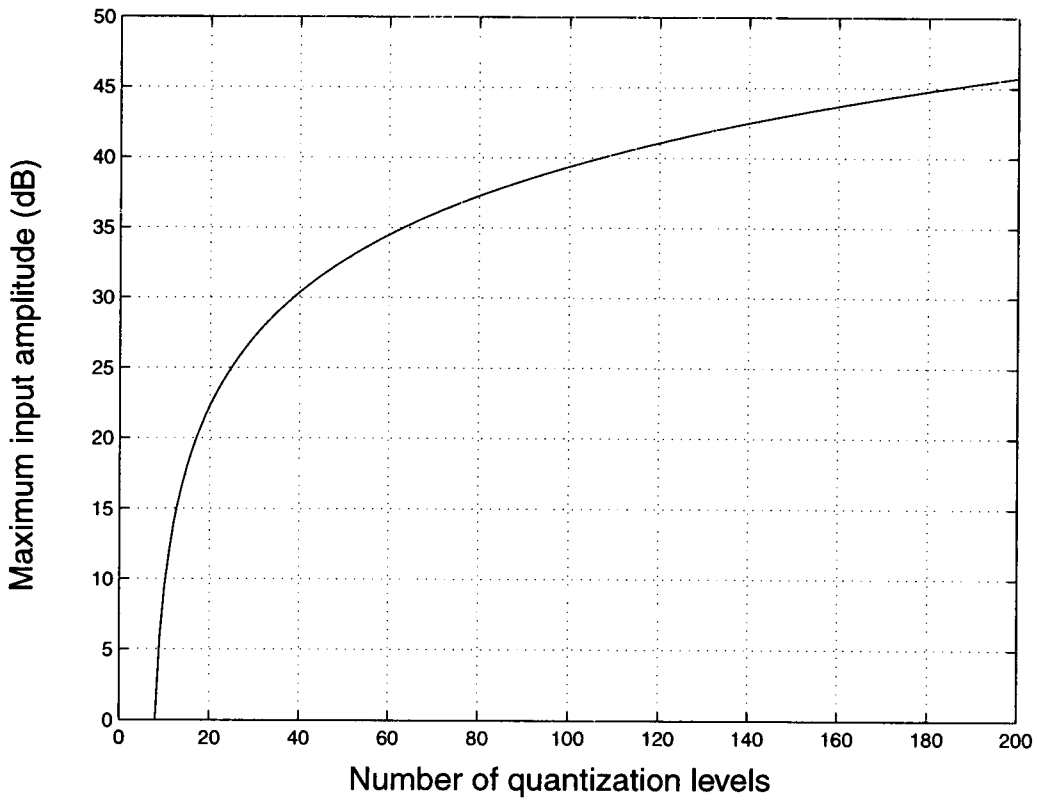


Figure 5.3: Maximum input amplitude VS. number of quantization levels.

choose to optimize his/her design and run the system at a slower speed and hence a lower OSR as long as it meets the SNR and other application requirements.

Since there is no specific applications targeted by this design example, the above method is not useful in determining the OSR. The SNR of an n^{th}-order modulator increases at roughly $(6n + 3)$ dB per octave of oversampling. However, low OSRs are desirable because one of the key motivations for going to multi-bit modulators is to expand the application range of $\Delta\Sigma$ technology. Therefore, 16 and 8 were both chosen to be the primary OSRs in this project so that some comparisons can be made. According to Figure 5.2 and Eq. (5.1), an OSR of 16 makes it possible for a modulator with an order of 4 or higher to achieve a peak SNR that is around or above 100 dB, given that the number of unit DAC elements is $m = 16$.

Although the modulator was designed and optimized at these particular OSRs, please note that one is always free to define any other value as the OSR of the DAC by changing the definition of the band-of-interest and focusing on a different bandwidth in the spectrum of the output analog signal.

5.2.4 Order of the Modulator

Most $\Delta\Sigma$ modulators on production lines today are low order modulators (i.e. 1^{st}- or 2^{nd}-order), although it is known that higher order NTFs can suppress the in-band noise more efficiently. There are two major obstacles which keep people away from higher order modulators. One is stability, the other is complexity. In the multi-bit case, the stability issue is laid to rest. Complexity is the major problem to be dealt with.

Although in many cases not much difference is made by increasing the order of a modulator with a low OSR because the magnitude response of a high-order NTF ramps up rapidly with frequency, it is the author's belief that the improvement would be noticeable if a high-order modulator is properly designed and optimized. Another consideration when making the decision is that it is one of the primary objectives of this thesis to demonstrate the feasibility of high-order modulators with multi-bit quantization. Therefore, the decision was made to design a 4^{th}-order and an 8^{th}-order modulators so that comparisons can be made.

As estimated before, with $OSR = 16$, it is possible to design a 4^{th}-order modulator to achieve a peak SNR around 100 dB if only the quantization noise is considered. Should we be able to suppress the quantization noise to such a level, the noise floor would likely be dominated by some other noise source, for example, mismatch noise. Therefore, increasing the modulator order would not make a measurable difference in the overall SNR. However, if the OSR is reduced to 8, a 4^{th}-order modulator will only be able to make the in-band quantization noise 80 dB below the peak signal, while an 8^{th}-order modulator can still make it better than 90 dB. Therefore, the difference should be visible if the design is carried out carefully enough to knock down other error sources below this level.

5.2.5 Summary of the System-Level Definitions

The system-level definitions and specifications are summarized in Table 5.1.

Table 5.1:System-level definitions and specifications.

# of Elements	OSR	Mod. Order	SNR
16	16 and 8	4 and 8	> 90 dB

5.3 System Diagram

The system diagram of the mismatch-shaped multi-bit ΔΣ SC DAC is shown in Figure 5.4. The system consists of two major blocks, the digital block and the analog block.

The top block in Figure 5.4 shows the digital portion of the system, which contains a ~4-bit (17-level quantization) delta-sigma modulator, an ESL which implements a mismatch-shaping algorithm, a pseudo-random scrambler, and a digital sine wave generator which supplies input digital test signals for the entire system. A Xilinx FPGA chip was used to implement the whole digital section. The digital signal generator was simply a pair of EPROMs which store a 16-bit digital sine wave. The printed circuit board (PCB) for the FPGA was designed by Forrest Hudson [49].

The analog portion of the system is shown in the bottom part of Figure 5.4. A 16-element SC DAC was designed and fabricated in the Orbit 1.2 μm CMOS process. The chip contains a digital interface, a 16-element SC DAC, a 1^{st}-order SC smoothing filter, a clock generator and a biasing circuit. A PCB was built as a test bed for the SC DAC.

The system operates as follows. The delta-sigma modulator takes the oversampled digital sine wave, *u,* which is stored in the EPROMs and quantizes the data to 4 bits, leaving the most of the quantization noise out of band. The ESL then translates *v* to the 16-bit *sv* vector using mismatch-shaping algorithms. Before transferring the data to the analog DAC board, a pseudo-random scrambler is used to scramble the *sv* lines to whiten their spectra and reduce the likelihood of corrupting the reference. The scrambled *sv*

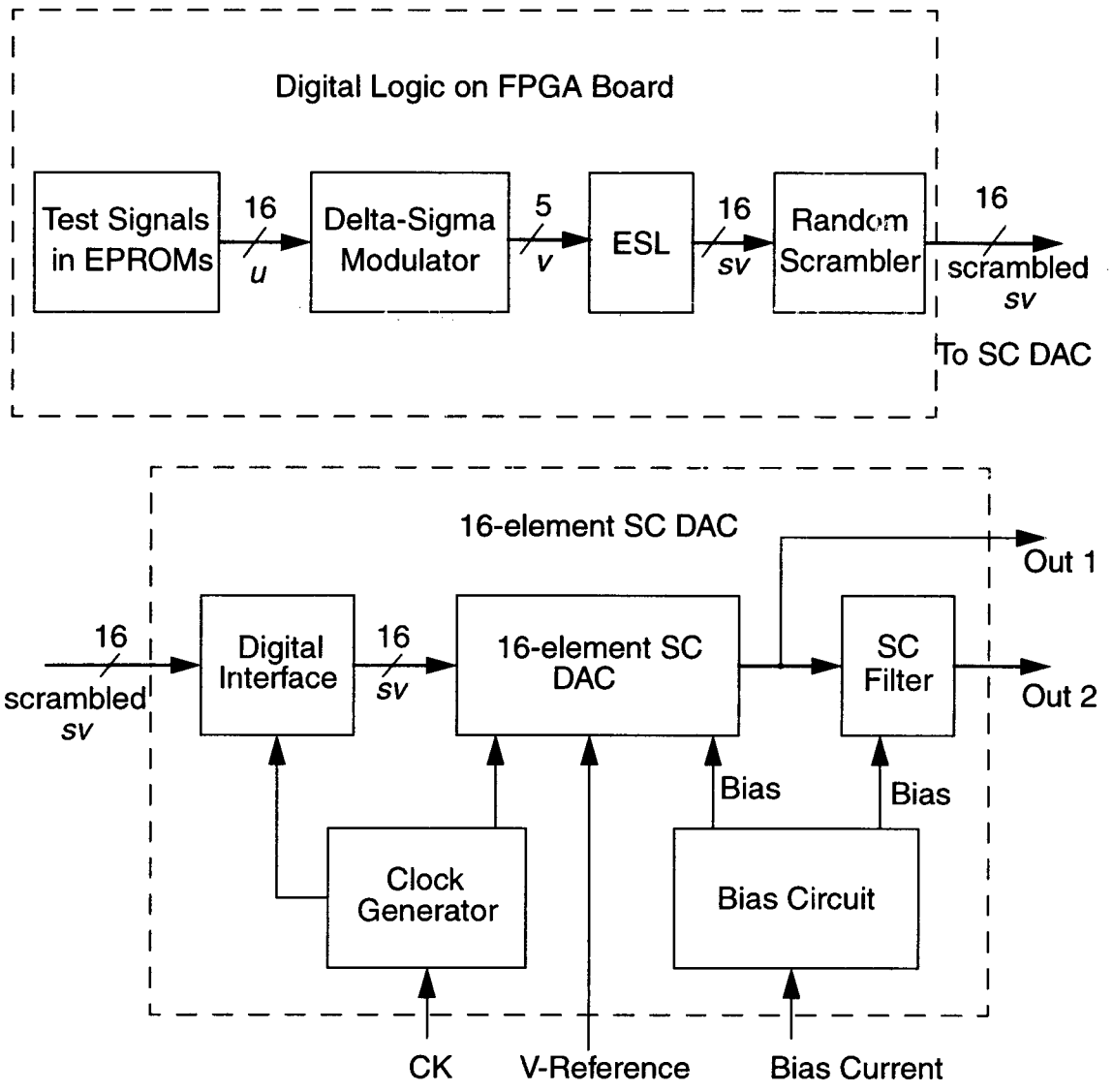


Figure 5.4: System diagram.

vector is then transferred to the SC DAC digital interface which unscrambles the data and synchronizes it with the internal DAC clock phases. The recovered *sv* vector is used to drive the DAC core which produces the analog representation of the signal. A 1^{st}-order SC discrete-to-continuous lowpass filter is put at the end of the main signal path. This SC filter smooths the output waveform by filtering out the high frequency noise. The SC filter can be turned off and the unfiltered DAC output signal can be measured directly so that bandpass systems can be implemented and tested.

Each block of the system is described in the following sections. Some alternative designs are also discussed.

5.4 16-Element SC DAC

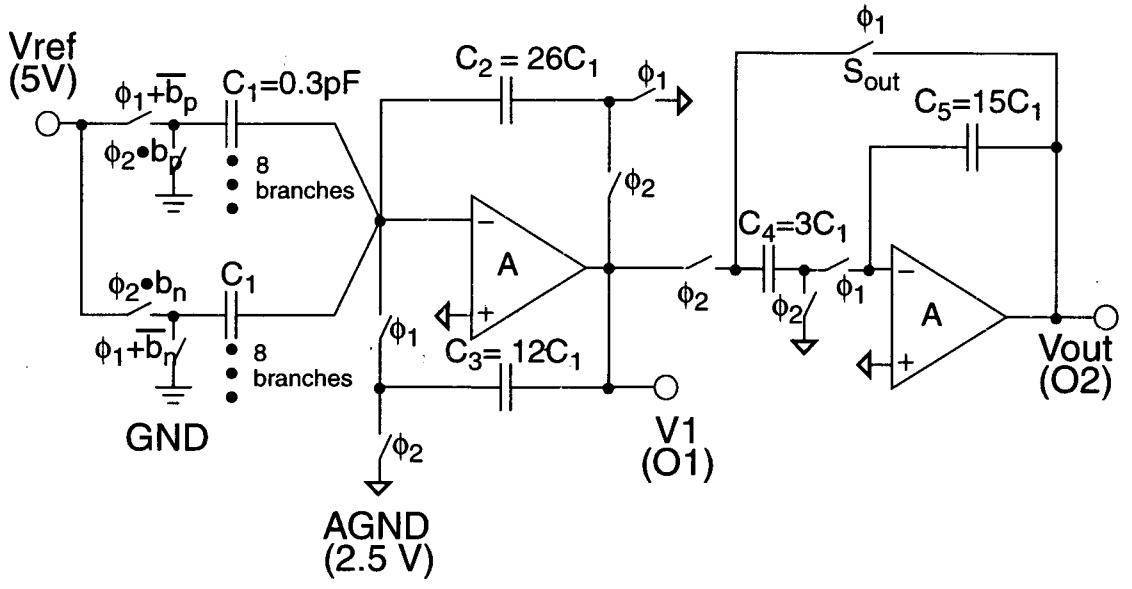
This section describes the main functional blocks of the 16-element SC DAC in terms of their design, layout and test procedure. In the first section, the overall structure of the DAC is introduced. Then each of the following sections deals with a particular block. The floor plan and the chip pin list are given last.

5.4.1 Top Level -- the DAC

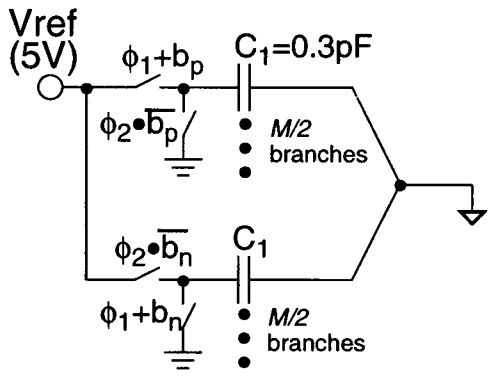
In this section, the main DAC core and the lowpass SC filter are described.

Figure 5.5 shows a switched-capacitor DAC which follows the amplifier structure of Haug et al. [50], along with an SC lowpass filter [51 - 53]. A single-ended version was used because it was viewed as more useful (no differential to single-ended circuit needed), as well as being easier to construct. Furthermore, MATLAB simulations showed that second-order opamp nonlinearity (which would be cancelled in a differential implementation) does not cause significant performance degradation.

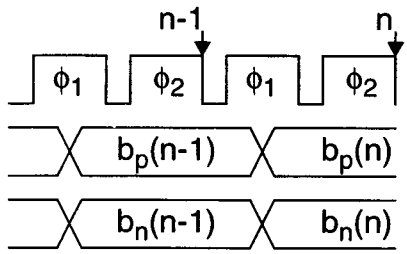
The first stage of the circuit is the main block of the design — a 16-element SC DAC. It is essentially a gain- and offset-compensated amplifier with a non-return-to-zero output. For low frequencies, the output of this first stage is given as



SC DAC and Lowpass Filter



Dummy Input Branch



Simple Clock Diagram

Figure 5.5: The SC DAC and output buffer.

$$V_1 = \frac{\sum_{i=1}^{16} (b_{p,i} - b_{n,i}) C_1 / C_2}{1 + (1 + C_1/C_2)\mu^2} V_{ref}, \tag{5.2}$$

where $\mu = 1/A$, and A is the opamp gain. Note that the gain of the op amp is effectively squared by this circuit for low frequency applications. Please refer to Figure 5.5 for explanations of $b_{n,i}$, $b_{p,i}$, C_1, C_2 and V_{ref}.

Since half of the unit-element array is used to produce positive outputs and half is used to produce negative outputs, this DAC is capable of bipolar operation. Because the *sv* vector from the ESL is unipolar, the b_n bits are the inverted logical levels of the corresponding *sv* bits. Both b_n and b_p are only allowed to change during ϕ_1. There are basically 2 phases needed to operate the circuit. However, the phases on some particular switches where charge injection is signal-independent are slightly delayed with respect to those switches where charge injection is signal-dependent in order to minimize distortion caused by charge injection. Specifically, ϕ_1 has two versions, namely ϕ_1 and ϕ_{1d}, and ϕ_2 also has two versions, ϕ_2 and ϕ_{2d}.

Aside from the 16 input branches to the DAC, there are 16 dummy input branches. The real branches and the dummy branches are identical in terms of their structure, but opposite in terms of operation so that there are always 16 unit capacitors to charge and discharge. The purpose of the dummy branches is to make the load on the voltage reference signal-independent.

A simple on-chip 1st-order SC lowpass filter (also shown in Figure 5.5) [51, 53] was also designed. The transfer function of the filter is

$$H(z) = \frac{C_4/C_5}{(1 + C_4/C_5) - z^{-1}}. \tag{5.3}$$

This lowpass filter helps to deal with the clock jitter problem because the output sampled-and-held signal contains lower high frequency components as a result of the lowpass filtering. This SC filter can be turned off and the unfiltered DAC output signal V_1 is then available directly at O_1.

5.4.2 Capacitor Sizing

The capacitor sizes are decided mostly based on thermal noise considerations [52, 54]. Thermal noise is caused by the random fluctuation of carriers due to thermal energy and is present even at equilibrium. In an SC circuit, the thermal noise bandwidth is usually limited only by the time constants of the switched capacitors or the bandwidth of the op-amps. Due to aliasing, when such wide-band noise is sampled, the calculation of thermal noise is often complicated. To avoid this complicated calculation and give the design some safety margin, all the thermal noise power was considered as being aliased into the band from 0 to $f_s/2$. The total noise power associated with a switching resistor in series with a capacitor C is $v_n^2 = \frac{kT}{C}$, where k is the Boltzmann constant, and T is the temperature in degrees Kelvin. Assume a 0.707 Vrms peak-to-peak sine signal and $OSR = 8$, the SNR due to thermal noise is

$$SNR_{thermal} = 20\log_{10}(0.707) - 10\log_{10}(v_n^2) + 3\log_2(8). \tag{5.4}$$

In the SC circuit shown in Figure 5.5, the thermal noise is determined by the capacitors in the input branches of the two SC stages. The thermal noise produced in the feedback branches is suppressed by the loop gain and hence is ignored. The total input capacitances to the first- and second- stage were chosen to be 4.8 pF and 0.9 pF, respectively, so that the thermal noise is below the -90 dB level.

5.4.3 Digital Interface and Clock Generator

As shown in Figure 5.6, the clock generator has two parts. The first block contains a clock-divider and a 90^{o}-phase-shifter. The external clock (which is twice as fast as the data rate) is divided by two, and two sets of clock signals (*CK* and *CK90*) 90 degrees apart are generated. One of the clock signals, *CK*, is fed into the second part of the clock circuit, i.e. the SC phase generator which generates the non-overlapping clock phases for the SC circuit. The other, *CK90*, is used to drive the D-flip-flops (DFF) in the digital interface to synchronize the input digital data. The schematic of the clock divider and the phase shifter is shown in Figure 5.7. Note that all the DFFs used in the circuit sample data at the falling edge of the clock.

Figure 5.8 shows a single cell of the digital interface. The function of the interface is to synchronize the input digital data with the internal clock, to descramble the data to recover the *sv* data, and to pass it to the SC DAC core on the right clock edges. The input scrambled *sv* lines are first synchronized with the internal clock by DFFs. Then a decoder is used to descramble the data. Each of the 16 descrambling cells is a mux controlled by a single-bit signal *scr* which is the same pseudo-random bit-stream used in the digital side to scramble the *sv* lines. Each mux does the same XOR function as that of the scrambler

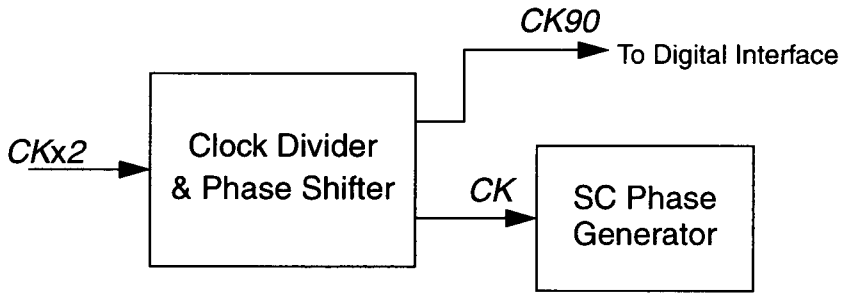


Figure 5.6: System diagram of the clock generator.

cells which will be described in 5.6.2. The following simple derivation shows how the decoder works:

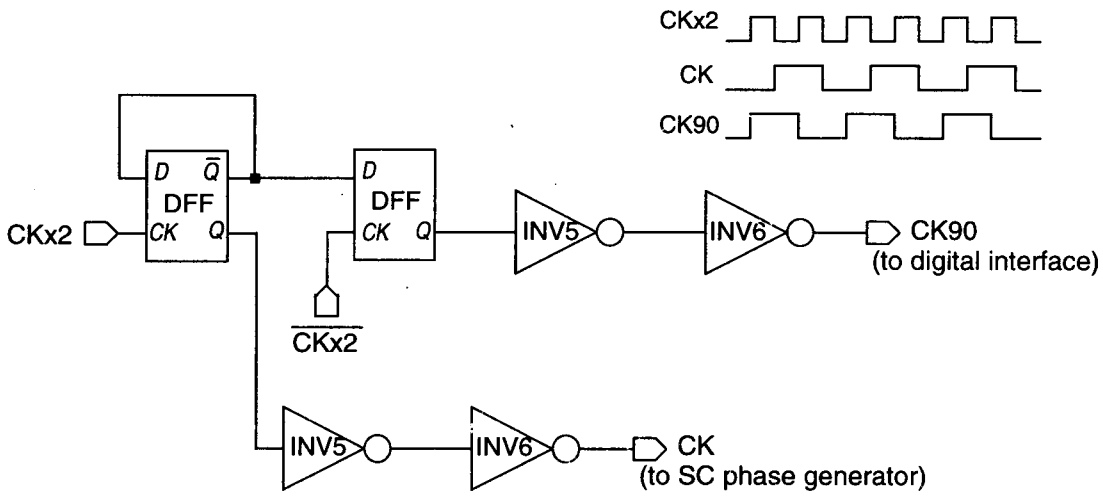


Figure 5.7: Clock-divider & phase-shifter.

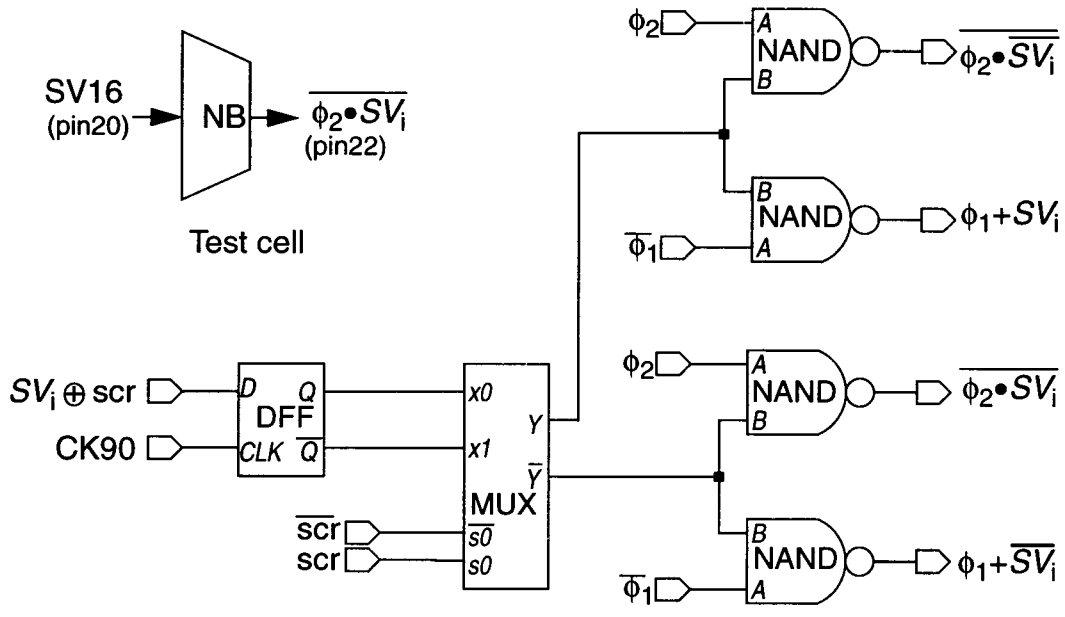


Figure 5.8: A cell of the digital-to-analog interface.

$$scrambled\text{–}sv \oplus scr = (sv \oplus scr) \oplus scr = sv. \quad (5.5)$$

After the *sv* lines are decoded, they are translated to the b_n and b_p bits as described in Section 5.4.1. The last stage of NAND gates make the DAC core see b_n and b_p bits change only during ϕ_1.

The SC clock generator [52] and the corresponding clock diagram are shown in Figure 5.9. All inverters have been sized based on their load capacitances, and each output

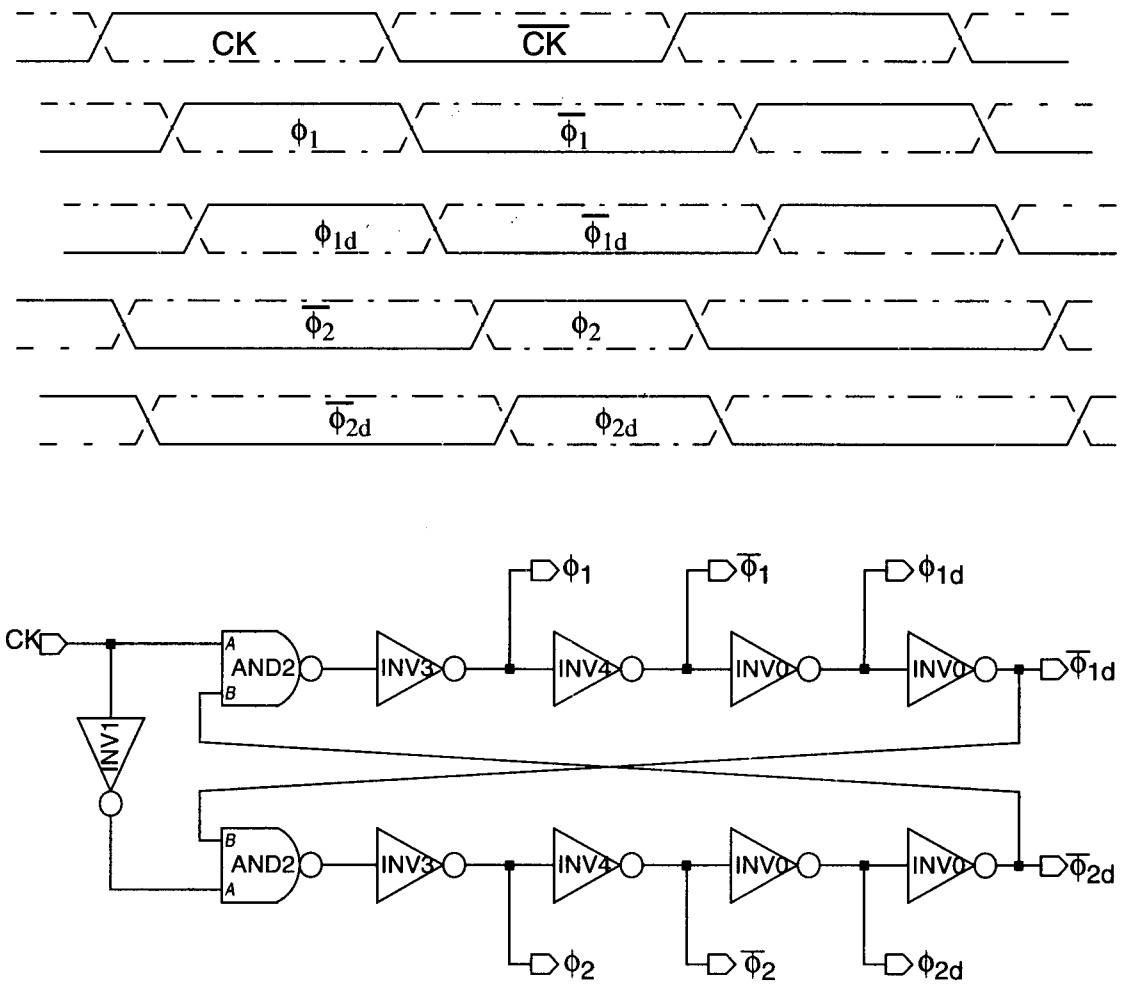


Figure 5.9: SC clock generator and diagram.

from the generator is buffered individually to ensure approximately equal rise/fall times on all clock signals.

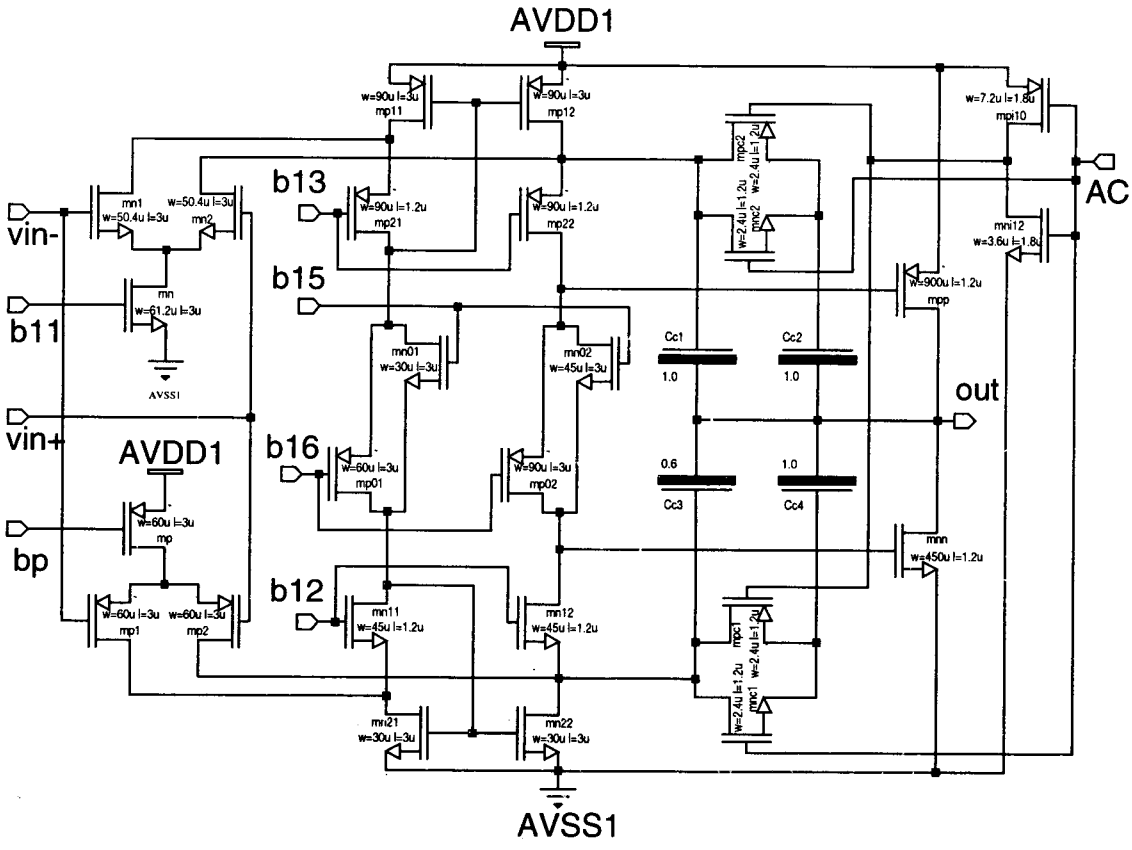
An additional cell of the digital interface was put on the chip to allow digital tests. It uses the SV16 signal as its input. The logical output, $\overline{\phi_2 \bullet SV16}$, can be read from *DTS*.

5.4.4 Op Amp

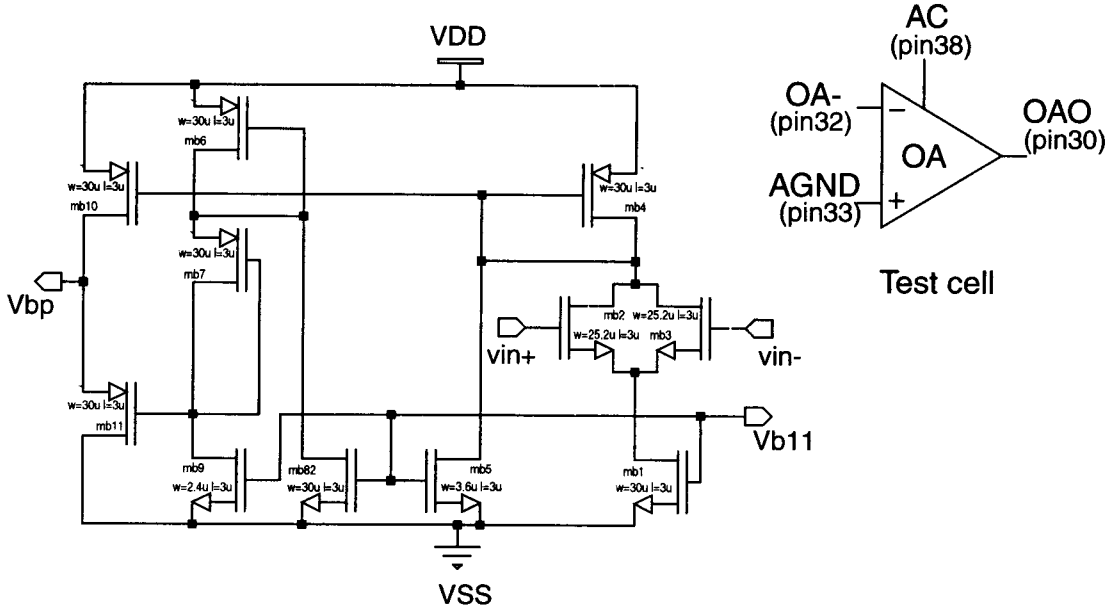
A constant-*gm* rail-to-rail input stage and a class-AB driver are combined to make a highly linear output buffer which prevents output signals from being distorted.

The main circuit is shown in Figure 5.10a [55-58]. An N-channel and a P-channel differential input pair are placed in parallel. This allows rail-to-rail common-mode input voltages. In order to make efficient use of the supply voltage and to obtain a large output swing, a common-source output stage is adopted. The class-AB action is performed by the voltage tracing between the gates of the two output transistors. During slewing, a positive feedback is introduced by the floating control pair (*mn02, mp02*). Another similar pair (*mn01, mp01*) forms a floating current source which does not contribute to the noise and offset of the amplifier. This floating current source also makes the quiescent current in the output transistors insensitive to supply voltage variations, because it has the same supply voltage dependency as that of the class-AB control pair. Additional compensation can be enabled by connecting AC to VDD.

A drawback of the rail-to-rail input stage is that its *gm* varies by a factor of two over the common-mode input range. A way to overcome this effect is to keep the sum of the square roots of the tail currents of the two differential pairs constant as the transconductance of a MOS transistor is approximately proportional to the square root of



(a) Buffer amplifier



(b) Bias circuit

Figure 5.10:The output buffer amplifier and its bias circuit.

the drain current. This can be achieved by properly biasing *mn* and *mp*. A circuit that does this is shown in Figure 5.10b [59].

The test cell is also shown in Figure 5.10. The additional compensation enable pin (AC) is shared by both buffer amplifier cells.

Table 5.2 shows the specs of the opamp with the additional compensation enabled. If AC is disabled, the unity-gain BW increases to 27 MHz, while the phase margin decreases to 45°. Since the slew rate is limited by the compensation capacitances, in AC-disabled case, the values are 66 V/μs and 57/μs for positive and negative slewing, respectively.

Table 5.2:Simulated op amp parameters.

DC gain	Phase Margin	Unity gain freq.	Power supply	Load capacitor	Output swing	Positive slew rate	Negative slew rate
100dB	75°	17MHz	+5V	20pf	1 ~ 4V	27 V/ms	28 V/ms

5.4.5 Bias Circuit [52, 60]

The bias circuit is shown in Figure 5.11. A bias current I_B(100 μA) is injected from outside of the chip. The circuit generates most of the bias voltages for the rest of the chip.

No separate test circuits were designed for the bias circuit. As shown in Figure 5.11, all the bias nodes and the voltage reference output node are connected to pins directly so that they can be tested and overridden if necessary.

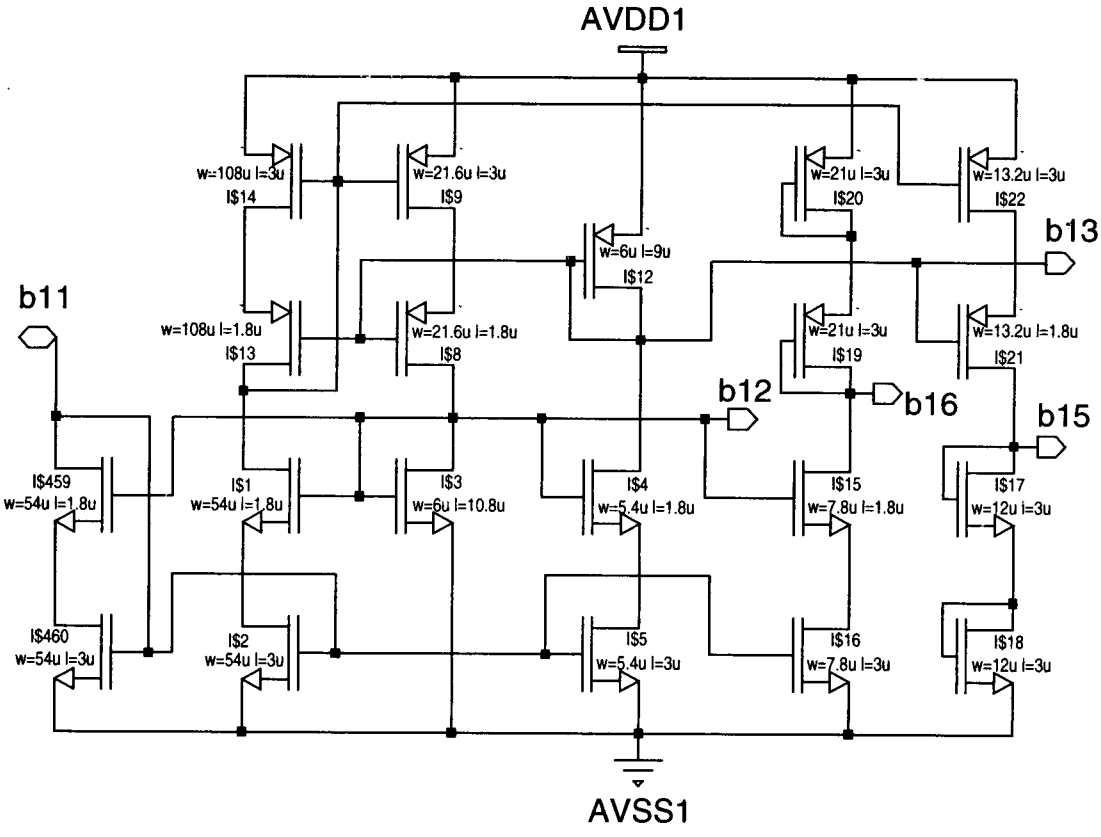


Figure 5.11:Bias circuit.

5.4.6 Floor Plan and Pin-Out of the Chip [52]

The floor plan of the chip is shown in Figure 5.12. The digital interface is sitting on the top of the floor plan, and the op amps and the bias circuit are at the bottom. In between these two major blocks, are the clock generator, the switches and the capacitor

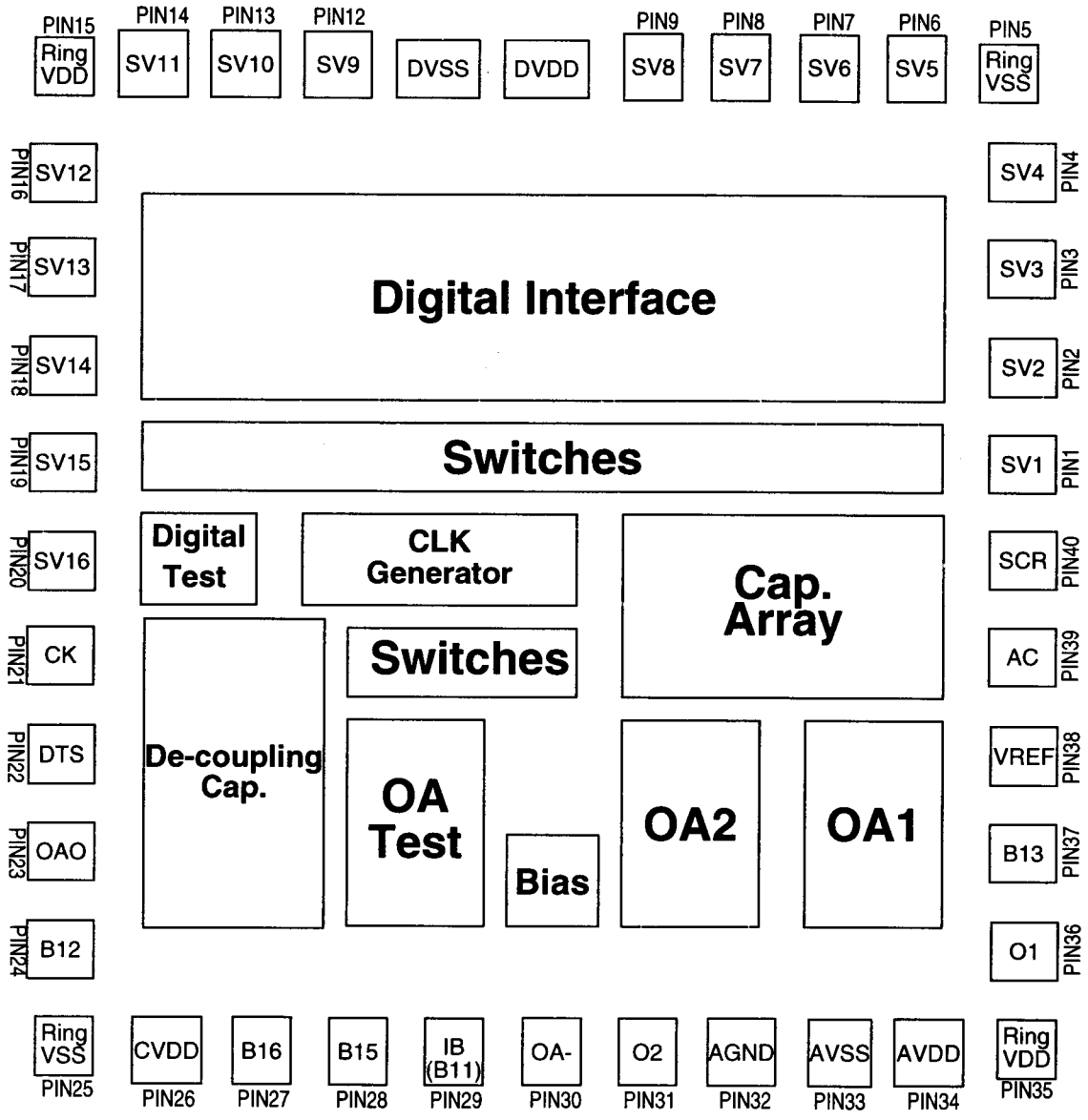


Figure 5.12:Floor plan for the SC DAC layout.

array. To prevent the digital noise from coupling into the analog part, a guard ring surrounds the digital block, some on-chip de-coupling capacitors occupy the blank areas, and the entire substrate is connected to AVSS. Three sets of power supplies run into the chip: the digital VDD (DVDD), the analog VDD (AVDD), and the clock VDD (CVDD). Aside from keeping the noisy clocks away from other power lines, the CVDD also allows the clock generator to be tuned separately. The digital pads and the analog pads are put on opposite sides of the package. The analog output pads and the reference pads are sitting away from the digital part and beside relatively quiet pads.

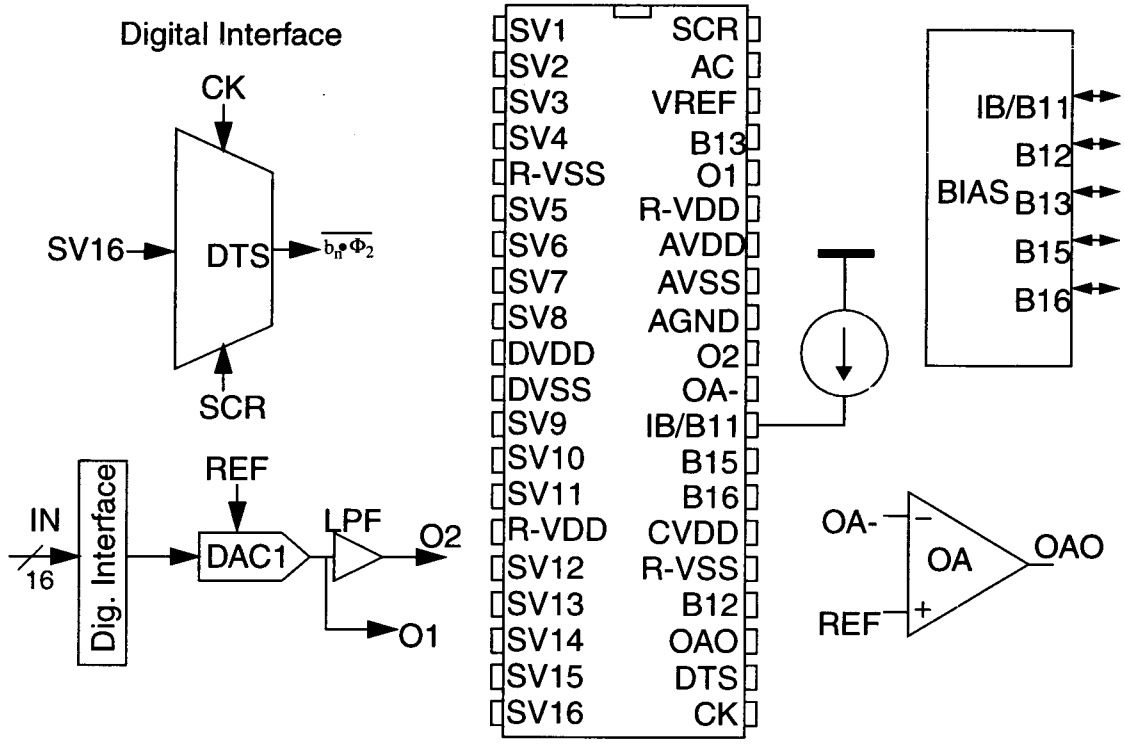


Figure 5.13:Pin-out of the SC DAC chip

A pin-out is shown in Figure 5.13. The 40-pin DIP package is used. Most digital pins are on the left, and the analog pins are on the right. The pins are listed in Table 5.3.

Table 5.3:Pin list of the SC DAC.

SV1-16	DAC input	CK	Clock input	O2	DAC output after LPF	B11-16	Bias test and over-ride
DTS	Testing pins for digital interface	SCR	Scrambling bit	OA-	Negative input for OA test	OAO	Output of the test OA
IB	Off-chip resistor	AC	Additional compensation	AGND	Analog GND (2.5v)	REF	Reference input pin
DVDD	Digital VDD (5v)	DVSS	Digital VSS (0v)	AVDD	Analog VDD (5v)	AVSS	Analog VSS (0v)
R-VDD	Ring-VDD (5v)	R-VSS	Ring-VSS (0v)	CVDD	Clock VDD	O1	DAC output before LPF

5.5 Modulator Design

For comparison purposes, two 4^{th}-order modulators and one 8^{th}-order modulator were designed. The designs were optimized with $OSR = 16$ and/or $OSR = 8$. An 8^{th}-order bandpass modulator was also designed for the demonstration of bandpass mismatch-shaping.

For each of the three lowpass modulators, a NTF was first synthesized and optimized using `synthesizeNTF`, a MATLAB routine in the ΔΣ Tool Box [62]. Then the modulator structure was derived from the NTF. The bandpass modulator was derived directly from a lowpass prototype.

5.5.1 Optimized rms output noise

The parameters need to be supplied to `synthesizeNTF` are the order of the modulator, the OSR, and the infinity norm $\|H\|_\infty$ of the NTF, among which, only $\|H\|_\infty$ has not been specified. As we know, the rms error produced by the quantizer increases with the value of $\|H\|_\infty$. If the rms output noise is specified, the value of $\|H\|_\infty$ can be determined out using the design chart of Figure 5.2.

As mentioned earlier, there is a trade-off between the accuracy of the unfiltered output of a ΔΣ modulator and the SNR of the filtered output. In other words, if there are two ΔΣ modulators of the same order, the one designed to have a higher in-band SNR will probably produce more quantization error too. However, the modulator which produces

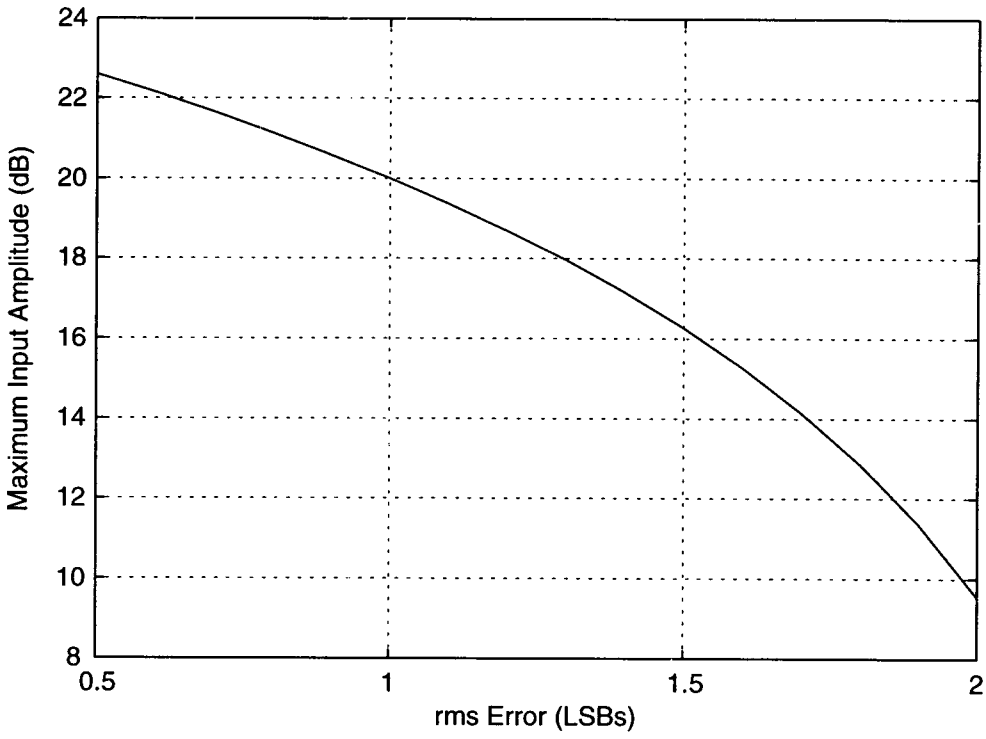


Figure 5.14: Maximum input amplitude VS. rms quantization error.

less total quantization noise tends to be more stable so that input signals with larger amplitudes could be allowed, and hence might yield an even higher peak SNR than that of the one with higher rms quantization errors. According to the empirical rule introduced in Section 5.2.1, the maximum peak-to-peak input amplitude is approximately $m - 7e_0$, where m is the number quantization steps and e_0 is the rms output error given a 1 LSB peak-to-peak input. The contribution to the peak SNR from this effect is then $20\log_{10}(m - 7e_0)$ dB which is plotted as a function of e_0 in Figure 5.14, given that $m = 17$ for this design example.

By combining Figure 5.14, Figure 5.2 and Eq. (5.1) together, it is possible to determine an optimum (or a nearly optimum) point which achieves a large peak SNR with small rms error. Table 5.4 summarizes the optimal points for the three modulators in this

Table 5.4: "Optimum" points in the design example.

Modulator Spec. (Order, OSR)	e_0 (LSBs)	Estimated SNR (dB)	$\|H\|_\infty$ chosen
4, 8	1.5	76	6
8, 8	1.5	100	6
4, 16	1	100	3.5

design example. When choosing these optimal points, other limitations, such as mismatch error, nonlinearity, and other circuit noise, should also be considered so that there is a "reasonable" maximum SNR. In filling Table 5.4, maximum SNRs around 100 dB were considered, leaving some margin for other error sources.

5.5.2 Design of Lowpass Modulators

This section covers the design of the three lowpass modulators listed in Table 5.4. Some details of the designs can be found in Appendix I.

For the design of each modulator, a NTF with complex zeros (split to efficiently suppress the in-band quantization noise) is derived first using `synthesizeNTF`. Then the coefficients are derived based on the Cascade-of-Resonator structure with only FeedBack branches (CRFB) [2, 62]. After the original NTFs and coefficients are figured out, scaling is done to make all the state variables have the same dynamic range. Then the coefficients are quantized to simple combinations of powers of 2 so that no multipliers are needed to implement the modulator.

Both of the 4^{th}-order modulators use the same structure. The schematic is shown in Figure 5.15. Notice that the first resonator which implements the pair of complex zeros closer to DC has double delays which gives more time for the corresponding adders to do their jobs. Therefore, the zeros are shifted from the unit circle onto the line $Re(z) = 1$. This, however, does not affect the performance of the modulator significantly because the pair of poles are at low frequencies and still very close to the unit circle. The 4.5-bit quantizer/limiter quantizes and hard-limits the modulator output to 16 levels. Since the coefficients are so simple and only require moderate accuracy, the same design can be used for an analog modulator (i.e. an ADC) too.

Figure 5.16 illustrates the system diagram of the 8^{th}-order modulator optimized with $OSR = 8$. One of the purposes in designing such a high-order modulator is to show that, even with low OSRs, higher-order NTFs can achieve better SNRs.

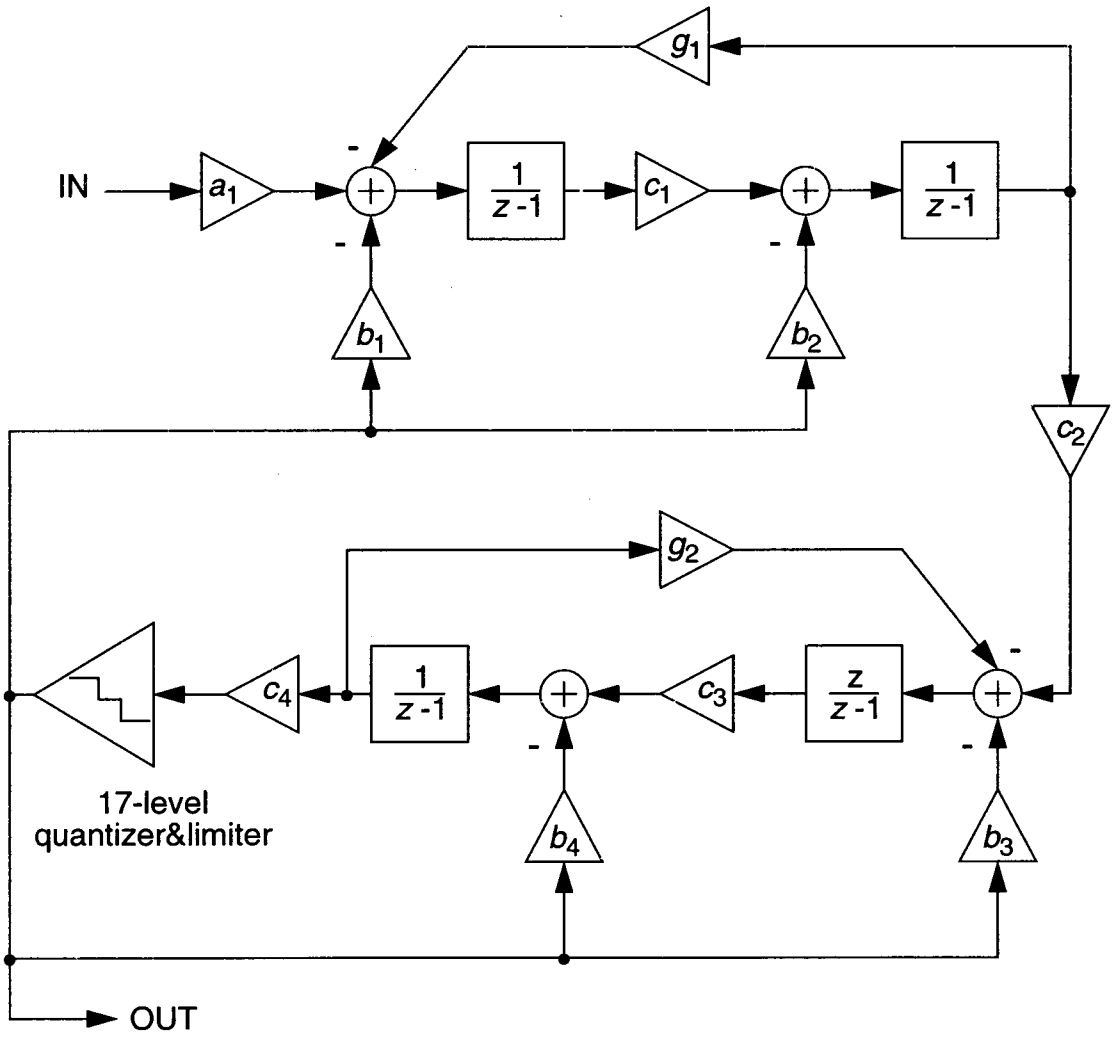


Figure 5.15:Structure of the 4^{th}-order modulator.

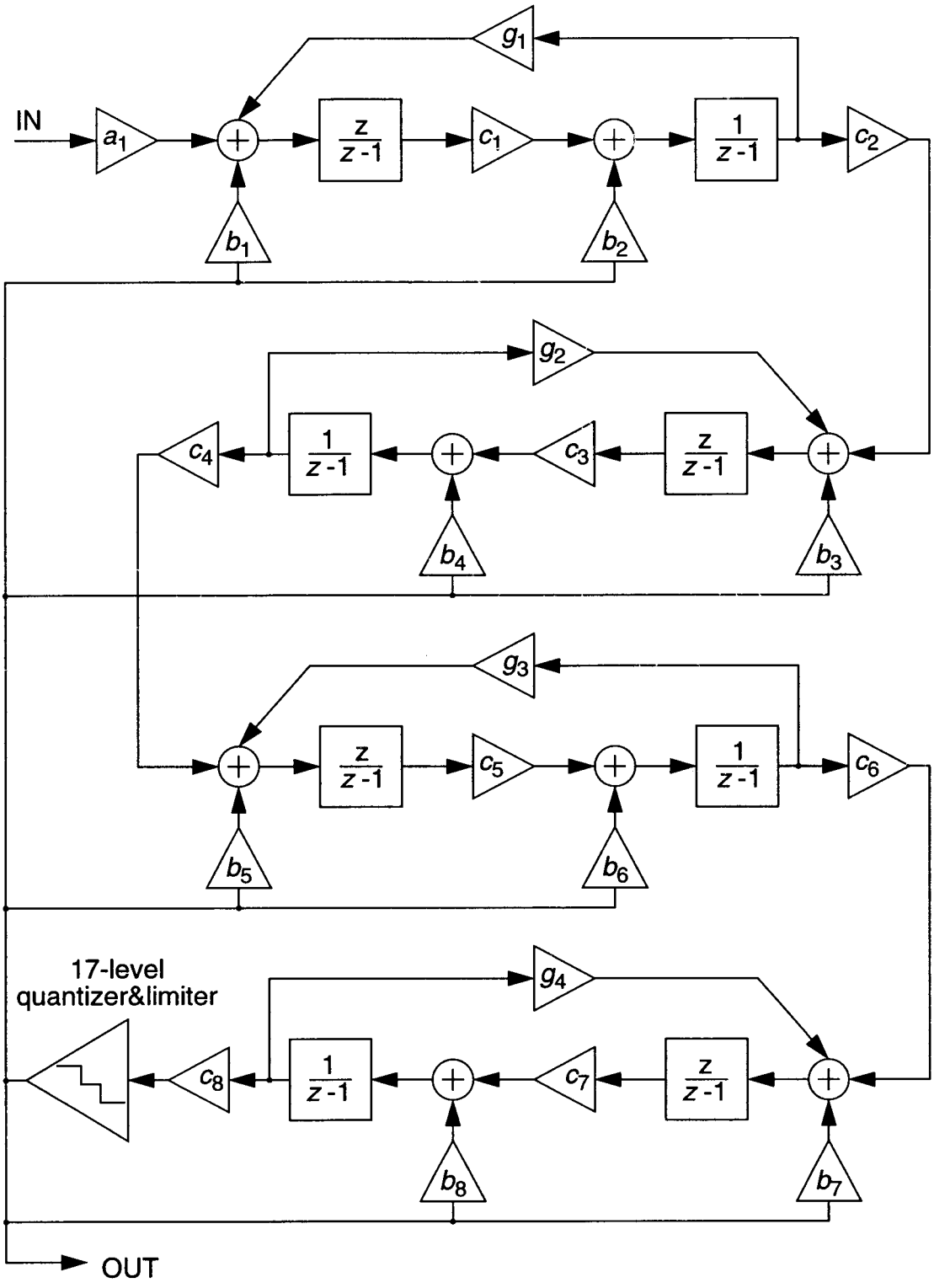


Figure 5.16:Structure of the 8^{th}-order modulator.

The quantized coefficients of the modulators are listed in Table 5.5, along with the peak SNRs obtained from the simulations.

Table 5.5:Coefficients and peak SNRs of the lowpass modulators.

	4^{th}-order, 8x OSR	4^{th}-order, 16x OSR	8^{th}-order, 8x OSR
a_1	2^{-5}	2^{-5}	2^{-6}
b_1	2^{-5}	2^{-5}	2^{-6}
b_2	2^{-5}	2^{-5}	$2^{-6}+2^{-8}$
b_3	2^{-3}	2^{-3}	2^{-5}
b_4	2^{-4}	2^{-4}	2^{-5}
b_5			2^{-4}
b_6			$2^{-5}+2^{-6}$
b_7			$2^{-3}+2^{-7}$
b_8			2^{-4}
c_1	2^{-1}	2^{-1}	$2^{-2}-2^{-4}$
c_2	2	2	2^{-1}
c_3	1	1	$2^{-1}-2^{-3}$
c_4	2^4+2^{-2}	2^4	1
c_5			$2^{-1}+2^{-3}$
c_6			$2+2^{-3}$
c_7			1
c_8			2^4+2^{-1}
g_1	2^{-5}	2^{-7}	2^{-5}
g_2	2^{-3}	2^{-5}	2^{-3}
g_3			$2^{-3}+2^{-5}$
g_4			$2^{-3}+2^{-6}$
Peak SNR	79 dB	107 dB	99 dB

Two simulation examples are shown in Figure 5.17 and Figure 5.18. Figure 5.17 shows the output waveform and the spectrum of the 4th-order modulator optimized with

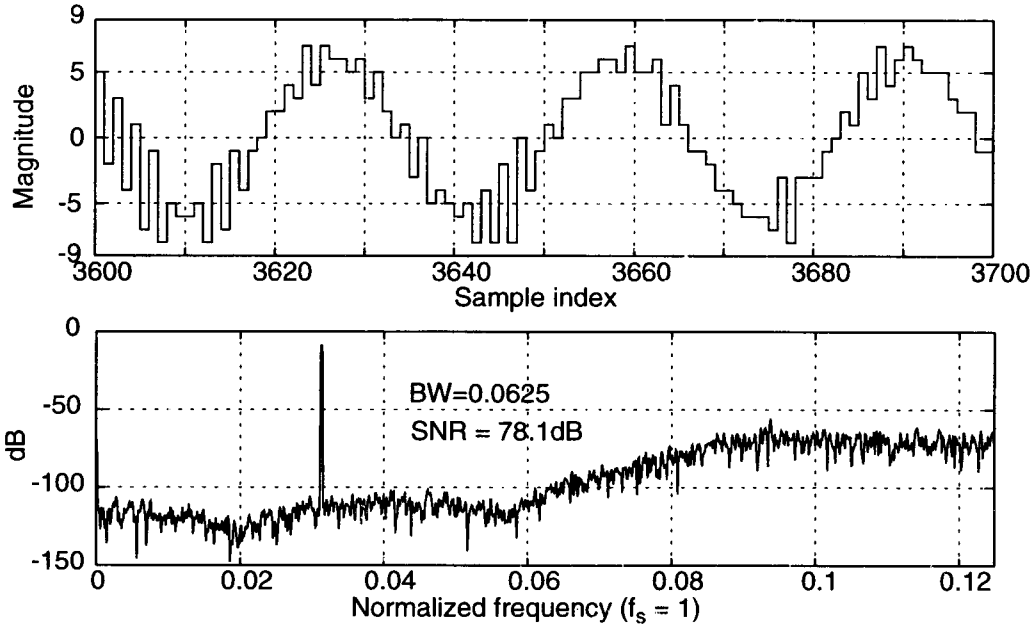


Figure 5.17:Output waveform and spectrum of the 4th-order modulator (*OSR* = 8).

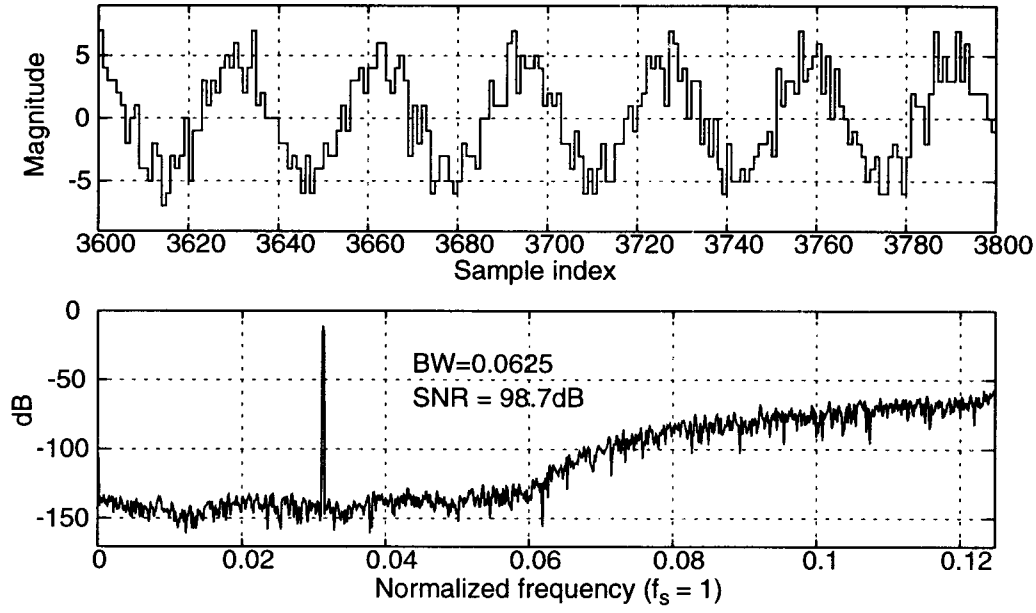


Figure 5.18:Output waveform and spectrum of the 8th-order modulator (*OSR* = 8).

$OSR = 8$ given a single-tone input located at the center of the band-of-interest with an amplitude of 0.72 of full scale. The two notches in the spectrum correspond to the two in-band zeros of the NTF. Figure 5.18 shows the 8^{th}-order modulator output waveform and the spectrum given a single-tone input located at the center of the band-of-interest with an amplitude of 0.58 of full scale. Both spectra are obtained by performing an 8192-point Hann-windowed FFT on the data from the time-domain simulations.

5.5.3 8^{th}-Order Bandpass Modulator

The lowpass prototype method was used to design the bandpass modulators. By applying the transformation $z \rightarrow -z^2$ to the two 4^{th}-order lowpass modulators described in Section 5.5.2, two 8^{th}-order bandpass modulators were derived. The zeros of the NTFs

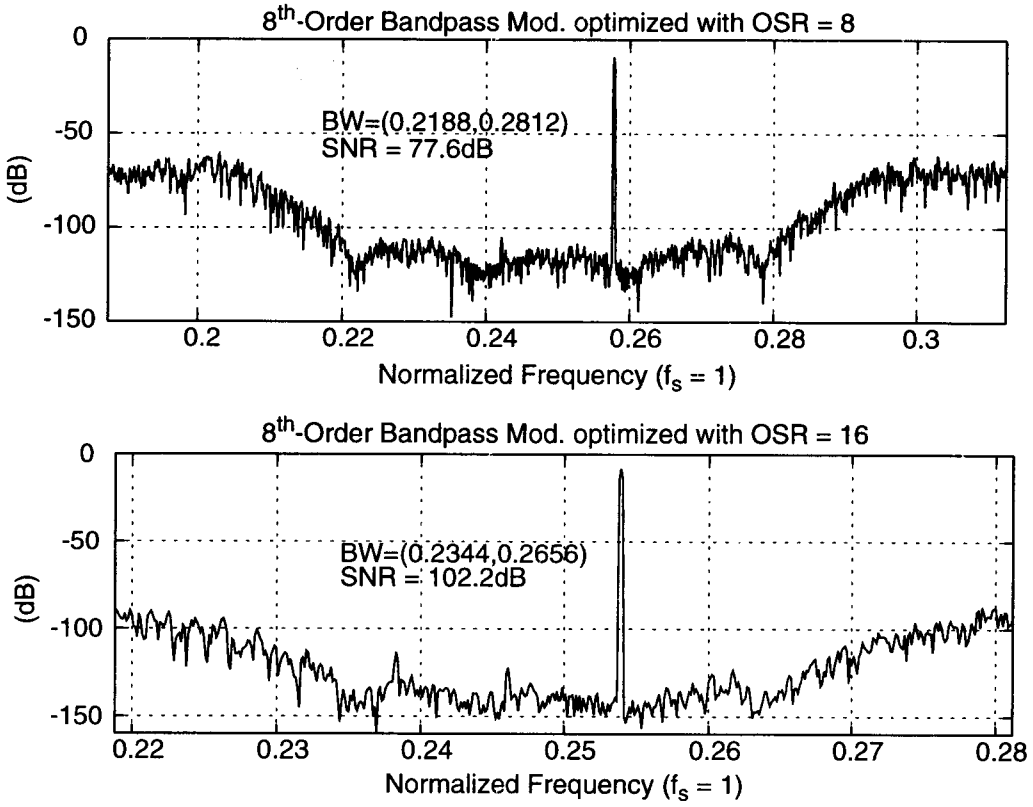


Figure 5.19:Output spectra of 8^{th}-order bandpass modulators.

are now centered at $\pm\frac{\pi}{2}$ (i.e. 1/4 of the sampling frequency) instead of dc. The transformation does not affect the dynamics of the prototype. The schematics can be obtained by simply changing all the z^{-1} blocks in Figure 5.15 to $-z^{-2}$ and leaving everything else the same. Two simulation examples are shown in Figure 5.19.

5.6 ESL and other Digital Logic

5.6.1 1st-Order ESL

As mentioned earlier, the other major digital block in the multi-bit ΔΣ DAC system is the element-selection logic, or ESL. The 1st-order ESLs described in Chapter 3 are used in the DACs to noise-shape the errors caused by the nonlinearity in the DAC elements. The ESLs take the 4-bit modulator output V and generate a 16-bit element selection vector SV which drives the 16 analog DAC elements (i.e. capacitors) by turning on/off the corresponding switches. For the bandpass modulators, the bandpass swappers described in Chapter 3 are used. Since the structure of the ESL has been covered in Chapter 3, it will not be discussed here.

Two simulation examples are shown in Figure 5.20 in the form of output spectra. The SC DAC structure of Figure 5.5 was used in the simulation, and an op amp gain of 60 dB was assumed. For the 4th-order lowpass modulator with $OSR = 8$, an input tone with a magnitude of 0.72 of full scale was applied to the system. With 0.5% uniformly distributed mismatch errors, the SNR of the DAC output is degraded from 77.2 dB in the ideal case (shown as the dotted line) to 62.9 dB without any mismatch-shaping (the dashed line), and the 2nd- and 3rd- order harmonics were about 60 dB and 70 dB below the signal level, respectively. With 1st-order mismatch-shaping (the thick solid line), the SNR is improved to 76.1 dB, the 3rd-order harmonic is knocked down to 90 dB below the signal level and the 2nd-order harmonic almost vanishes into the noise floor. The 2nd-

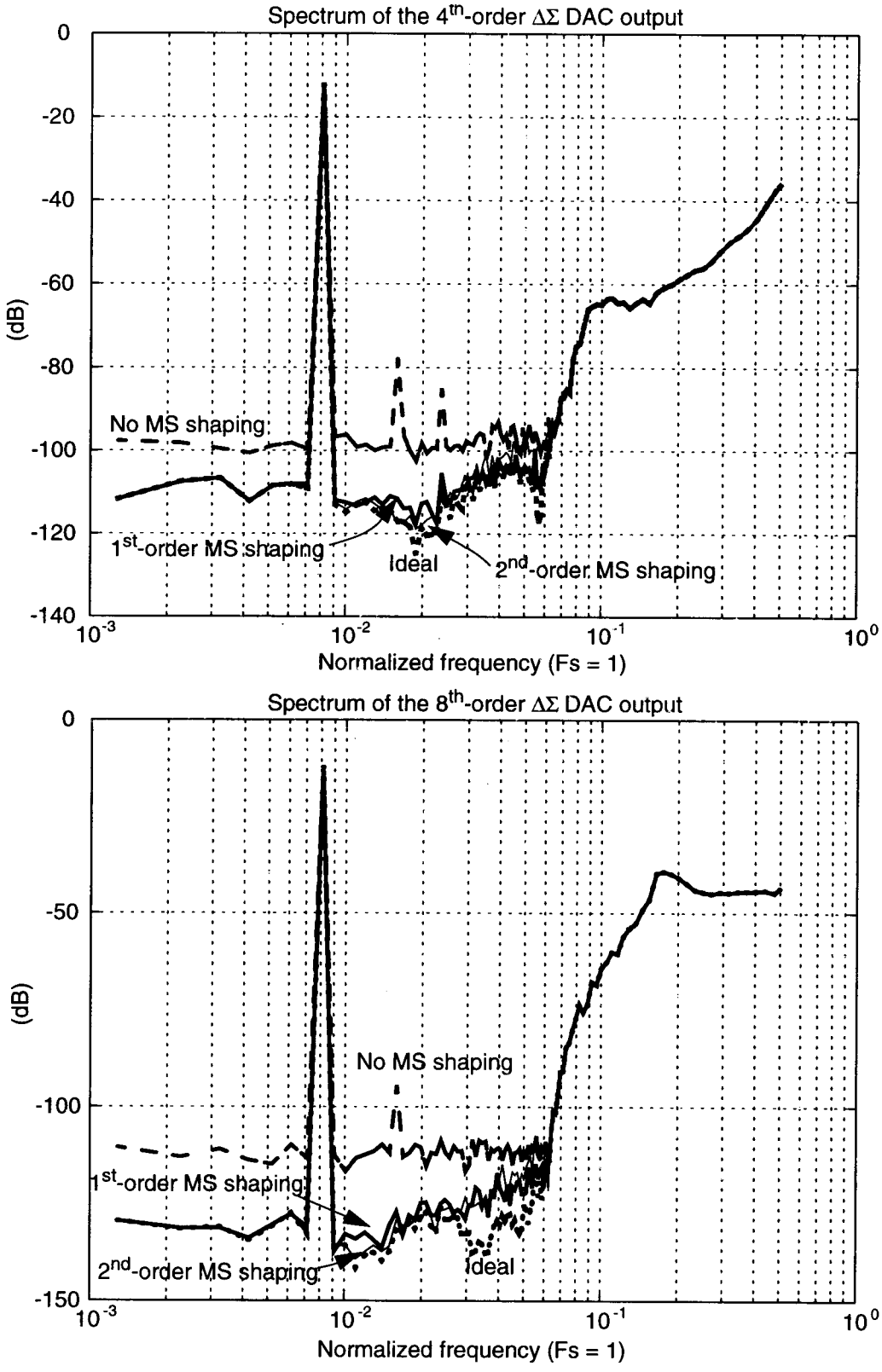


Figure 5.20:Output spectra of the mismatch-shaping ΔΣ DAC.

order mismatch-shaping (the thin line), however, does not help more than the 1st-order does. The noise floor with 2nd-order mismatch-shaping is a little lower at the low frequency end, and so are the harmonics. But the overall SNR is only 72.6 dB which is lower than that of the 1st-order. For the 8th-order lowpass modulator with $OSR = 8$, the results are similar. With 0.1% mismatch and input magnitude of 0.58 of full scale, the SNRs are 96.3 dB in the ideal case, 78.8 dB without mismatch-shaping, 91.4 dB with 1st-order mismatch-shaping, and 88.2 dB with the 2nd-order mismatch-shaping. In both simulations, the input tones were located at 1/8 of the band-of-interest.

From the above simulation results, it is reasonable to conclude that the 2nd-order, and hence even higher order mismatch-shaping schemes are not suitable for low OSRs. The situation is analogous to that of high-order single-bit $\Delta\Sigma$ modulators in that such systems do not out-perform the simple 1st-order modulator at low oversampling ratios, unless optimizations are done to distribute the zeros through the band-of-interest. Considering the complexity involved, implementing such high order mismatch-shaping ESLs is not practical with today's technology.

5.6.2 Pseudo-Random Scrambler

As mentioned in Section 5.4.3, before the 16-bit *SV* vectors are transferred to the analog SC DAC, each bit is scrambled randomly by a pseudo-random bit-stream *scr* so that the signal content in sv_i which might otherwise couple into the power supplies [2] is destroyed. The randomization is done by taking the exclusive-OR combination of each *sv* bit and *scr* (i.e. $sv_i \oplus scr$) at the output of the digital ESL. The same operation (as illustrated by Eq. (5.5)) is done on the analog DAC side so that the random bit-stream is descrambled and the *sv* bits recovered.

A feedback shift register [63] is used to generate the pseudo-random bit sequence *scr*. As shown in Figure 5.21, an exclusive-OR gate generates the serial input signal for a shift register of length 17 bits. The two input bits of the exclusive-OR gate are the 14th and 17th bits of the shift register. The circuit goes through all 2^{17} states except the zero state before repeating.

5.7 Summary

Designs of some high-order digital modulators with mismatch-shaping ESLs and a 16-element SC DAC IC were described. The DAC was designed in the Orbit 1.2 μm double-poly CMOS process. Relatively low OSRs were used in the design. The purpose is to demonstrate the effectiveness of element mismatch-shaping technology in making highly linear DACs using high-order wide-band multi-bit ΔΣ modulators.

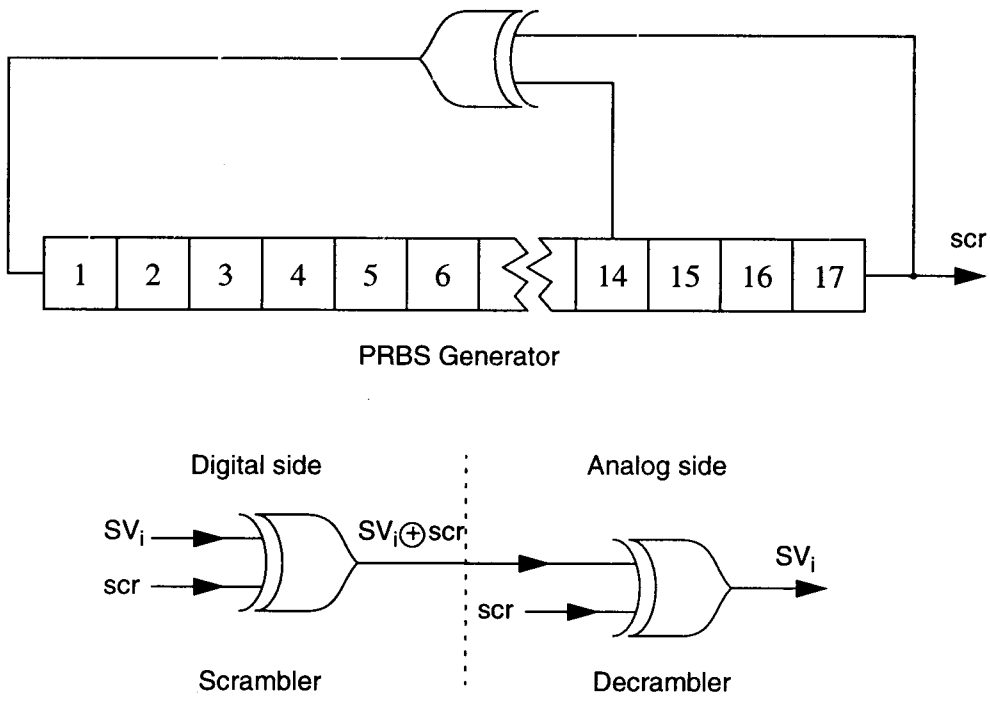


Figure 5.21:Scrambling and decrambling *SV* bits.

Chapter 6. Experimental Results on the Mismatch-Shaping SC DAC

In Chapter 5, a 16-element SC mismatch-shaping DAC was designed. The SC DAC core was fabricated in the Orbit 1.2 μm double-poly double-metal process. The 4^{th}-order 16-times oversampling lowpass delta-sigma modulator designed in Chapter 5 was implemented on a Xilinx 4000E series FPGA along with a 1^{st}-order butterfly shuffler, a 1^{st}-order ESL and a 2^{nd}-order ESL. Using the lowpass prototype, a bandpass system was also derived and implemented on the FPGA with a bandpass butterfly shuffler and a bandpass ESL. The experimental results are given and discussed in this chapter.

6.1 The SC DAC Chip

The photo of the SC DAC chip is shown in Figure 6.1. As described in the last chapter, the chip contains a 16-element SC DAC core, a digital interface, a clock generator

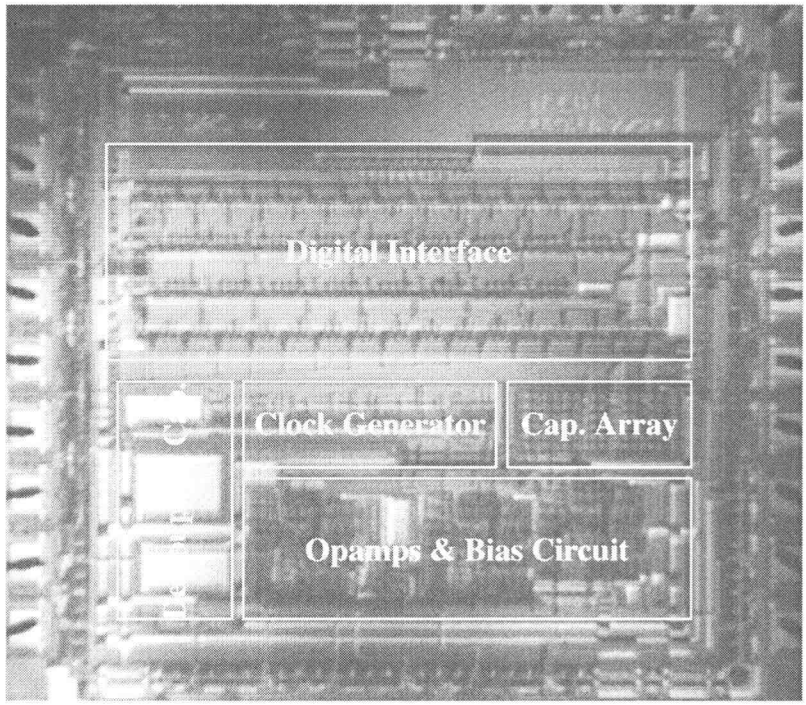


Figure 6.1: Chip photo.

and some test cells. The die size is 1900 x 1900 μm^2 and the 1260 x 1260 μm^2 chip area accommodates about 2400 transistors.

6.2 The Test Bed

A test bed was built to test the mismatch-shaping SC DAC. As shown in Figure 6.2, it includes two PCBs: a Xilinx FPGA application board which was made by Forrest Hudson [49], and a DAC board which accommodates the prototype SC DAC and its supporting circuits. The test signal is a 16-bit oversampled half-scale digital sine wave which is stored in a pair of EPROMs on the FPGA board. The delta-sigma modulators and the ESL circuits described in the previous Chapters were implemented on the Xilinx FPGA (XC 4005E or XC4010E) which in turn drives the 16-element SC DAC. The DAC output spectra were measured using an HP3585B spectrum analyzer.

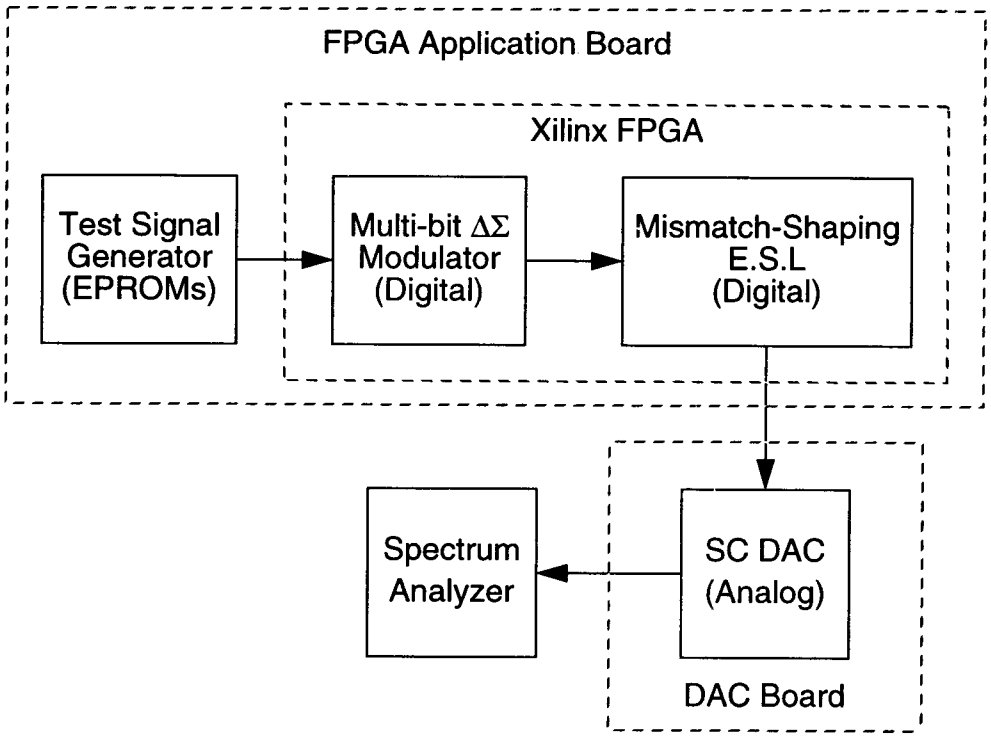


Figure 6.2: The mismatch-shaping SC DAC test bed.

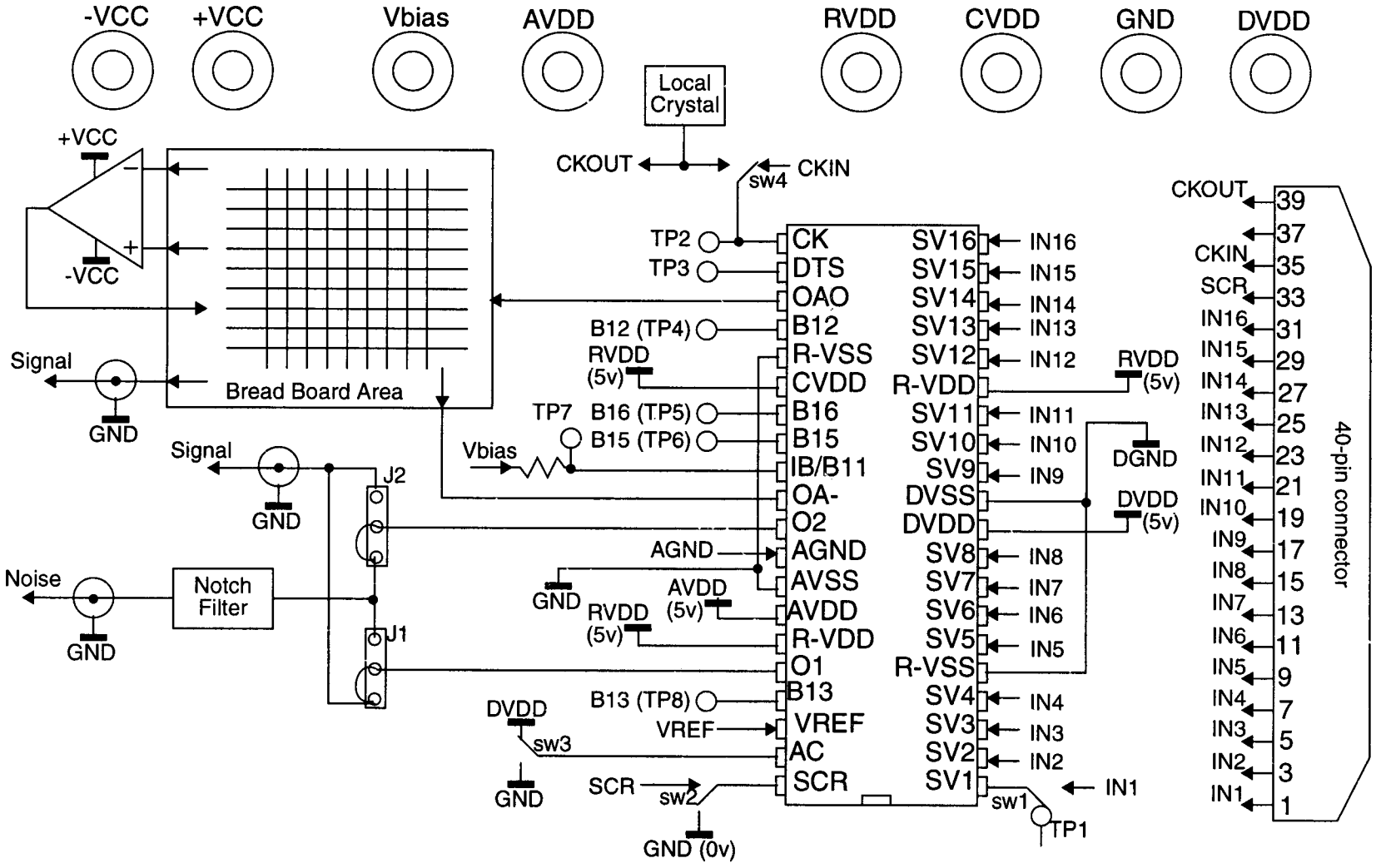


Figure 6.3: The DAC board.

The schematic of the DAC board is shown in Figure 6.3. The DAC board is connected to the FPGA board through a 40-pin connector which transfers the data between the two boards. The *sv* pins of the DAC are all pulled to ground (the pulldowns are omitted from the schematic) when the connector is floating. Five sets of 5-V power supplies are used. They are DVDD (digital VDD), AVDD (analog VDD), RVDD (ring VDD), CVDD (clock VDD) and Vbias (which is used to generate reference and bias current). The clock signal can be either generated on the DAC board or taken from the FPGA board through the connector. A bread board area is put on the board to construct additional test circuits. A set of power supplies (+VCC and -VCC) are available to these supporting circuits.

6.3 Op Amp Test

The major op amp parameters were measured. The test setups and the results are described in this section.

6.3.1 Offset Voltage

The op amp offset voltage test circuit is shown in Figure 6.4 [64]. The results from three sample chips are 1.00 mV, 1.02 mV and 0.90 mV. Therefore, the op amp offset voltage is around 1 mV which is fairly good.

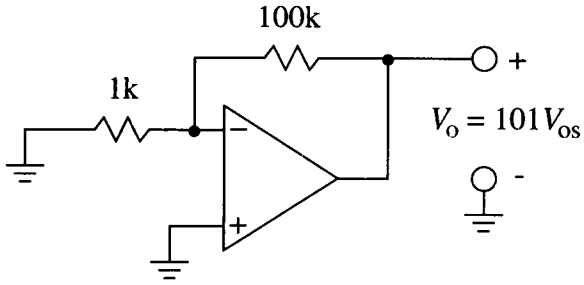


Figure 6.4: V_{os} test circuit.

6.3.2 DC Open-Loop Gain

The open-loop gain of an op amp is an important parameter to measure. By using the device under test (DUT) inside a feedback loop, it is possible to measure the change in input voltage required to produce a known change in output voltage. Figure 6.5 shows one of such circuits [64]. In this circuit, the control voltage, V_c, is varied from -1.8 V to -3.2 V, causing the DUT output to vary from 1.8 V to 3.2 V (neglecting the offset). Since V_{in} is attenuated from E_o (which can easily be measured) by the R_F/1kΩ voltage divider, the open-loop gain can readily be computed. The measured open-loop gain of the op amp is around 15,000 (83 dB).

6.3.3 Unity-Gain Bandwidth

The unity-gain bandwidth *GBW* was measured using the closed-loop configuration shown in Figure 6.6. A 10 mV sine wave was used as the input signal. The frequency was swept to determine *GBW*. With the additional compensation capacitor

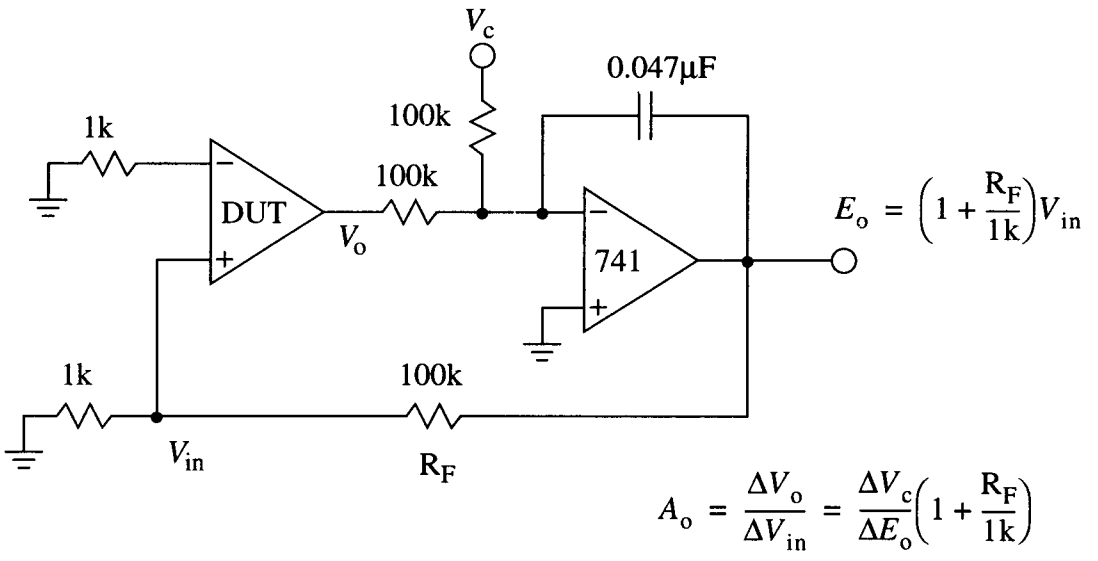


Figure 6.5: A_o test circuit.

turned on (AC pin set to 5 V), the measured unity-gain frequency is 7 MHz. Without the additional compensation, the unity-gain frequency is 9 MHz.

6.3.4 Step Response

Figure 6.7 shows the circuit used to measure the step response and the results of that measurement. The op amp is connected as an unity-gain buffer to maximize the feedback and thus minimize the phase margin. The input signal is a 260 kHz square wave. As illustrated by the plot of Figure 6.7, the settling curve has a nonlinear slow-rising portion at the beginning, although no obvious slewing is observed.

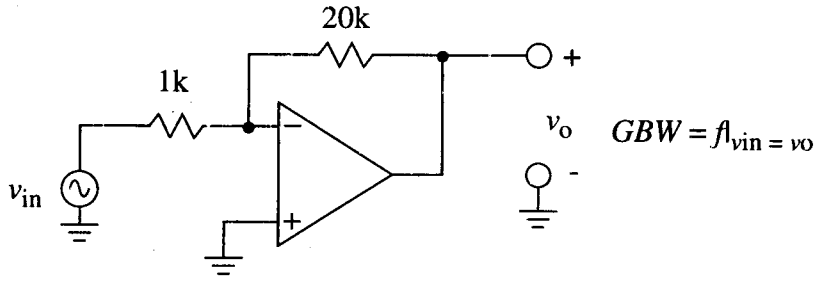


Figure 6.6: *GBW* test circuit.

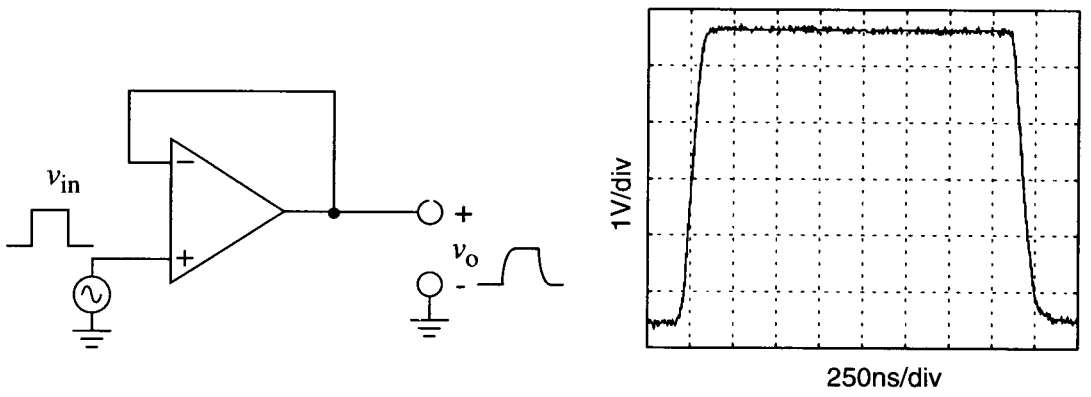


Figure 6.7: The step response of the op amp.

6.4 Element Matching

The SC DAC was connected to the FPGA in which the modulators and ESL circuits were implemented. The DAC output spectra were measured using a spectrum analyzer, and compared with the simulations. Second-order mismatch-shaping was demonstrated.

6.4.1 1st-Order Mismatch-Shaping DAC

In Chapter 5, a 4th-order lowpass modulator was designed and optimized for OSR = 16. The modulator was implemented on a XC4008E FPGA as part of the system under test. Two unit-element mismatch-shaping schemes, namely the element-rotation (i.e. 1st-order delta-sigma ESL) and the 1st-order butterfly shuffler, were used to demonstrate 1st-order mismatch-shaping. Both mismatch-shaping schemes were also implemented on the FPGA. The input test signal was a half-scale 16-bit sine wave oversampled by 64 times (i.e. 128 sample points in one period). The tests were performed at 500 kHz clock frequency, and the signals were measured at O_2 (after the de-glitching filter).

As shown in Figure 6.8, at 500 kHz, without any mismatch-shaping effort, the 2nd-order and the 3rd-order harmonics are both 73 dB and 69 dB below the signal tone, respectively. When the element-rotation logic is turned on, the 2nd-order harmonic drops 13 dB to 86 dB below the signal level, and the 3rd-order harmonic vanishes into the noise floor. When the 1st-order butterfly shuffler is used, both the 2nd-order and the 3rd-order harmonics disappear. The observation that the butterfly shuffler produces less harmonic distortion than the delta-sigma ESL does in the 1st-order case was discussed in Chapter 3. The noise floors in both shaping cases are the same, dropping 4 dB from the no-shaping case and resulting in an SNR of 70 dB assuming a signal band from dc to 15 kHz.

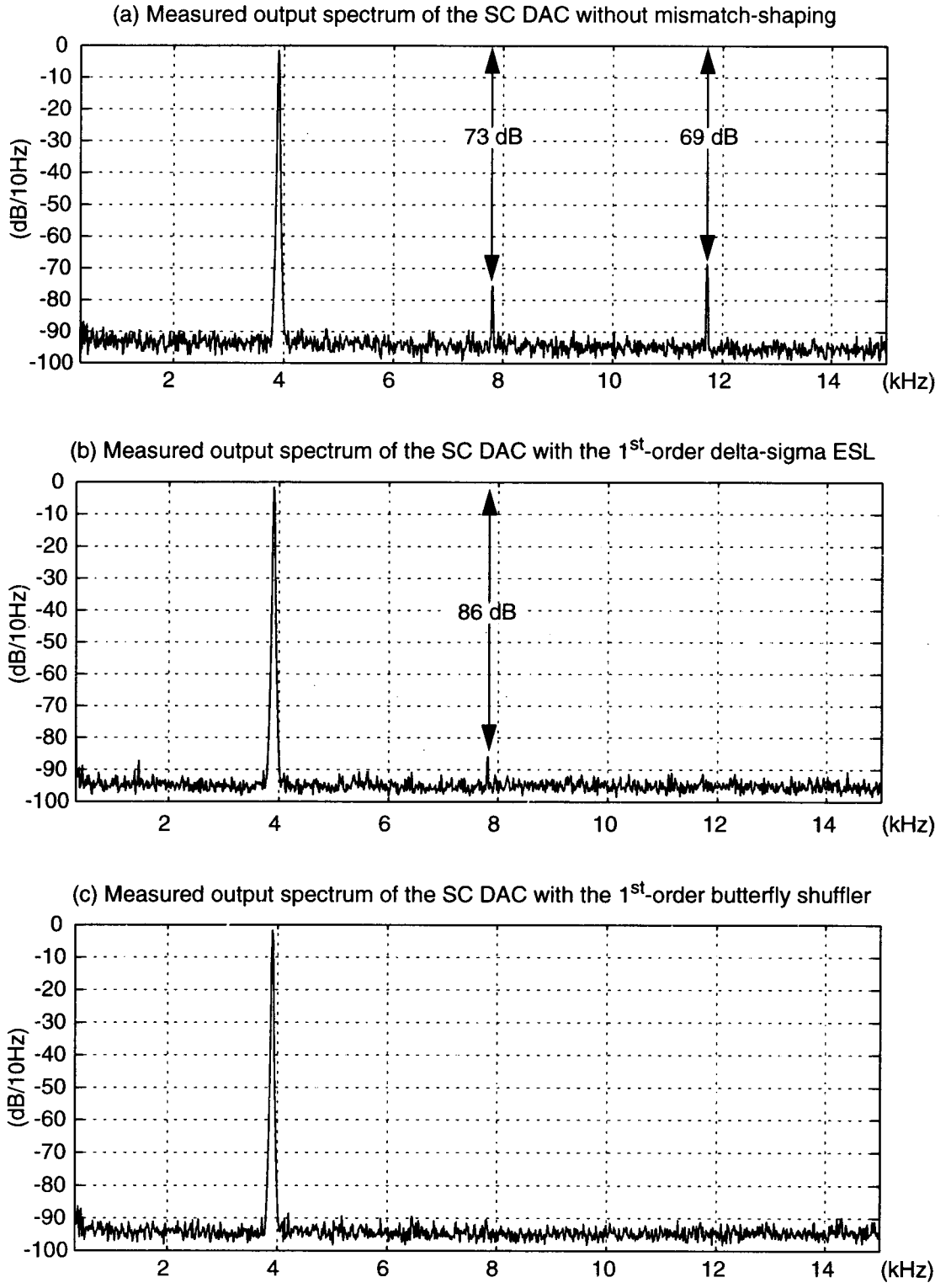


Figure 6.8: Measured spectra of the 4th-order delta-sigma SC DAC with 4-bit quantization at a sampling frequency of 500 kHz.

To verify the results, simulations were done given the measured DAC element values. The spectra are plotted in Figure 6.9 by performing a 16384-point Hanning-windowed FFT on the time-domain simulation results. Calculations on the simulation data indicate that the 2nd-order harmonics are 88 dB and 104 dB below the input signal for the element-rotation and the butterfly shuffler, respectively. Notice the shaping difference between the noise floors of the two simulated spectra. The simulated SNR of the element-rotation is 84 dB which is 3 dB better than that of the butterfly shuffler. As discussed in Chapter 3, this fact is probably due to the limitations on the element usage patterns caused by the particular way that the butterfly shuffler is wired, which may produce larger "selection errors" than that of a competing delta-sigma ESL. However, the measurement results are limited by error sources other than the element mismatch and do not exhibit the shaping on the in-band noise floor.

6.4.2 2nd-Order Mismatch-Shaping DAC

A 2nd-order delta-sigma ESL was also designed and implemented. Due to the complexity of the algorithm, the 2nd-order delta-sigma ESL was designed to drive an 8-element DAC instead of a 16-element DAC, and a larger FPGA, XC4010E was used. The same 4th-order delta-sigma modulator as the one used in the 1st-order mismatch-shaping demonstrations was used after some minor modifications to perform 9-level quantization. The entire design (the 2nd-order ESL and the 4th-order modulator) consumed 399 out of the 400 CLBs in the XC4010E. In order to do comparisons, a 1st-order ESL driving 8 elements was also implemented. Only half of the SC DAC was used in this part of the test.

The 2nd-order ESL is illustrated in Figure 6.10. Note that only 8 of the 16 elements are used. Each sv bit is generated by an error-feedback noise-shaping loop which is very similar to a delta-sigma modulator. The MTF is chosen to be $H_2 = (1 - z^{-1})^2$. The

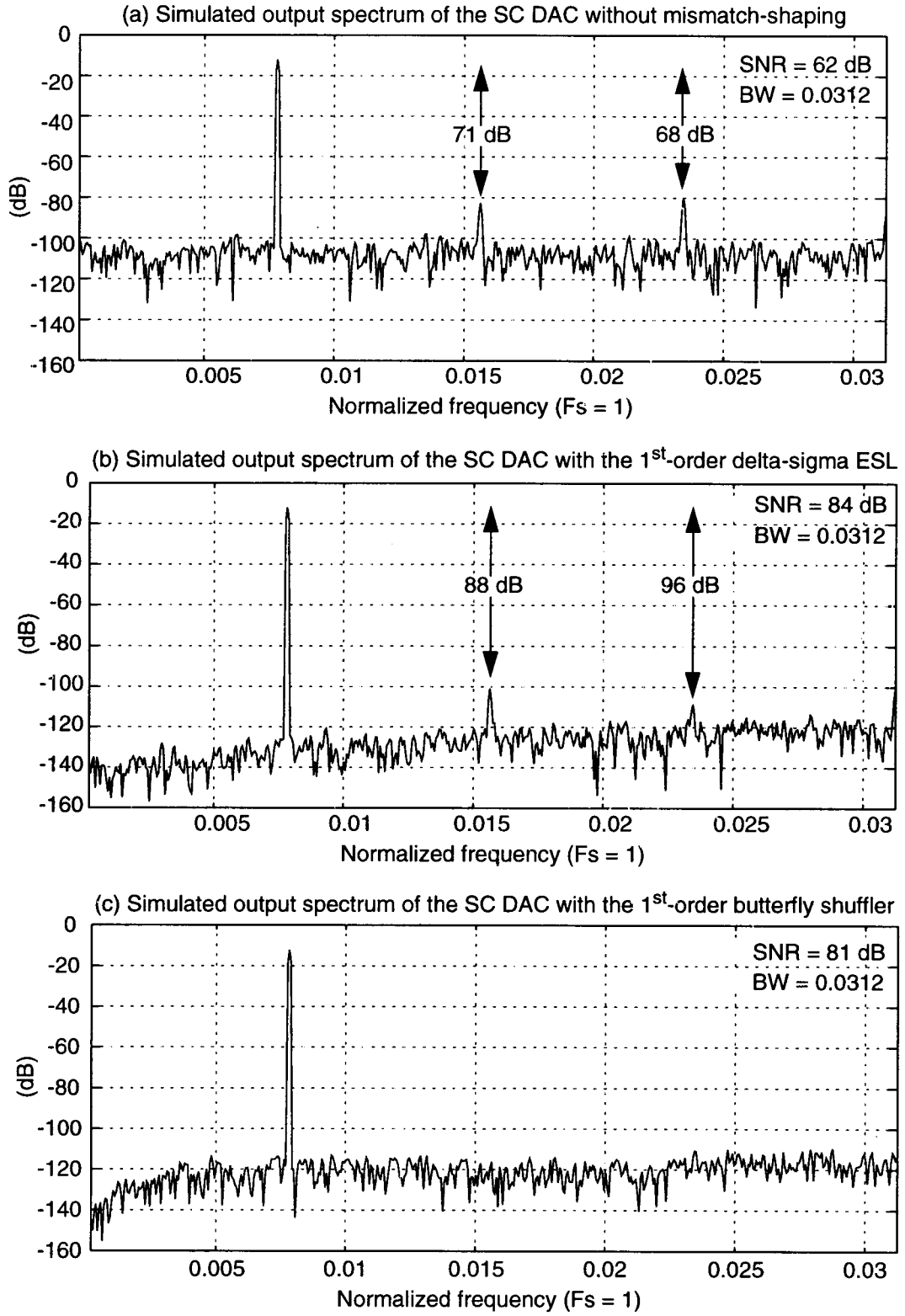


Figure 6.9: Simulated spectra of the 4th-order delta-sigma SC DAC with 4-bit quantization at a sampling frequency of 500 kHz.

output of the block min() is the smallest of its 8 5-bit inputs. The most complicated part is the vector quantizer, which completely sorts 8 3-bit numbers (i.e. $sy_1 \sim sy_8$) in a single clock cycle. The structure of the vector quantizer was also presented in Chapter 3. Aside from its complexity, the vector quantizer also has a large gate delay. Therefore the ESL can only run at relatively low speeds. Please refer to Appendix II for a set of detailed schematics of the 2nd-order ESL.

The system was clocked at 500 kHz, and some measured spectra are plotted in Figure 6.11. Without mismatch-shaping, the 2nd-order and the 3rd-order harmonics are 59 dB and 83 dB lower than the signal, respectively. After turning on the 1st-order ESL, the 2nd-order harmonic is reduced to -84 dB, and the 3rd-order harmonic is reduced to -86 dB. When the 2nd-order ESL is introduced, both 2nd-order and 3rd-order harmonics are

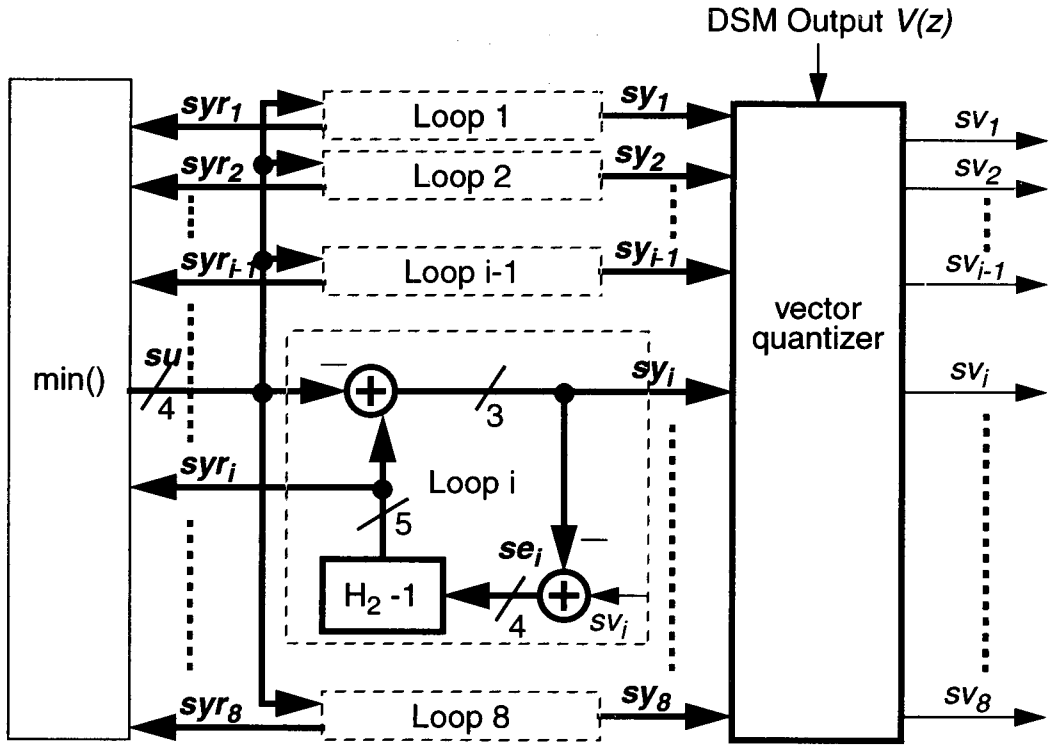


Figure 6.10: A 2nd-order 8-element ESL.

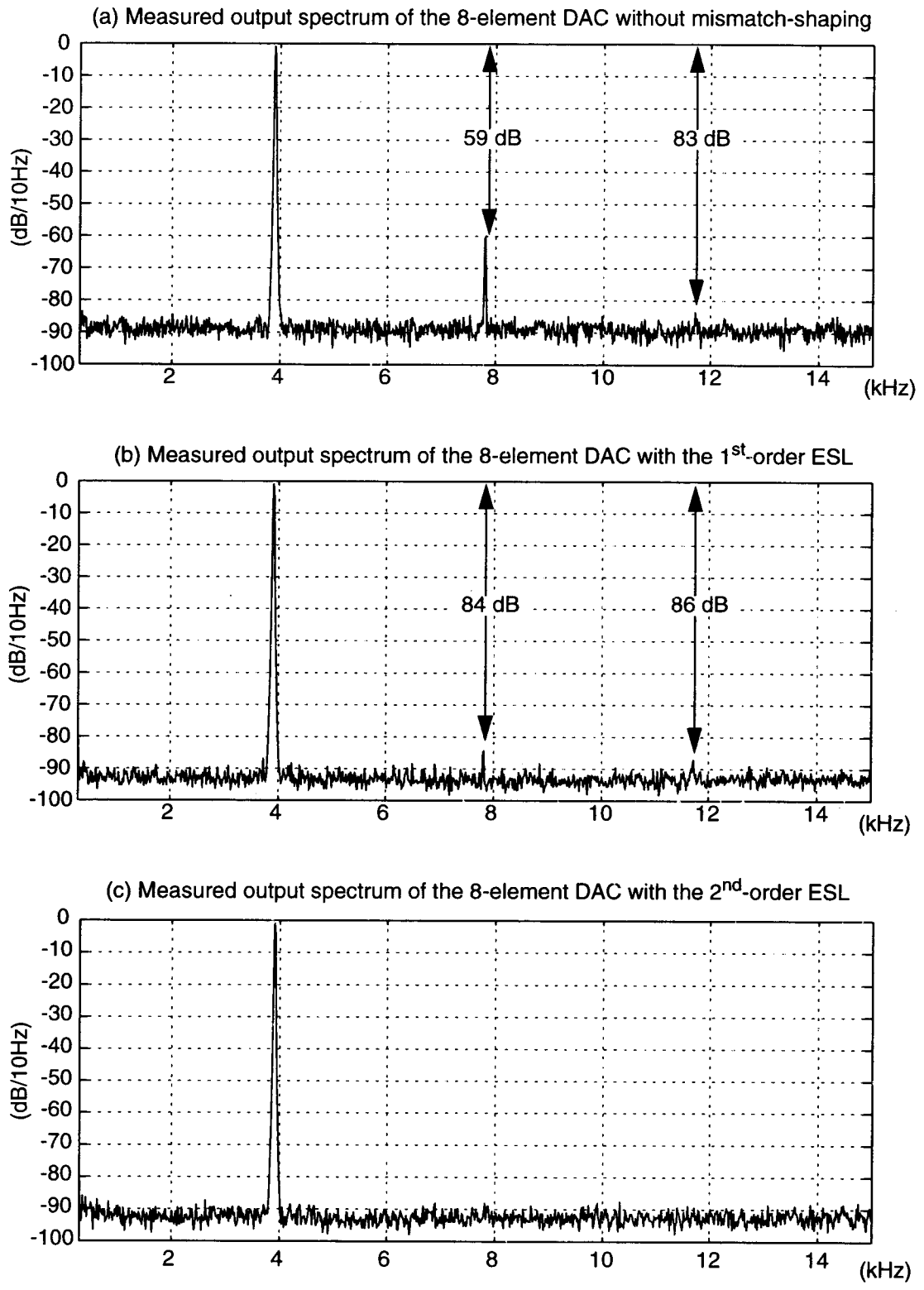


Figure 6.11:Measured spectra of the 4^{th}-order delta-sigma SC DAC with 3-bit quantization at a sampling frequency of 500 kHz.

gone. Further zooming into the noise floor did not distinguish them from the noise bins which are below the -90 dB level. Both of the ESLs attenuate the in-band noise power by 4 dB. The performance is then limited by noise sources other than the mismatch error. Therefore, the difference in the shaping of the mismatch noise is not apparent in these measurements.

In order to verify 2nd-order mismatch-shaping, the element mismatch was deliberately increased so that it became the dominant error source. Specifically, the value of Element 1 was doubled by connecting two unit elements together, and Element 5 was disconnected so that its value became zero. The results are shown in Figure 6.12. The output noise floor of the DAC driven by the 2nd-order ESL has a 40 dB/decade slope

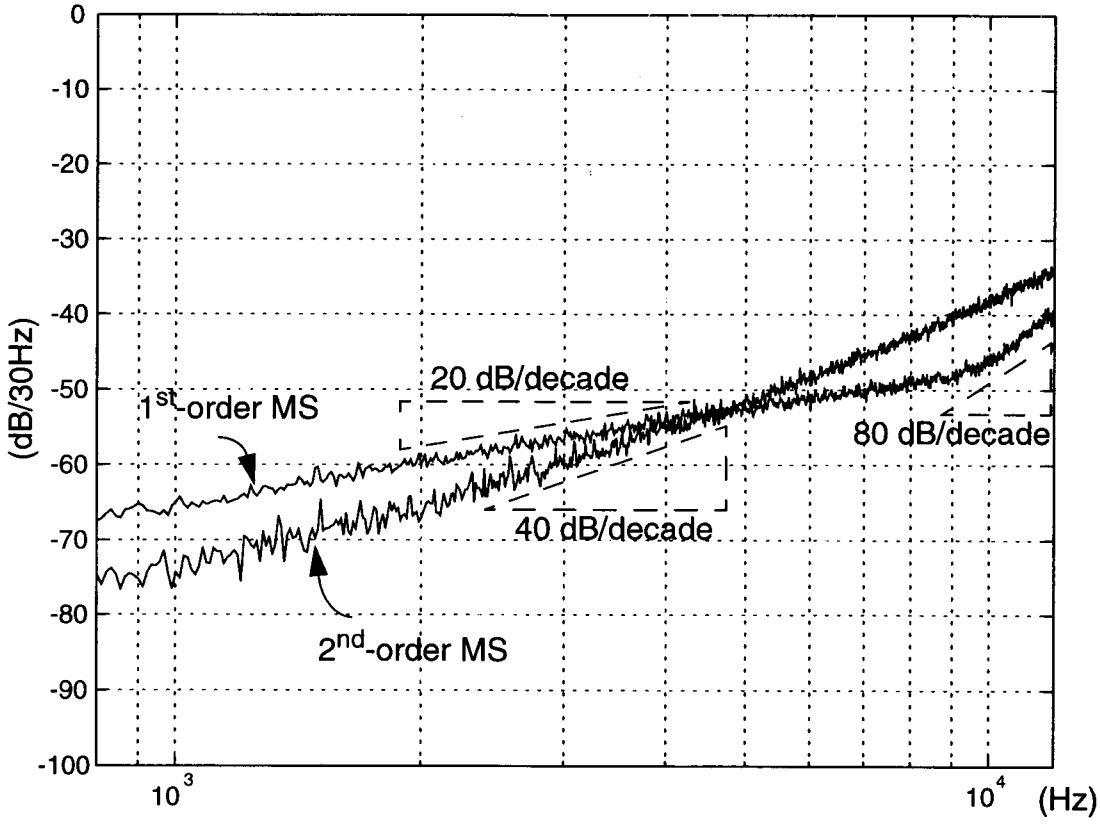


Figure 6.12:Demonstration of 1st-order and 2nd-order mismatch shaping.

which is in consistent with 2nd-order mismatch-shaping. When driven by 1st-order ESL, the slope starts at 20 dB/decade at the low frequencies, and ramps up to 80 dB/decade (which is the noise floor of the 4th-order modulator) when quantization noise kicks in. As indicated by the measured spectra, the 2nd-order ESL suppresses mismatch error more effectively than 1st-order ESL does at low frequencies, but produces more error at high frequencies.

6.4.3 Bandpass Mismatch-Shaping DAC

Using the lowpass prototype, an 8th-order bandpass modulator was derived by applying the $z \rightarrow -z^2$ transformation. Two bandpass mismatch-shaping schemes: the 2nd-order bandpass butterfly shuffler as described in 3.3.5 and a bandpass ESL, were used to shuffle the modulator output and drive the DAC. The center frequency is $f_s/4$. Again, the system was implemented on an XC4008E.

As described in Section 3.3.5, the bandpass butterfly shuffler has an MTF of $1+z^{-2}$. The bandpass ESL was implemented with an MTF $1-z^{-4}$ by simply substituting 4 delays for the single delay stage in the 1st-order lowpass ESL. Both MTFs have a zero at $f_s/4$ and hence give 1st-order mismatch-shaping at the nearby frequencies.

Because the center frequency is $f_s/4$, the output was measured at O_1 (i.e. before the discrete-to-continuous lowpass filter). Figure 6.13 shows the measured spectra at f_s = 500 kHz. Since the signal is near $f_s/4$, the even-order harmonics are outside the band-of-interest. However, some odd-order harmonics fall inside the band. The measured harmonic distortions are summarized in Table 6.1. The term "N/M" which stands for "not measurable" is used when a tone merges into the noise floor and cannot be distinguished from the noise bins. As indicated by the measurement, both bandpass mismatch-shaping

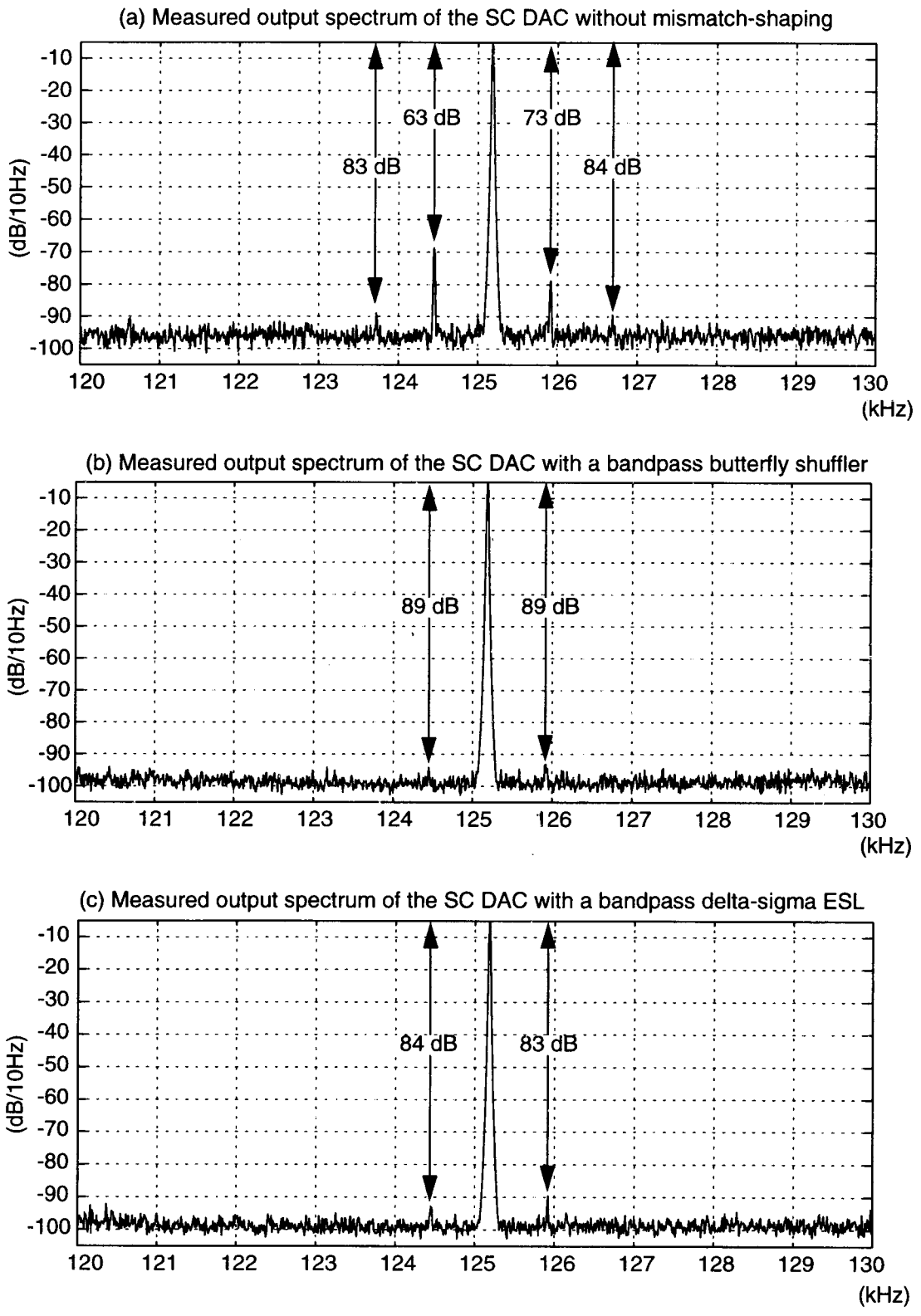


Figure 6.13:Measured spectra of the 8th-order bandpass delta-sigma SC DAC with 4-bit quantization at a sampling frequency of 500 kHz.

schemes reduce the harmonic distortions greatly. The butterfly shuffler works slightly better than the ESL. This is because the MTF of the bandpass ESL has two extra zeros at DC and $f_s/2$, which introduce a gain factor of approximately 2 near $f_s/4$, and therefore degrade the performance by approximately 6 dB in that region. The noise floor decreases by 4 dB when either of the mismatch-shaping schemes is turned on. Since the noise level is then dominated by other error sources, the SNR difference between the two mismatch-shaping schemes is not measurable.

Table 6.1:The measured harmonic distortion of the bandpass DAC.

	HD_3	HD_5	HD_7	HD_9
No-shaping	-63 dB	-73 dB	-83 dB	-84 dB
Bandpass butterfly shuffler	-89 dB	-89 dB	N/M	N/M
Bandpass ESL	-84 dB	-83 dB	N/M	N/M

6.5 Some Circuit Level Considerations for Mismatch-Shaping SC DAC

In this section, starting with the measurements of 1st-order mismatch-shaping at a 2 MHz clock rate, a few important circuit issues in designing high-performance mismatch-shaping SC DACs are discussed.

6.5.1 Mismatch-Shaping at 2 MHz Clock Rate

Figure 6.14 shows the measured output spectra at a 2 MHz clock rate with the 4th-order lowpass modulator and the two 1st-order mismatch-shaping schemes. The measurements were done at O_2 (after the de-glitching filter). As shown by the plots, the performance of the mismatch-shaping systems is almost identical. The 2nd- and the 3rd-order harmonics are 77 dB and 83 dB below the signal, decreasing by 13 dB and 16 dB,

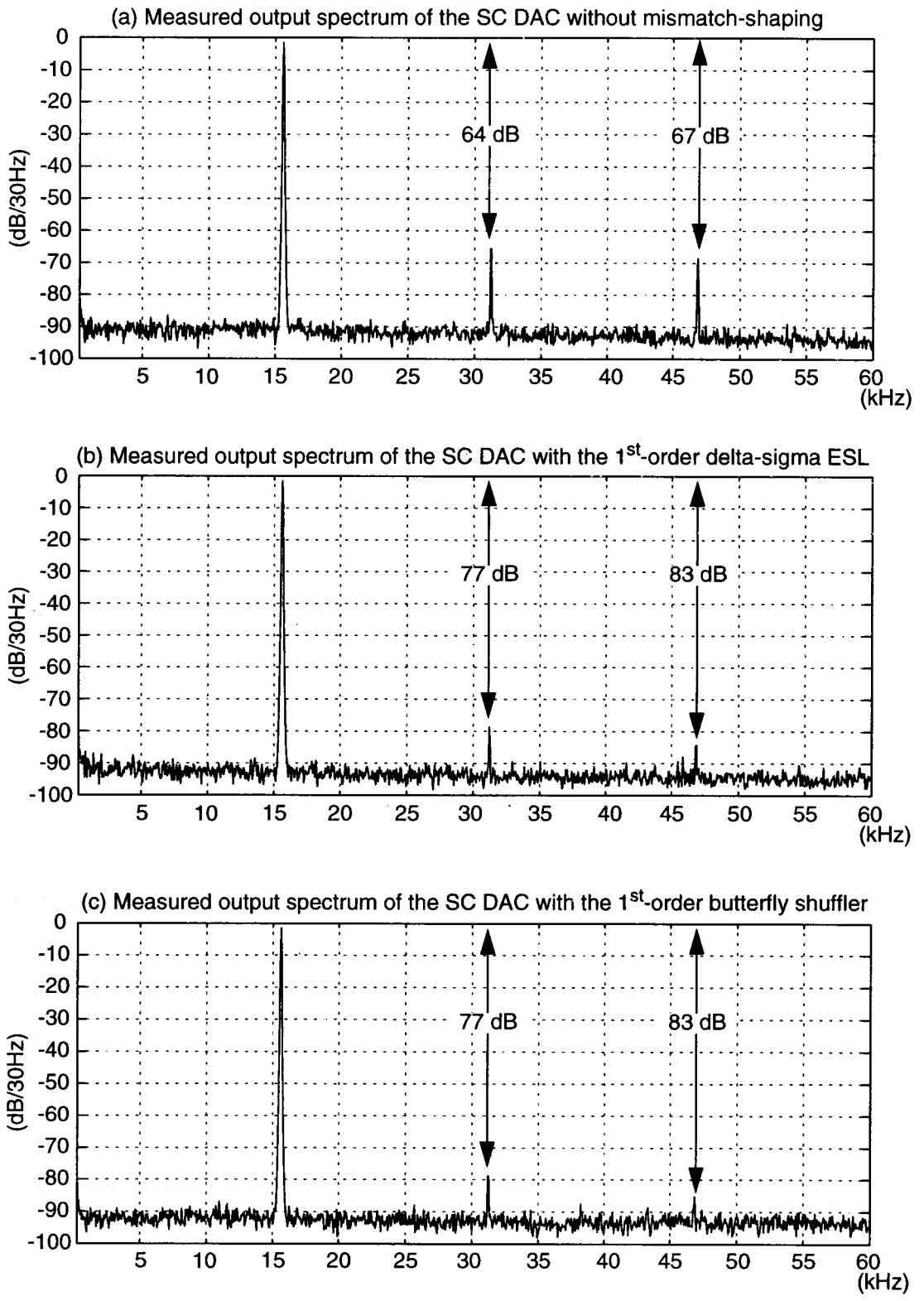


Figure 6.14:Measured spectra of the 4th-order delta-sigma SC DAC with 4-bit quantization at a sampling frequency of 2 MHz.

respectively, from the non-shaping case. However, according to the DC measurements and simulations, these numbers should be very much the same as those measured at 500 kHz, namely about 10 dB lower. The degradation is due to the nonlinearity in the output stage which will be discussed in the following sections. As the clock rate increases, the degradation gets big, until mismatch-shaping finally fails at 3 MHz.

In both shaping and non-shaping cases, the noise densities are 3 dB better than the corresponding data obtained at 500 kHz clock rate. This fact suggests that there is less circuit noise in the high frequencies. Given a band-of-interest from dc to 56 kHz, the SNRs in both shaping cases are approximately 66 dB which is still limited by noise sources other than element mismatch.

6.5.2 Effect of Incomplete Settling

As the clock rate is increased, incomplete settling will eventually occur. Even if a SC DAC circuit exhibits linear settling at any given input, incomplete settling may still cause a problem because the output RC time-constant may depend on the input code. Superposition will then fail, and so will mismatch-shaping since it relies on superposition. Therefore, when designing a mismatch-shaping SC DAC, one should avoid using structures which have signal-dependent load.

Fortunately, the implemented DAC (refer to Figure 5.5) does not have this problem. Since all the input branches are always connected to the feedback loop, the feedback factor remains constant no matter how many branches are selected. Therefore the equivalent load is signal-independent. As long as linear settling holds, superposition will also hold.

Although, by itself, incomplete linear settling is harmless, it is still not desirable, since clock jitter can then make the circuit very noisy.

If the transient response is not a linear function of the signal, incomplete settling results in distortion. This error term may be tolerated if the nonlinear settling, (in many case, slewing) only occupies a very short period in the transient response. Otherwise, the DAC must be given enough time to settle so that a mismatch-shaping scheme can do its job.

6.5.3 More Measurements at 2 MHz and 500 kHz

In order to understand the cause(s) of the degradation when the clock rate increases, more measurements were done. First the DC values of the DAC elements are listed. Then the DAC output at O_1 (i.e. before the de-glitching filter) is studied.

The DAC steps (element DC values) were measured at 500 Hz and 2 MHz. The results are listed in Table 6.2. An interesting fact is that the DC values of the DAC elements "vary" with the clock frequency. The DC difference between the two clock

Table 6.2:The DAC under test matching data.

	500 kHz	2 MHz
sv1	0.0733 V	0.0722 V
sv2	0.0720 V	0.0706 V
sv3	0.0716 V	0.0703 V
sv4	0.0716 V	0.0703 V
sv5	0.0719 V	0.0705 V
sv6	0.0721 V	0.0709 V
sv7	0.0722 V	0.0710 V

Table 6.2:The DAC under test matching data.

	500 kHz	2 MHz
sv8	0.0724 V	0.0711 V
sv9	0.0724 V	0.0711 V
sv10	0.0720 V	0.0708 V
sv11	0.0717 V	0.0704 V
sv12	0.0716 V	0.0702 V
sv13	0.0717 V	0.0703 V
sv14	0.0718 V	0.0705 V
sv15	0.0719 V	0.0707 V
sv16	0.0723 V	0.0710
STD	0.6%	0.7%

frequencies is approximately 1.3 mV. This suggests that the DAC does not settle well when clocked at 2 MHz. The measurements below 1 MHz show that each element produces a constant step. Measurements at 3 MHz show that the DAC is largely unsettled, and mismatch-shaping does not work. A rough estimation is carried out as the following: The total load capacitor at O_1 is around 10 pF, and the output impedance of the class-AB output stage of the op amp is around 5 kΩ based on the original design. Neglecting the on resistance of a switch, the RC-time constant is about 50 ns. Therefore within 250 ns which is 1/2 of a clock cycle at 2 MHz, the DAC should be able to settle to 0.7% within the final value. For a step of 70 mV, 0.7% is approximately 0.5 mV, which is about the same order as that of the measured results.

Although the incomplete settling amount is bigger than the mismatch error at 2 MHz, as shown earlier in Figure 6.14, mismatch-shaping still works, which suggests

that the settling is largely linear. Measurements also show that superposition holds at 2 MHz.

Given the largest step input, the transient output of the DAC was measured at O_1 and plotted in Figure 6.15. There is no remarkable slewing on the curves. However, at the beginning of each transient, there is a nonlinear portion. To decide whether it is harmful, the input was scaled down by 1/2 (i.e. half the elements were disabled), and the output transient curves were scaled by 2 then over-printed on the corresponding plots. The observation can be made that superposition holds.

Notice that over-shoot exists in the step response. Remember as shown in Figure 6.7, there was no over-shoot when testing the step response of op amp using the unity-gain feedback configuration. The reason is probably because here at O_1, the capacitive load is heavier causing the second pole of the two stage op amp to move closer to the dominant pole so the phase margin and the damping ratio are decreased.

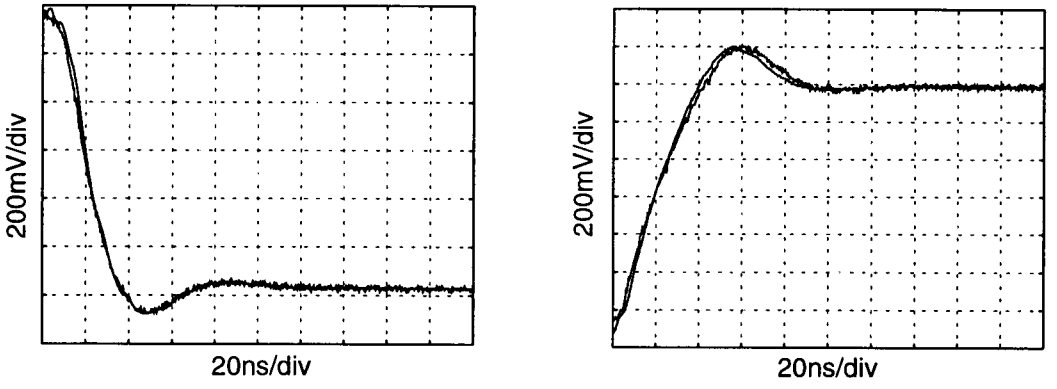


Figure 6.15:The transient output of the SC DAC at O_1 given step input.

6.5.4 The Use of the Deglitching filter

Aside from smoothing the output waveform, the major reason to use a lowpass de-glitching filter is to isolate the nonlinearity at O_1. Although as discussed above that superposition holds at O_1, mismatch-shaping does not gain too much at this node when nonlinear settling gets serious. The measured spectra with the 4th-order modulator and the 1st-order ESL are plotted in Figure 6.16. The experiment with the butterfly shuffler was

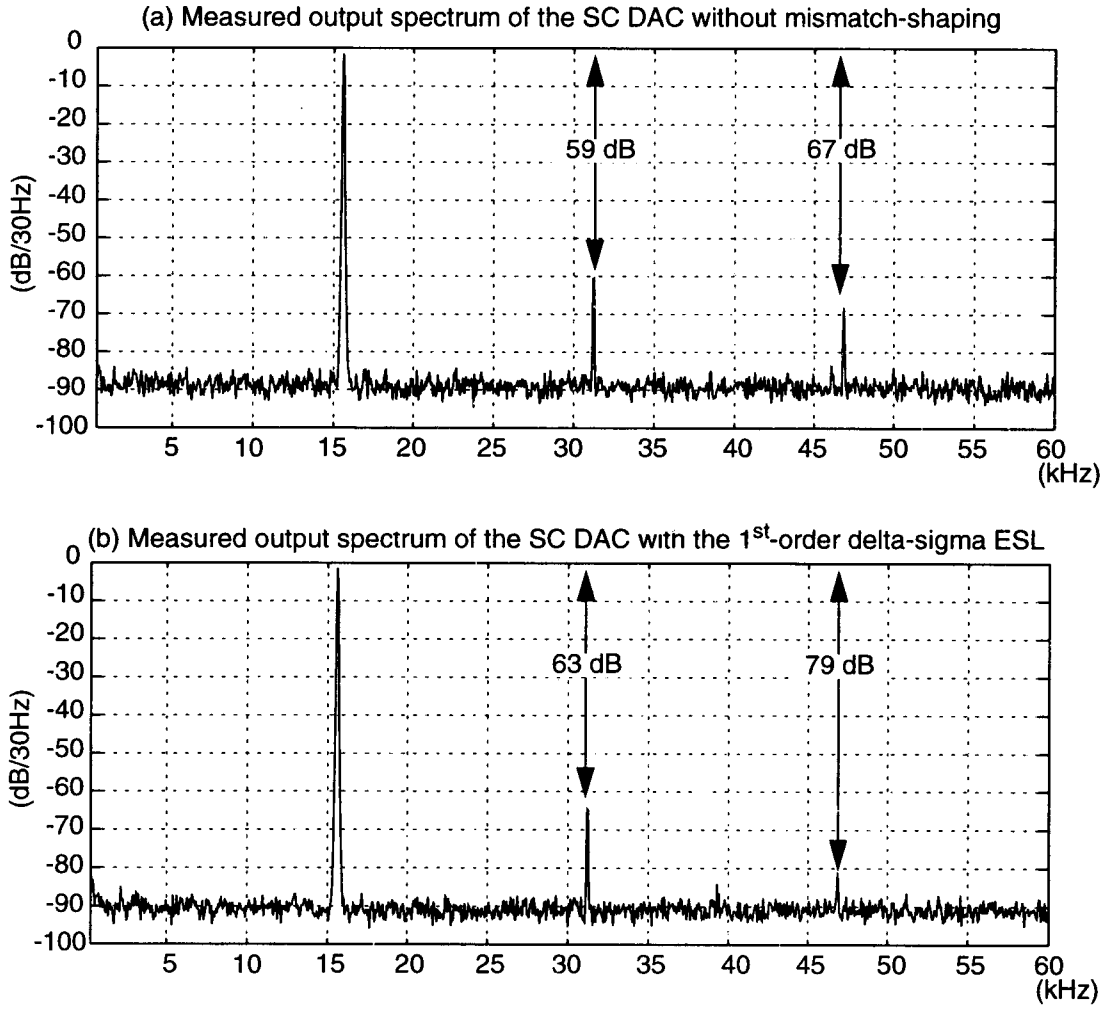


Figure 6.16:Measured spectra of the 4th-order delta-sigma SC DAC at O_1 with a sampling frequency of 2 MHz.

also done. Since the result is the same as that of the ESL, it is omitted here. As shown in the plots, the 2nd-order harmonic is decreased by only a few dB. This fact suggests that the 2nd-order nonlinearity in the transient response is about the same as that caused by element mismatch. As shown in Figure 6.14, this errors term is isolated by the de-glitching filter, and O_2 is much cleaner.

From the above discussion, we can predict that the transient step response at O_2 should have less harmonic content than O_1 does. The transient response at O_2 was measured and plotted in Figure 6.17. Notice that the settling is much slower and smoother than the plots in Figure 6.15, because of the lowpass filter effect of the de-glitching stage. The lowpass nature also tends to prevent slewing.

Since the errors caused by nonlinear settling at O_1 is blocked by the de-glitching filter and invisible at O_2, the nonlinearity of the de-glitching filter then becomes the limiting factor as clock goes high. The linearity of this stage can be improved a little by carefully adjusting the bias points. However, since it was not designed to be tuned

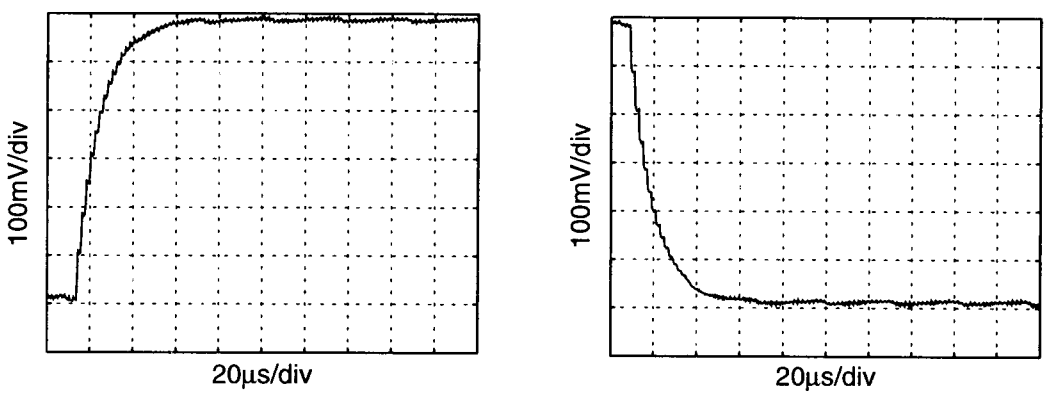


Figure 6.17:The transient output of the SC DAC at O_2 given a step input.

separately, only a few dBs can be gained before the tuning corrupts the previous stage. Careful design on this stage is crucial.

6.5.5 "Dynamic Mismatch"

As discussed in the previous sections, the nonlinear settling error at O_1 is big. According to measurements, this error term largely contributes to the 2nd-order harmonic and can be comparable to the mismatch-introduced 2nd-order harmonic. Yet an interesting fact was observed that this "dynamic" error can also be suppressed by mismatch-shaping. As shown in Figure 6.18, without mismatch-shaping, the 2nd- and 3rd-order harmonics are both 63 dB lower than the signal at O_1. While as shown in Figure 6.8, at O_2, these numbers become 73 dB and 69 dB, respectively. The difference is due to the nonlinear portion in the settling at O_1. Since the distortion caused by static mismatch error is down at the -70 dB level as being observed at O_2, the distortion at O_1 is dominated by some dynamic error such as nonlinear settling. However, when mismatch-shaping is turned on, the 2nd-order harmonic is reduced to -78 dB and the 3rd-order harmonic disappears at O_1. Therefore, this dynamic error should be associated with element mismatch, or, "element-dependent".

To locate the error source, the main DAC circuit was re-studied. A positive input branch and a negative input branch are re-drawn in Figure 6.19. A "dynamic" mismatch exists between the positive branch and the negative branch. This is because the reference nodes of the two branches are reversed, therefore, the operations of the two branches do not match. As shown in Figure 6.19, when an element in the negative branch is chosen to be turned on (i.e. to make a positive step) in ϕ_2, it has already been "pre-discharged" in the previous ϕ_1 (please note that the "inverting" structure is used so that discharging an input capacitor makes the output decrease by one step) and no switching is needed, while

the positive branch has to switch to GND in phase (ϕ_2) to discharge in order to make a positive step. This difference will cause transient mismatch when the circuit is settling, and it can not be seen at O_2 as long as the O_1 settles at the end of ϕ_2. Therefore, this is a "dynamic" mismatch.

To wipe out this error term, unipolar structures may be used. However, doing so requires the virtual ground (AGND) to be tuned away from the mid-point between the

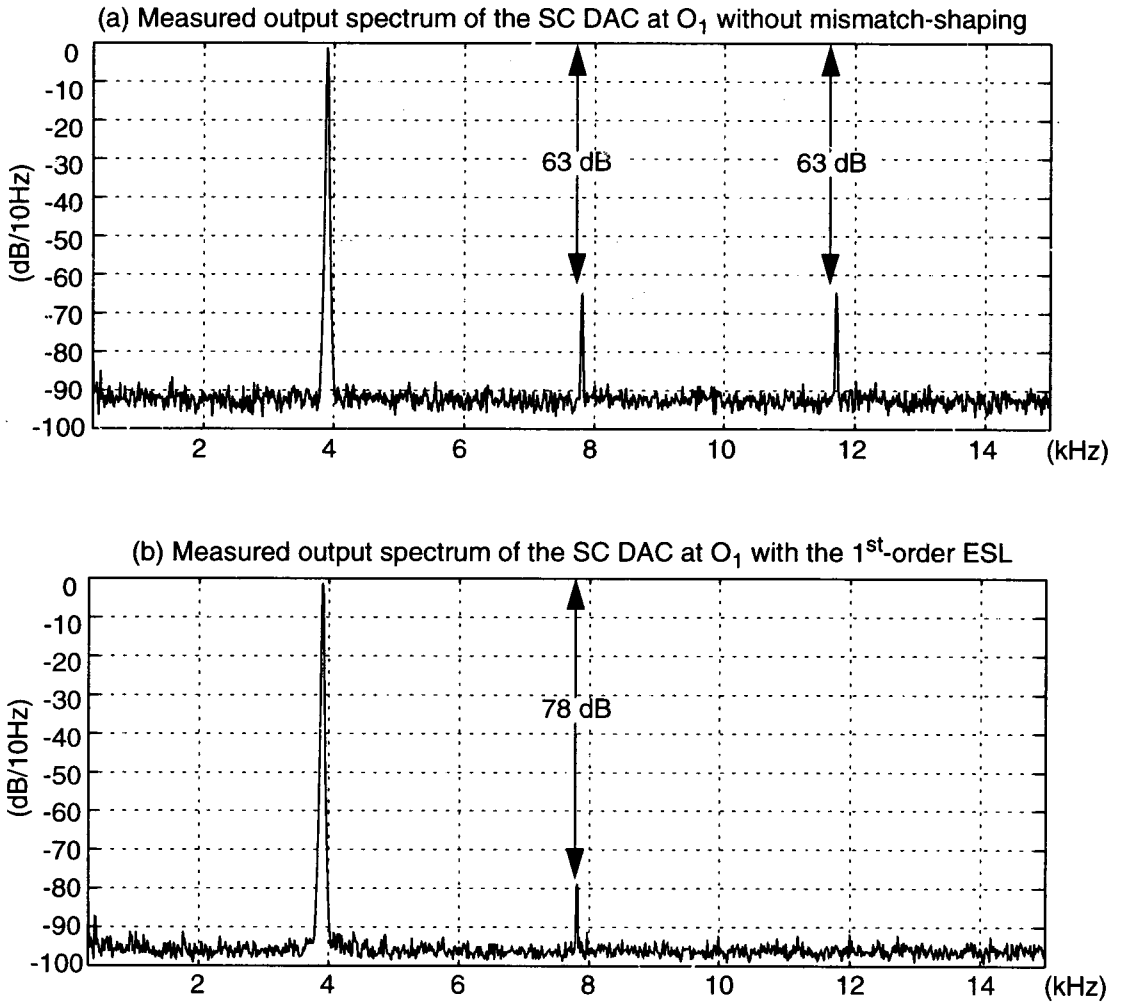


Figure 6.18:Measured spectra of the 4th-order delta-sigma 16-element DAC at O_1 with a sampling frequency of 500 kHz.

supply voltages if the DAC output swing needs to be maximized. In return, a "shifted" virtual ground may pose stricter requirements on the op amp common-mode input range.

To verify that there is a difference between the settling characteristics of positvie and negative branches, an 8-element DAC was organized using only negative branches. There should not have been any bias to choose the group polarity. The only reason was that one of the elements on the positive side seemed to deviate from the mean by too much. Then the 8-element DAC was studied at O_1 with the 1st-order ESL turning on and off. The measured spectra are shown in Figure 6.20. The 3rd-order harmonic decreases to -83 dB with mismatch-shaping, because it is dominated by element static mismatch as already indicated by the results shown in Figure 6.18. The 2nd-order distortion remains the same when the ESL turns on and off, while with both positive and negative branches, it can make more than 10 dB difference (as illustrated in Figure 6.18). Therefore, it was the dynamic mismatch between the positive and the negative branches that dominated the show.

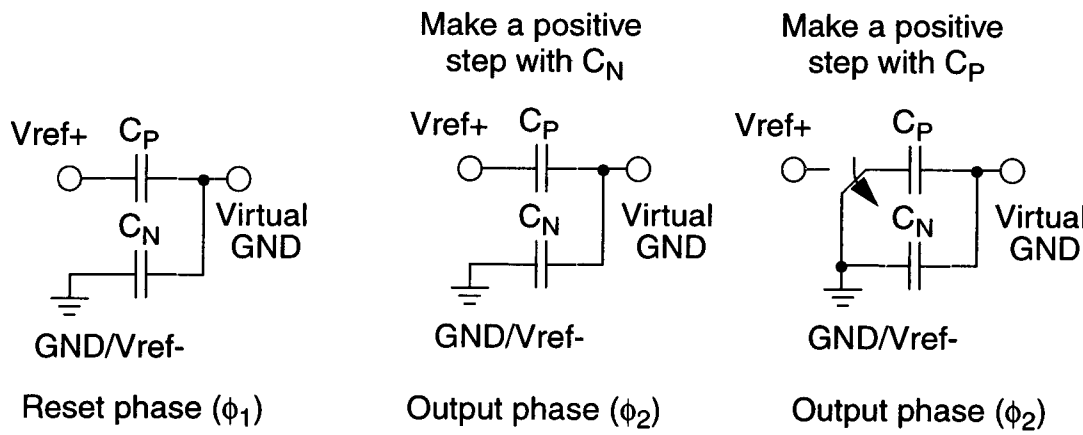


Figure 6.19:The dynamic mismatch between a positive branch and a negative branch.

One more interesting point about this dynamic mismatch is that it is not an error term or a bad thing, because when combined with mismatch-shaping, the performance is better than no dynamic mismatch. To show this, the 8-element DAC was re-routed so that there were 4 elements from the positive side and 4 from the negative side. The results are plotted in Figure 6.21. Without mismatch-shaping, the 2nd-order harmonic is 3 dB worse than the "8-negative-element" DAC. But when mismatch is on, it decreases by 9 dB, even better than no dynamic mismatch.

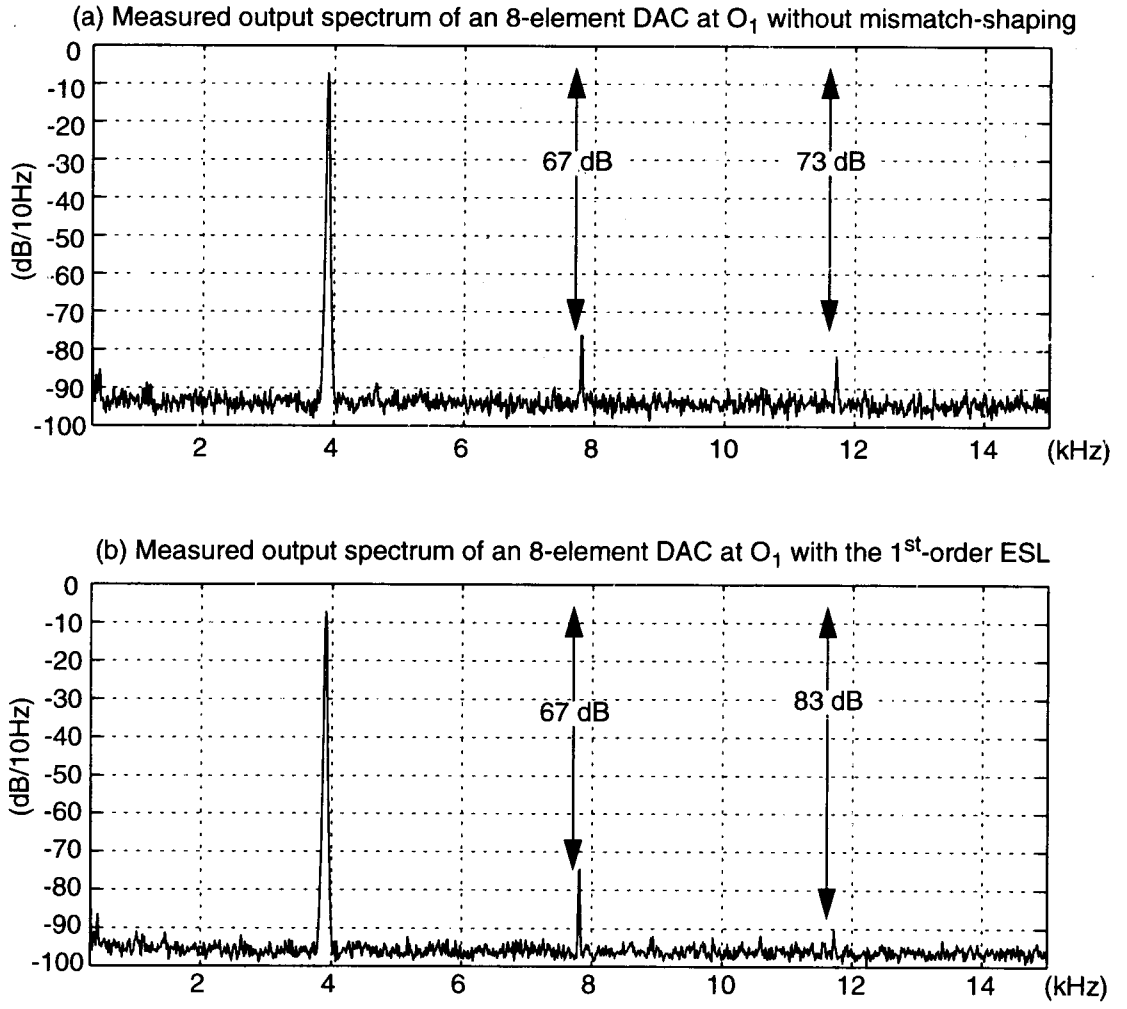


Figure 6.20:Measured spectra of the 4th-order delta-sigma 8-negative-element DAC at O_1 with a sampling frequency of 500 kHz.

6.5.6 Other Issues

The circuit noise turns out to be another source of trouble. The mismatch-shaping schemes are only able to improve the SNRs by a few dBs before other noise sources take over the show. One large noise source seems to be the clock. Originally, the clock signal driving the SC DAC was taken from the Xilinx FPGA; this resulted in increased noise. Later when the DAC picked up the clock from the crystal on its own board, the noise

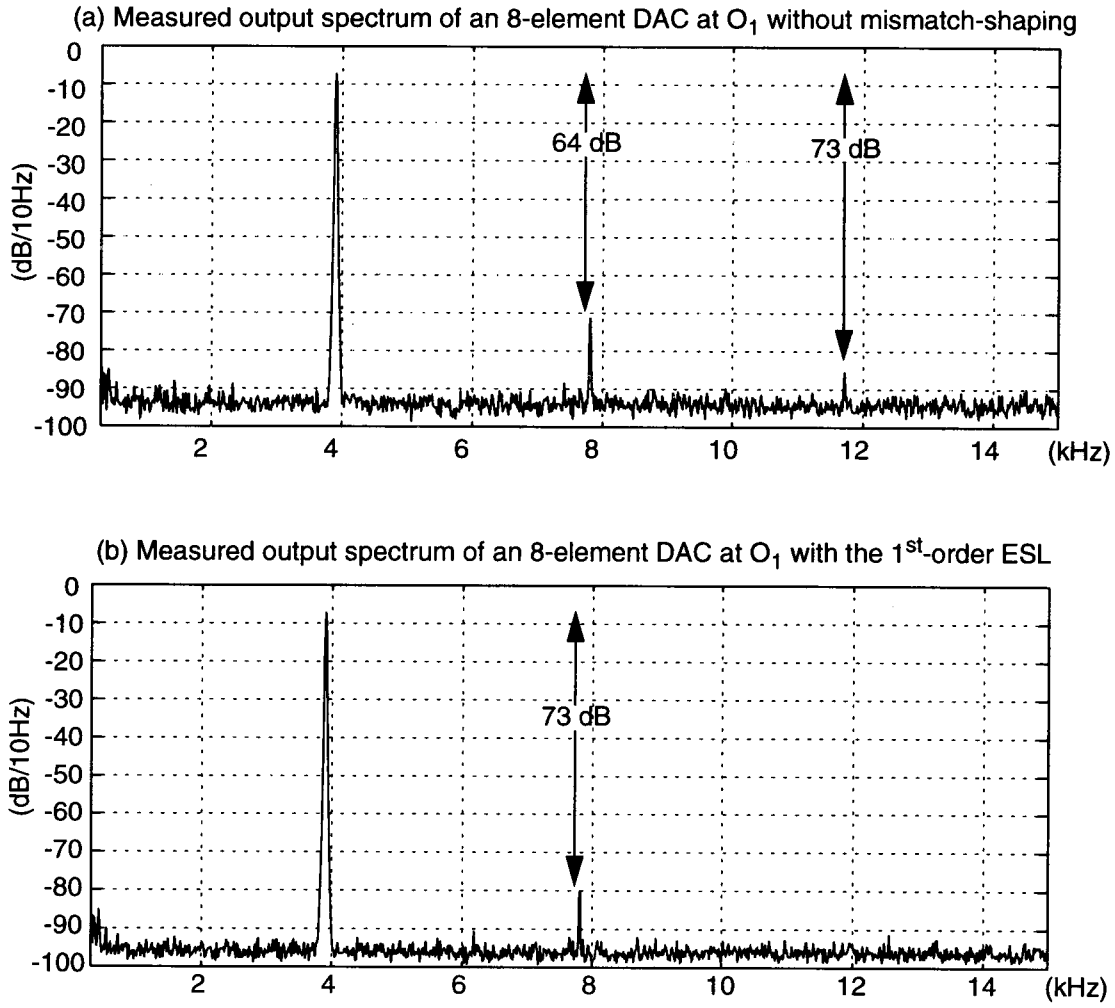


Figure 6.21:Measured spectra of the 4th-order delta-sigma 8-element DAC at O_1 with a sampling frequency of 500 kHz.

decreased by almost 10 dB. Interference from the clock still could be the dominant noise source after the mismatch noise is taken care of. Other noise sources such as 1/f noise can also contribute especially at low frequencies.

It was once worried that the signal might interfere with the reference. That was the reason the *sv* scrambler/descrambler was designed. Fortunately, this error term did not show up. It was observed that turning on and off the scrambler makes no difference.

6.6 Conclusions

The prototype SC DAC IC designed in Chapter 5 was fabricated and tested with a 4^{th}-order modulator and various mismatch-shaping schemes. The 1^{st}-order butterfly shuffler is found to be the most effective given the DAC under test. The 2^{nd}-order delta-sigma ESL is the most powerful in suppressing the mismatch errors at low frequencies but it is hardware intensive and slow.

Chapter 7. Conclusions and Future Work

7.1 Summary

In Chapter 2, the background knowledge for delta-sigma modulation and mismatch-shaping was introduced. A few existing mismatch-shaping schemes were reviewed.

Chapter 3 took a closer look at some dynamic mismatch-shaping architectures. Comparisons between the delta-sigma ESL and the tree-shuffler were made using simulations and analytical techniques. A design of the vector quantizer and a revised delta-sigma ESL were proposed. A generalized butterfly shuffler architecture capable of high-order and bandpass mismatch-shaping was presented.

In Chapter 4, a multi-bit delta-sigma DAC was evaluated and designed at the system level given a CMOS switched-current 2-segmented DAC core built by Analog Devices, Inc. Since the mismatch error in the segmented DAC is dominated by the MSB cells, applying mismatch-shaping only to the MSB cells suppresses the bulk of the mismatch error with a low hardware cost.

In Chapter 5, a prototype 16-element SC mismatch-shaping DAC was designed. Three lowpass $\Delta\Sigma$ modulators (two 4^{th}-order, one 8^{th}-order) were designed as examples to drive the mismatch-shaping DAC. An 8^{th}-order bandpass modulator was derived from one of the 4^{th}-order lowpass prototypes to demonstrate bandpass mismatch-shaping. The SC DAC core was fabricated using the Orbit 1.2 μm double-poly double-metal CMOS process.

The 16-element SC DAC was tested and the results were listed and discussed in Chapter 6. Three mismatch-shaping schemes: 1^{st}-order butterfly shuffler, 1^{st}-order delta-sigma ESL, and 2^{nd}-order delta-sigma ESL were designed and implemented along with one of the 4^{th}-order modulators designed in Chapter 5 using Xilinx FPGAs. By applying mismatch-shaping on the SC DAC, spurious-free dynamic ranges as large as 90 dB were achieved.

By accomplishing the work described above, the thesis made contributions to the design and demonstration of high-order mismatch-shaping, the design of high-order multi-bit delta-sigma modulators, and the generalization of element mismatch-shaping.

7.2 Future Work

A fully integrated system combining the modulator, the mismatch-shaping logic and the analog DAC core could be an interesting and educational project. Such a high resolution DAC would have numerous applications in many fields such as audio, multi-media and communication. Also, a multi-bit delta-sigma ADC employing mismatch-shaping is a challenging and exciting possibility. There are undoubtedly numerous design issues worthy of study in such a high-performance ADC.

Further theoretical work includes deepening the understanding of mismatch-shaping, extending the technique to non-unit-element DACs or developing more effective mismatch-shaping structures.

Bibliography

[1] J. C. Candy and G. C. Temes, "Oversampling methods for A/D and D/A conversion," in *Oversampling Delta-Sigma Converters*, J. C. Candy and G. C. Temes, Eds. New York: IEEE Press, 1991.

[2] *Delta-Sigma Data Converters: Theory, Design, and Simulation*, S. R. Norsworthy, R. Schreier and G. C. Temes, Eds. New York: IEEE Press, 1996.

[3] R. L. Geiger, P. E. Allen and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, pp. 642-647, McGraw-Hill Publishing Company, 1990.

[4] R. J. Van de Plassche and D. Goedhart, "A monolithic 14-bit D/A converter," *IEEE Journal of Solid-State Circuits*, vol. 14, no. 3, pp. 552-556, June 1979.

[5] L. R. Carley and J. Kenney, "A 16-bit 4'th order noise-shaping D/A converter," Proceedings of the 1988 *IEEE Custom Integrated Circuits Conference*, pp. 21.7.1-21.7.4, Rochester NY, May 1988.

[6] L. R. Carley, "A noise-shaping coder topology for 15+ bit converters," *IEEE Journal of Solid-State Circuits*, vol. SC-24, pp. 267-273, April 1989.

[7] B. H. Leung and S. Sutarja, "Multi-bit Σ–Δ A/D converter incorporating a novel class of dynamic element matching," *IEEE Transactions on Circuits and Systems - II*, vol. 39, pp. 35-51, Jan. 1992.

[8] F. Chen and B. H. Leung, "A high resolution multibit sigma-delta modulator with individual level averaging," *IEEE Journal of Solid-State Circuits*, vol. SC-30, no. 4, pp. 453-460, April 1995.

[9] T. Cataltepe, G. C. Temes, and L. E. Larson, "Digitally corrected multi-bit Σ–Δ data converters," *Proceedings of the 1989 IEEE International Symposium on Circuits and Systems*, pp. 647-650, May 1989.

[10] M. J. Story, "Digital to analogue converter adapted to select input sources based on a preselected algorithm once per cycle of a sampling signal," U.S. patent number 5138317, Aug. 11 1992 (filed Feb. 10 1989).

[11] W. Redman-White and D. J. L. Bourner, "Improved dynamic linearity in multi-level $\Sigma\Delta$ converters by spectral dispersion of D/A distortion products," *IEE Conference Publication European Conference on Circuit Theory and Design*, pp. 205-208, Sept. 5-8 1989.

[12] H. S. Jackson, "Circuit and method for cancelling nonlinearity error associated with component value mismatches in a data converter," U.S. patent number 5221926, June 22 1993 (filed July 1 1992).

[13] R. T. Baird and T. S. Fiez, "Improved ΔΣ DAC linearity using data weighted averaging," *Proceedings of the 1995 IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 13-16, May 1995.

[14] R. T. Baird and T. S. Fiez, "Linearity enhancement of multibit DS A/D and D/A converters using data weighted averaging," *IEEE Transactions on Circuits and Systems II*, vol. 42, no. 12, pp. 753-762, December 1995.

[15] R. W. Adams and T. W. Kwan, "Data-directed scrambler for multi-bit noise-shaping D/A converters," U.S. patent number 5404142, April 4 1995 (filed Aug. 1993).

[16] R. Schreier and B. Zhang, "Noise-shaped multibit D/A convertor employing unit elements," *Electronics Letters*, vol. 31, no. 20, pp. 1712-1713, Sept. 28 1995.

[17] R. M. Gray, "Oversampled sigma-delta modulation," *IEEE Transactions on Communications*, vol. 35, no. 5, pp. 481-489, May 1987.

[18] R. Schreier, "Noise-shaped coding," *Ph.D. thesis, University of Toronto*, 1991.

[19] H. Wang, "A geometric view of SD modulations," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 39, no. 2, pp. 402-405, June 1992.

[20] H. Wang, "On the stability of third-order sigma-delta modulation," *Proceedings of the 1993 IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 1377-1380, May 3-6 1993.

[21] R. Schreier and M. Snelgrove, "Stability in a general SD modulator," *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1769-1772, May 1991.

[22] R. Schreier and Y. Yang, "Stability tests for single-bit sigma-delta modulators with second-order FIR noise transfer functions," *IEEE Proc. ISCAS'92*, vol. 3, pp. 1316-1319, May 1992.

[23] S. C. Pinault and P. V. Lopresti, "On the behavior of the double-loop sigma-delta modulator," *IEEE Transactions on Circuits and Systems II*, vol. 40, no. 8, pp. 467-479, Aug. 1993.

[24] S. Hein and A. Zakhor, "On the stability of sigma delta modulators," *IEEE Transactions on Signal Processing,* vol. 41, no. 7, pp. 2322-2348, July 1993.

[25] M. Goodson, B. Zhang and R. Schreier, "Proving stability of delta-sigma modulators using invariant sets," P*roceedings, IEEE International Symposium on Circuits and Systems,* May 1995, vol. 1, pp. 633-636.

[26] W. L. Lee, "A novel higher order interpolative modulator topology for high resolution oversampling A/D converters," Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1987.

[27] K. C. H. Chao, S. Nadeem, W. L. Lee, and C. G. Sodini, "A higher order topology for interpolative modulators for oversampling A/D conversion," *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 309-318, March 1990.

[28] P. H. Gailus, W. J. Turney and F. R. Yester, Jr., "Method and Arrangement for a sigma delta converter for bandpass signals," US Patent number 4,857,928, filed January 28, 1988, awarded August 15, 1989, Motorola, Inc., Schaumburg, Ill.

[29] R. Schreier and W. M. Snelgrove, "Bandpass sigma-delta modulation," *Electronics Letters*, vol. 25, no. 23, pp. 1560-1561, November 9 1989.

[30] B. Zhang, "Delta-sigma modulators employing continuous-time circuits and mismatch-shaped DACs," *Ph.D. thesis, Oregon State University*, 1996.

[31] S. A. Jantzi, R. Schreier and M. Snelgrove, "A bandpass $\Sigma\Delta$ A/D convertor for a digital AM receiver," *Proceedings of the IEE International Conference on Analogue-to-Digital and Digital-to-Analogue Conversion*, Swansea, UK, pp. 75-80, September 1991.

[32] S. A. Jantzi, W. M. Snelgrove and P. F. Ferguson, Jr., "A fourth-order bandpass sigma-delta modulator," *IEEE Journal of Solid-State Circuits,* vol. SC-28, no. 3, pp. 282-291, March 1993.

[33] H. Lin, J. da Silva, B. Zhang and R. Schreier, "Multi-bit DAC with noise-shaped element mismatch," *Proceedings of the 1996 IEEE International Symposium on Circuits and Systems,* vol. 1, pp. 235-238, May 1996.

[34] I. Galton, "Noise-shaping D/A converters for $\Delta\Sigma$ modulation," *Proceedings of the 1996 IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 441-444, May 1996.

[35] I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," to appear in *IEEE Transactions on Circuits and Systems II*, 1997.

[36] O. J. A. P. Nys and R. K. Henderson, "An analysis of dynamic element matching techniques in sigma-delta modulation," *Proceedings of the 1996 IEEE International Symposium on Circuits and Systems,* vol. 1, pp. 231-234, May 1996.

[37] A. Yasuda and H. Tanimoto, "Noise shaping dynamic element matching method using tree structure," *Electronics Letters*, 16th, vol. 33, no. 2, pp. 130-131, Jan. 1997.

[38] A. Yasuda, H. Tanimoto and T. Lida, "A 100Hz 9.6mW multi-bit $\Delta\Sigma$ DAC and ADC using noise shaping dynamic element matching with tree structure," *1998 IEEE International Solid-State Circuits Conference*, pp. 64-65, Feb. 1998.

[39] D. Robertson, et al., "Thermometer-coded data-directed scrambler for multi-bit noise-shaping D/A converters," filed for U.S. patent, 1996.

[40] D. Mercer and L. Singer, "A 12-bit 125MSPS CMOS D/A designed for special performance," *1996 International Symposium on Low Power Electronics and Design*, pp. 243, 1996.

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

[42] Y. Matsuya, K. Uchimura, A. Iwata and T. Kaneko, "A 17-bit oversampling D-to-A conversion technology using multistage noise shaping," *IEEE Journal of Solid-State Circuits,* vol. 24, no. 4, pp. 969-975, August 1989.

[43] L. A. Williams, III and B. A. Wooley, "A third-order sigma-delta modulator with extended dynamic range," *IEEE Journal of Solid-State Circuits*, vol. SC-29, pp. 193-202, March 1994.

[44] Analog Devices, Inc., *ADICE5 User's Manual.*

[45] Orbit Semiconductor, Inc. web-page, "http://www.orbitsemi.com/".

[46] Jen-I Pi, "MOSIS scalable CMOS design rules (revision 7)," *The MOSIS Service*, Information Sciences Institute, University of Southern California, April, 1995.

[47] *XACT Libraries Guide*, April, 1994.

[48] R. Schreier, "Mismatch-shaping digital-to-analog conversion," *The 103rd Convention of Audio Engineering Society*, 1997.

[49] W. F. Hudson, "Experimental verification of a mismatch-shaping DAC," *Master's thesis, Oregon State University*, 1997.

[50] K. Haug, G. C. Temes and L. Martin, "Improved offset-compensation schemes for SC circuits," *Proceedings of the 1984 IEEE International Symposium on Circuits and Systems*, pp. 1054-1057, 1984.

[51] J. A. C. Bingham, "Applications of direct-transfer SC integrator," *IEEE Transactions on Circuits and Systems*, vol. 31, pp. 419-420, April 1984.

[52] R. Gregorian and G. C. Temes, *Analog MOS ICs for Signal Processing*, Wiley, 1986.

[53] C. Y. Chen, "Design of a postfilter for a delta-sigma digital-to-analog converter," *Master's thesis, Oregon State University*, 1994.

[54] Y. Huang, "Design techniques of high-performance switched-capacitor circuits in the presence of component imperfections," *Ph.D. thesis, Oregon State University*, 1997.

[55] R. Hogervost, J. P. Tero, R. G. H. Eschauzier and J. H. Huijsing, "A compact power-efficient 3v CMOS rail-to-rail input/output operational amplifier for VLSI cell libraries," *IEEE Journal of Solid-State Circuits,* vol. 29, no. 12, Dec. 1994.

[56] J. Fisher and R. Koch, "A highly linear CMOS buffer amplifier," *IEEE Journal of Solid-State Circuits,* vol. sc-22, no. 3, June 1987.

[57] J. F. Duque-Carrillo, J. M. Valverde and R. P. Perez-Aloe, "Constant-g_m rail-to-rail common-mode range input stage with minimum CMRR degradation," *IEEE Journal of Solid-State Circuits,* vol. 28, no. 6, June 1993.

[58] W. S. Wu, W. J. Helms, J. A. Kuhn and B. E. Byrkett, "Digital-compatible high performance operational amplifier with rail-to-rail input and output ranges," *IEEE Journal of Solid-State Circuits,* vol. 29, no. 1, Jan. 1994.

[59] K. Nagaraj, "Constant transconductance CMOS amplifier input stage with rail-to-rail input common mode voltage range," *IEEE Transactions on Circuits and Systems*, vol. 42, no. 5, May 1995.

[60] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, second edition, Wiley, 1984.

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

[62] R. Schreier, "Design tools for delta-sigma converters," *Center for the Design of Analog-Digital Integrated Circuits*, July 1995.

[63] P. Horowitz and W. Hill, *The Art of Electronics*, 2nd edition, Cambridge University Press, New York, 1989.

[64] "*How to test basic operational amplifier parameters*," Application Notes, vol. I, Analog Devices, Inc.

[65] J. LaMay and H. Bogard, "How to obtain maximum practical performance from state-of-the-art delta-sigma analog-to-digital converters," *IEEE Transactions on Instrumentation and Measurement*, vol. 41, no. 6, Dec. 1992.

APPENDICES

Appendix I. Design Details of the Modulators in Chapter 4

The NTF pole-zero plots, the SNR curves and the bit-true system diagrams of the lowpass modulators designed in Chapter 4 are appended. The details of the bandpass modulators can be derived directly from their lowpass prototypes, and hence are omitted here.

App.1.1 4^{th}-Order Lowpass Modulator optimized with OSR = 8

Figure A.1.1 illustrates the original (before coefficient quantization) pole-zero placement in the Z-domain and the magnitude response of the NTF. The bit-true schematic of the modulator after coefficient scaling and quantization is shown in Figure A.1.2.The modulator was simulated in the time domain. The simulation results are summarized on the SNR VS. input amplitude curves in Figure A.1.3. The dashed line and the solid line correspond to the modulators before and after the coefficient quantization, respectively.

Simulations show that the peak SNR is around 79 dB. Figure A.1.4 shows an example of the modulator output waveform and the spectrum given a single-tone input located at the center of the band-of-interest with an amplitude of 0.72 of the full scale. The spectrum is obtained by performing an 8192 Hann-windowed FFT on the data from the time-domain simulation. The two notches in the spectrum correspond to the two in-band zeros of the NTF.

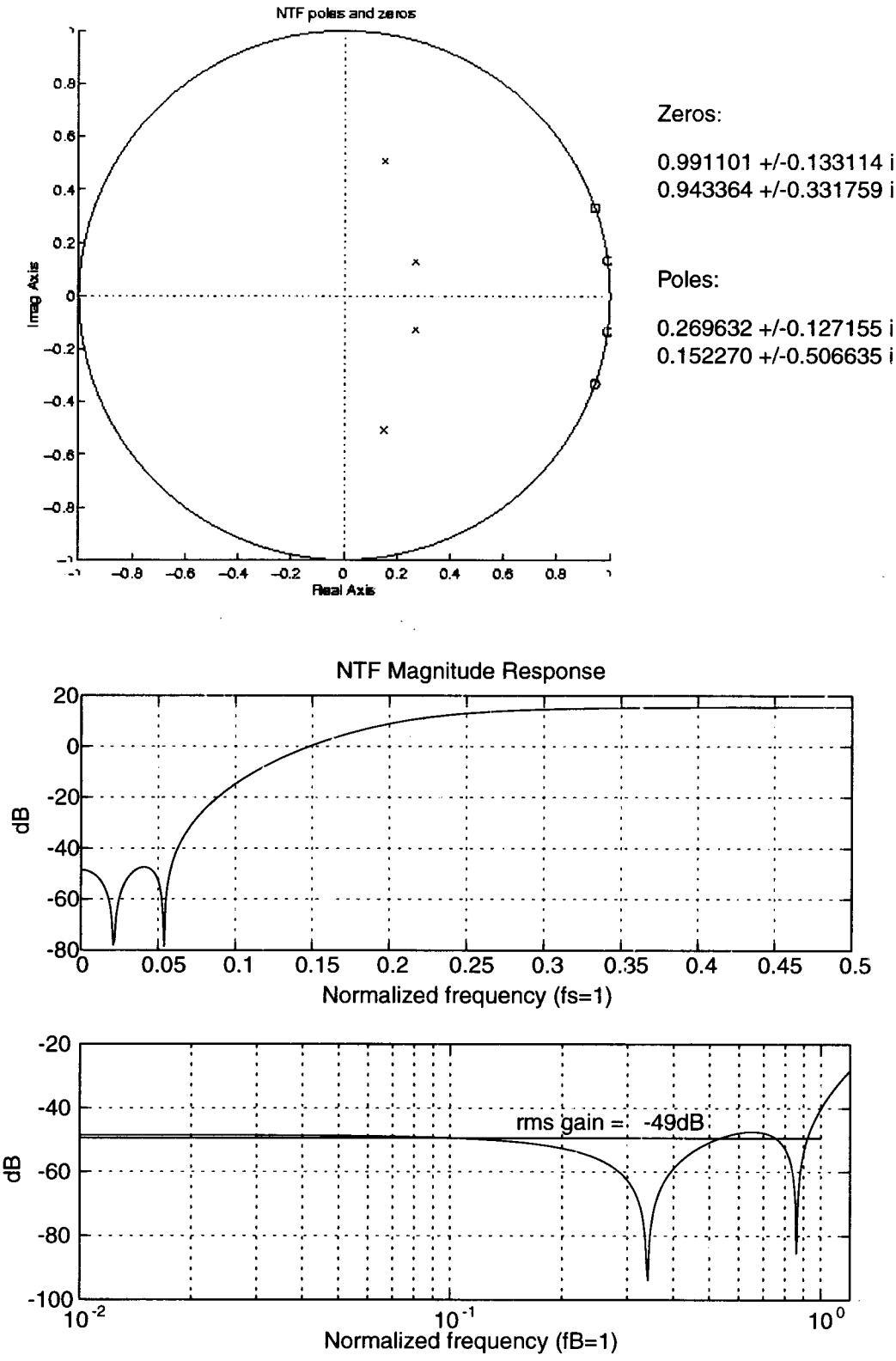


Figure A.1.1: Original 4th-order NTF optimized with *OSR* = 8.

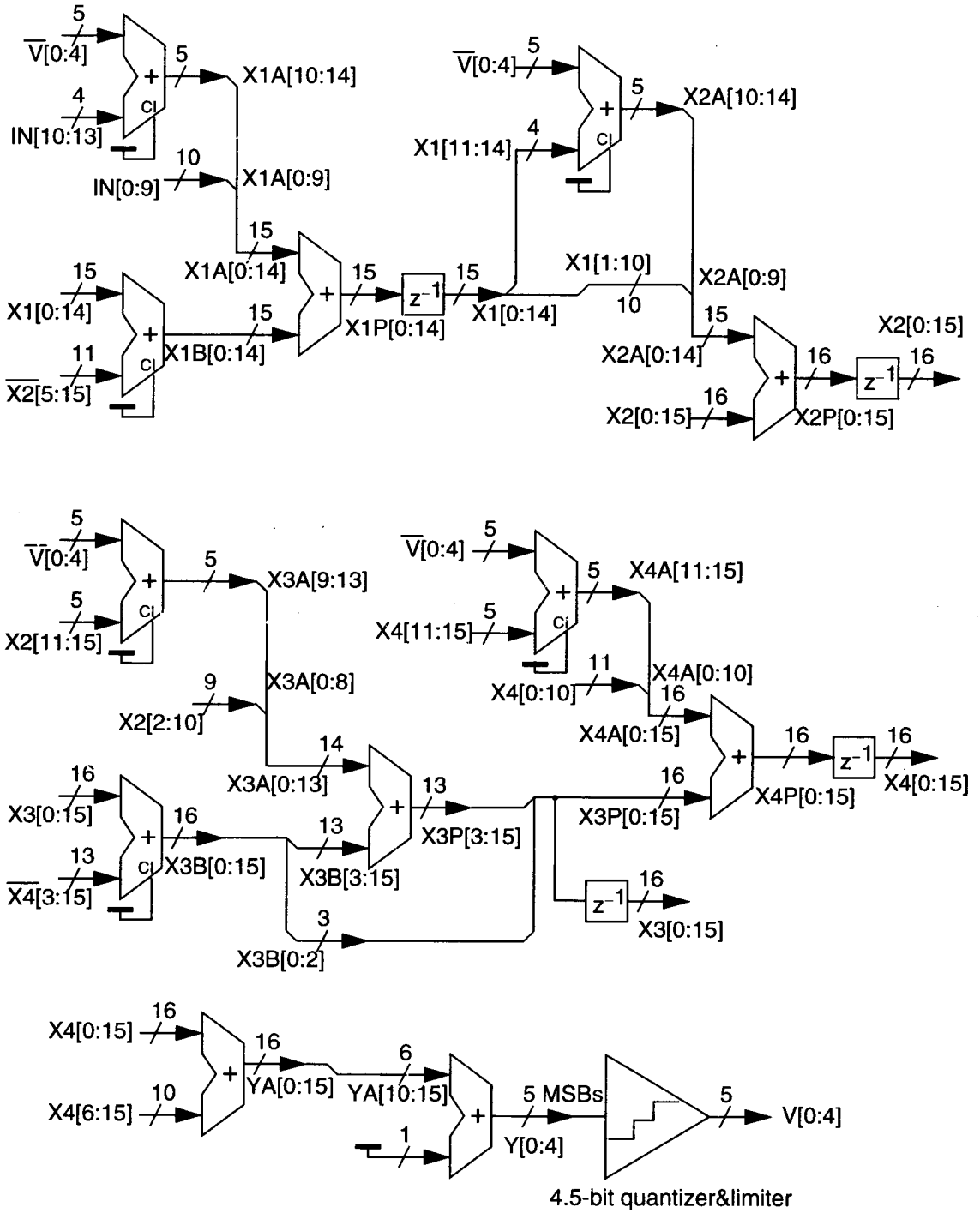


Figure A.1.2: Schematic of a 4^{th}-order 4.5-bit $\Delta\Sigma$ modulator optimized with $OSR = 8$.

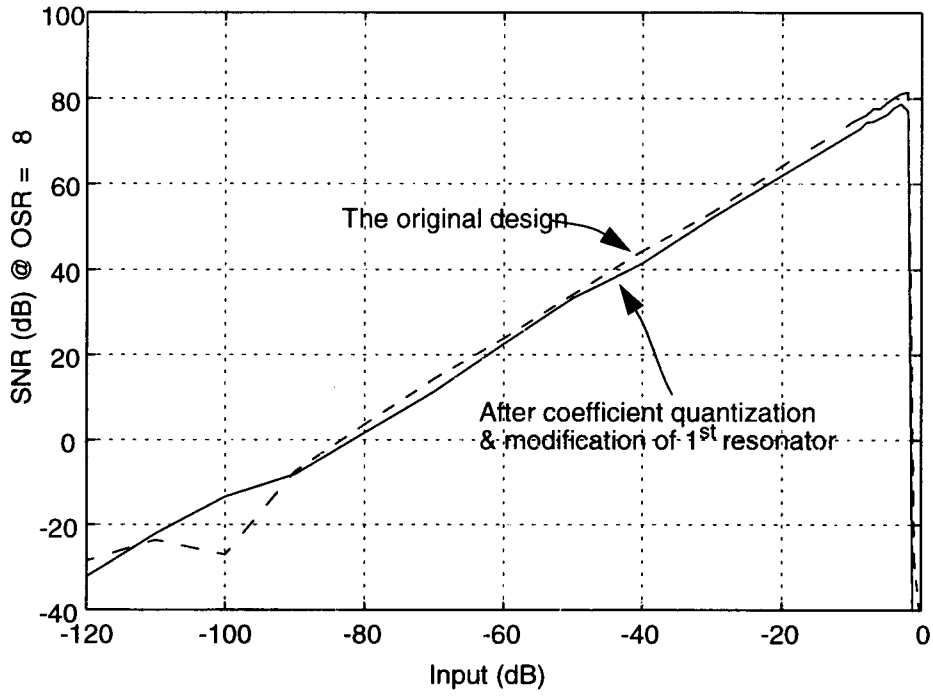


Figure A.1.3: SNR VS. input tone amplitude. (4^{th}-order, $OSR = 8$)

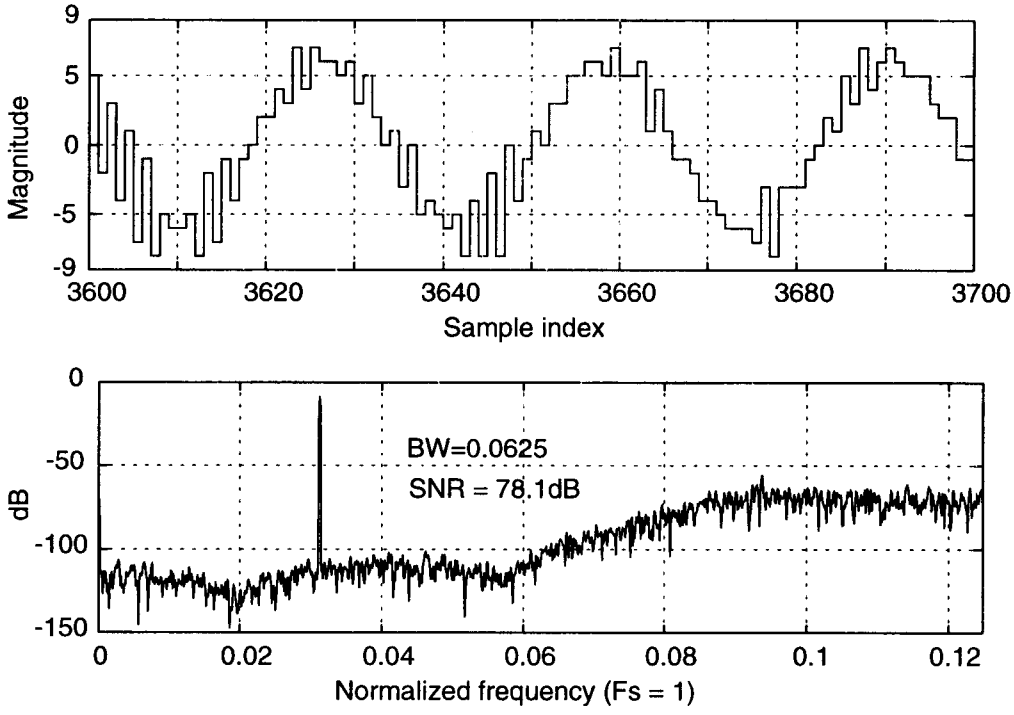


Figure A.1.4: Output waveform and spectrum of the 4^{th}-order modulator ($OSR = 8$).

App.1.2 4th-Order Lowpass Modulator optimized with OSR = 16

The pole-zero plot and the frequency response of the NTF are illustrated in Figure A.1.5. The bit-true schematic of the modulator is drawn in Figure A.1.6. Figure A.1.7 shows the SNR v.s. input magnitude curves obtained from simulations. The peak SNR is around 107 dB.

App.1.3 8th-Order Lowpass Modulator optimized with OSR = 8

The pole-zero plot and the frequency response of the NTF is illustrated in Figure A.1.8. Figure A.1.9 illustrates the bit-true schematic of the modulator. The modulator was simulated in time-domain before and after the coefficient quantization. The SNR curves are shown in Figure A.1.10. Figure A.1.11 shows an example of the modulator output waveform and the spectrum given a single-tone input located at the center of the band-of-interest with an amplitude of 0.58 of full scale.

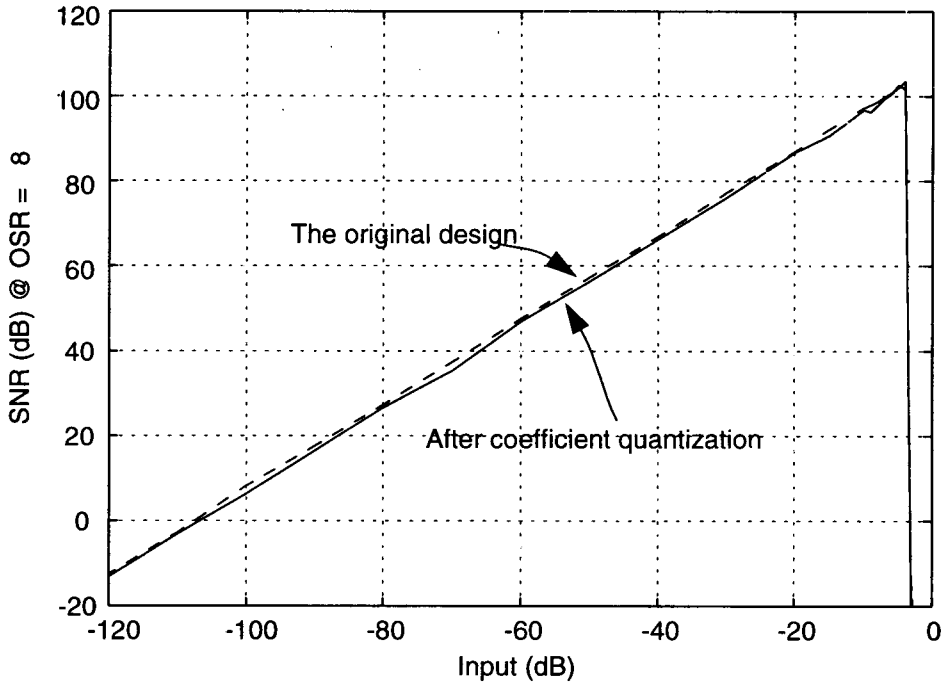


Figure A.1.10: SNR v.s. input tone amplitude. (8th-order, *OSR* = 8)

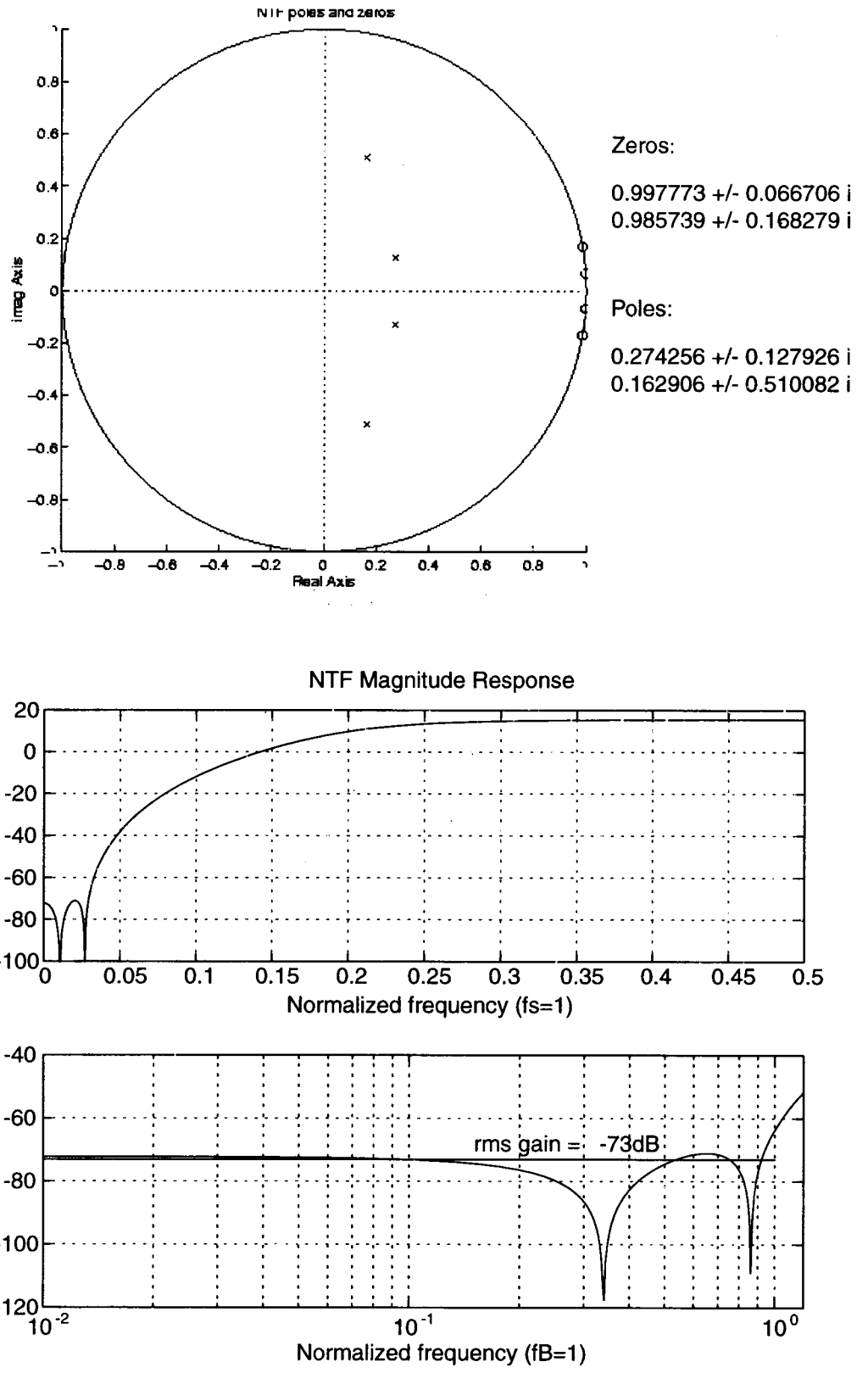


Figure A.1.5: Original 4[th]-order NTF optimized with *OSR* = 16.

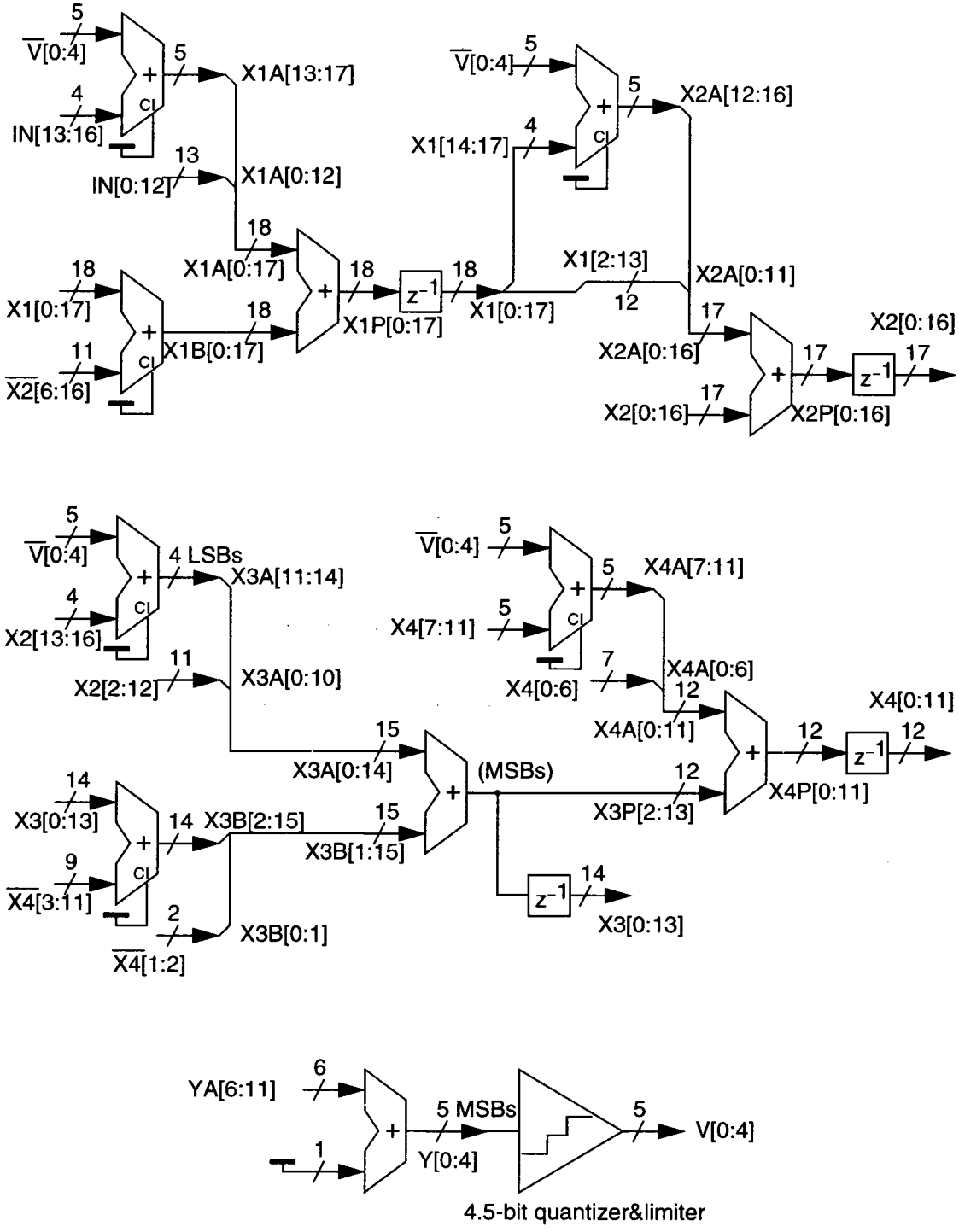


Figure A.1.6: Schematic of a 4^{th}-order 4.5-bit $\Delta\Sigma$ modulator optimized with $OSR = 16$.

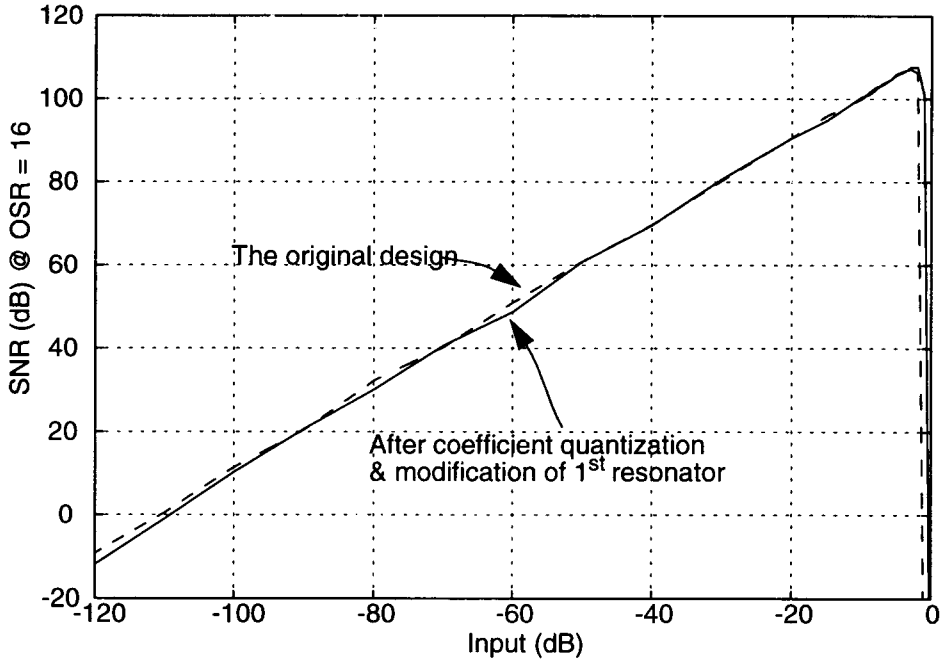


Figure A.1.7: SNR v.s. input tone amplitude. (4^{th}-order, $OSR = 16$).

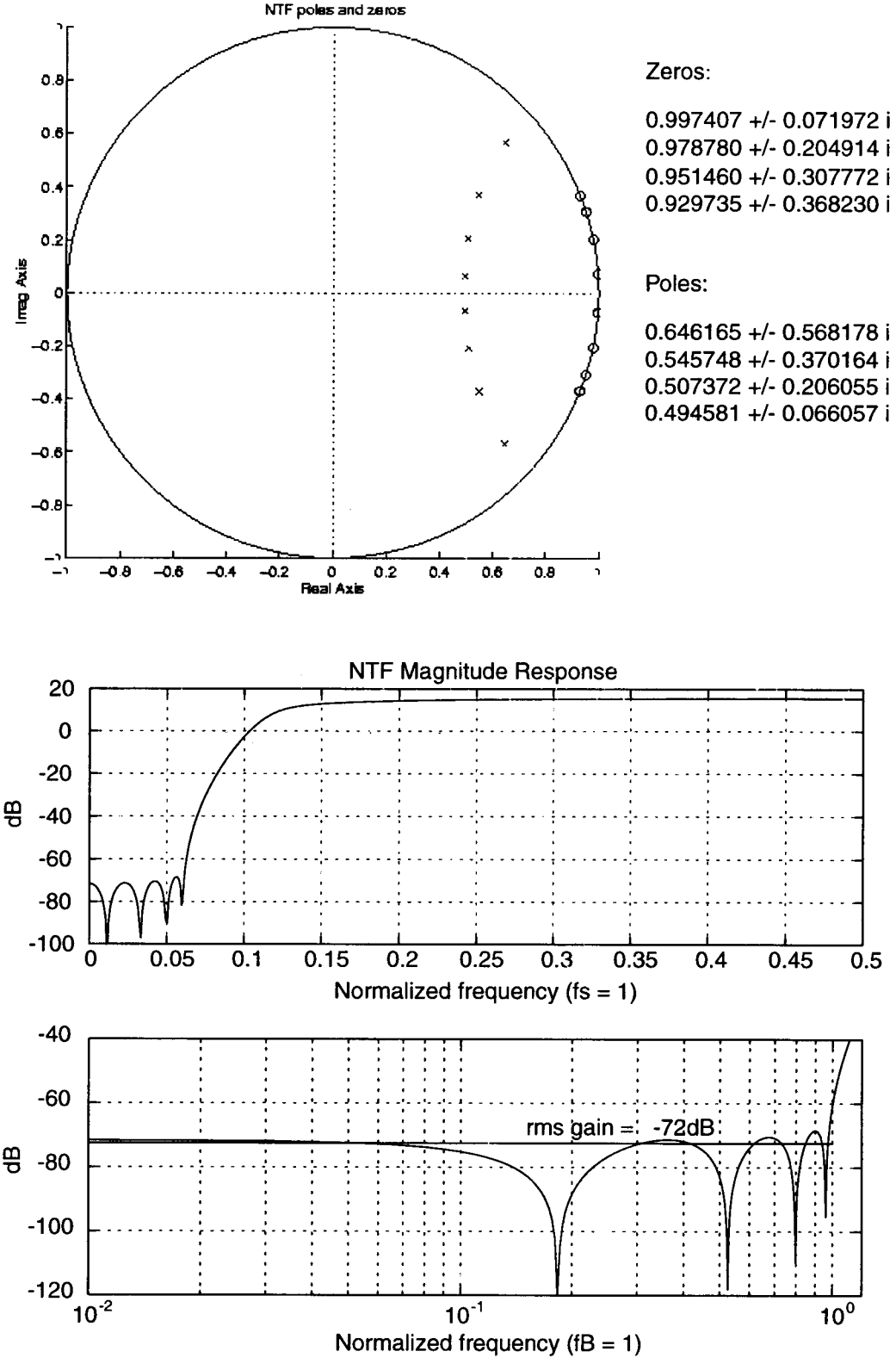
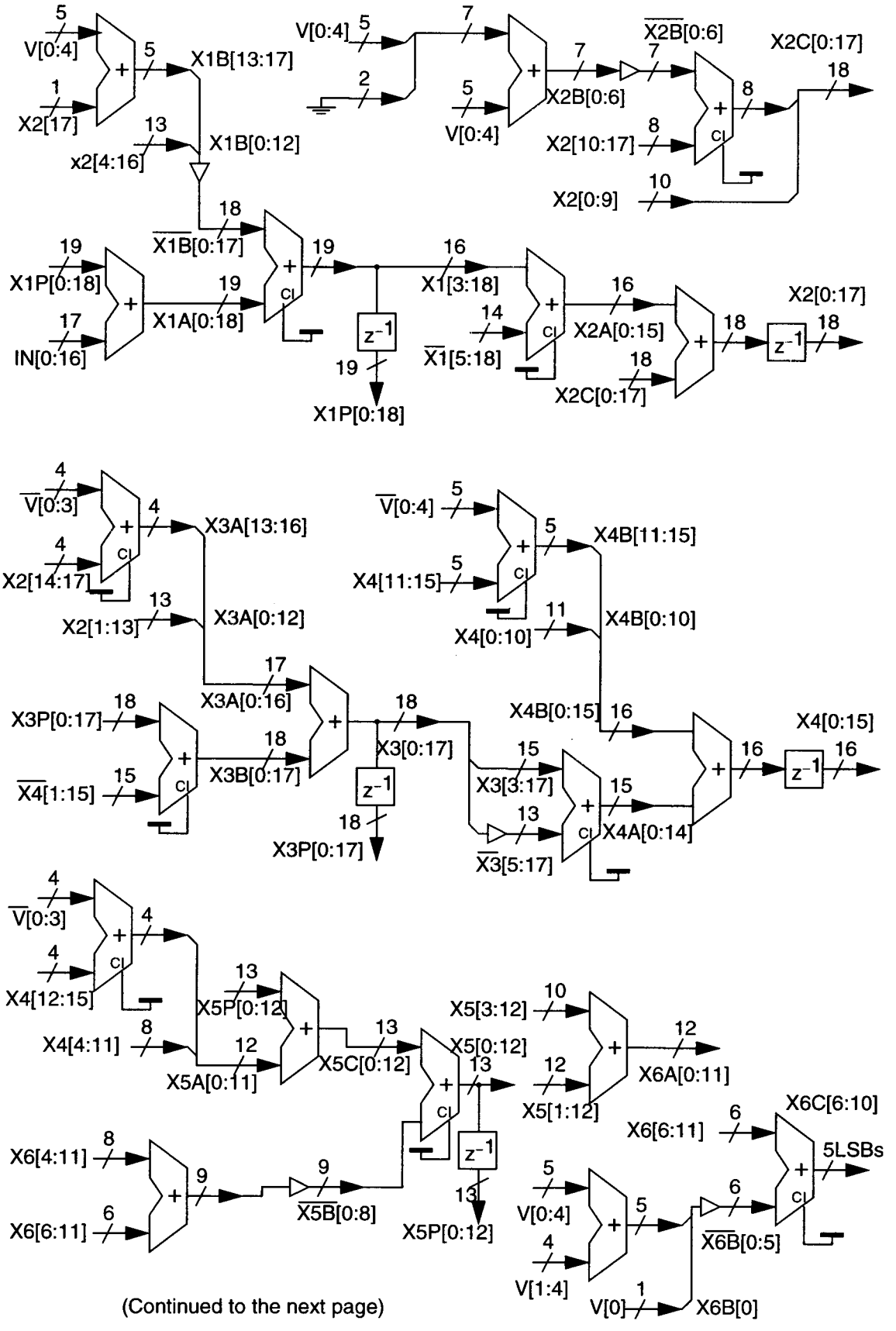


Figure A.1.8: Original 8th-order NTF optimized with $OSR = 8$.



(Continued to the next page)

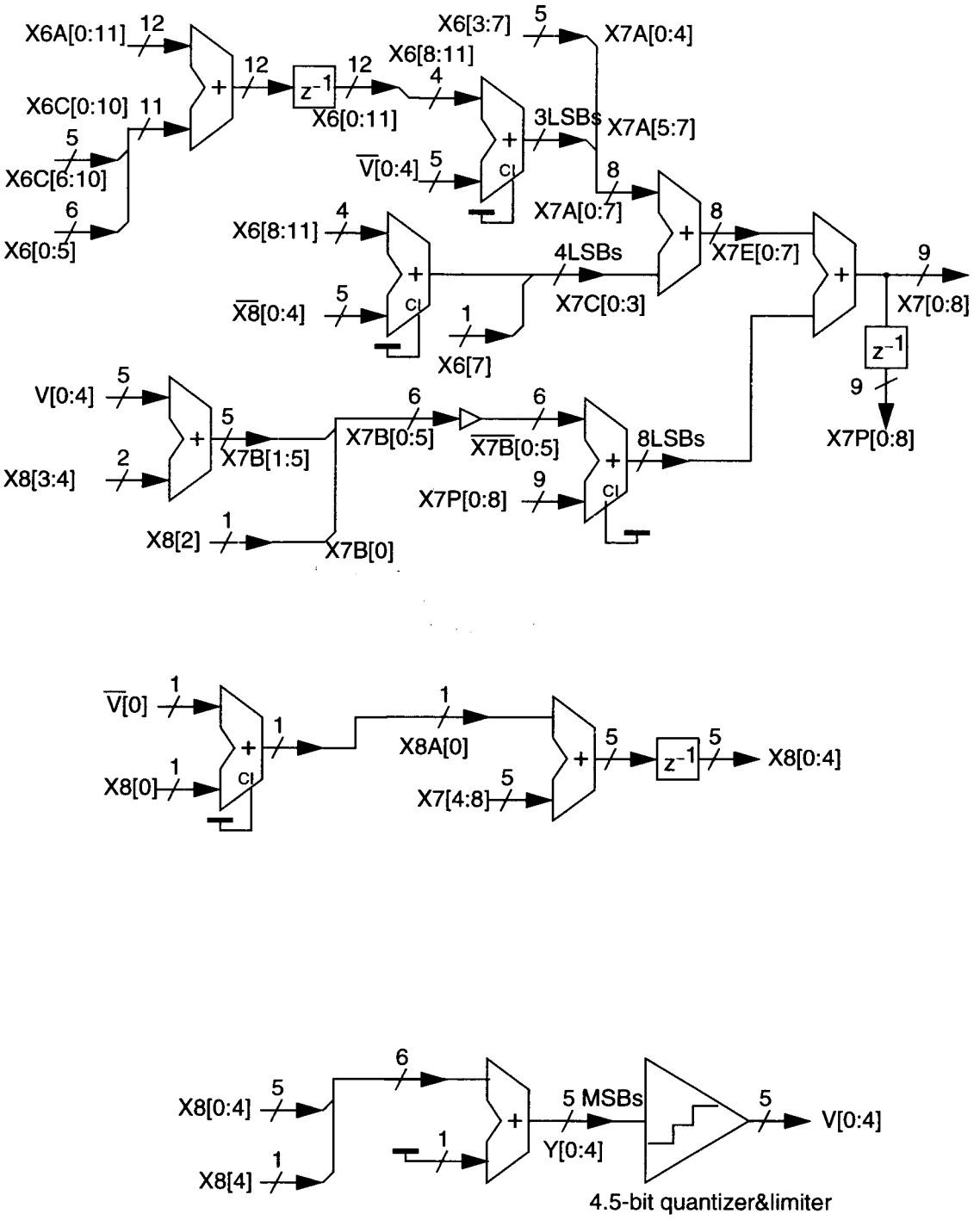


Figure A.1.9: Schematic of a 8^{th}-order 4.5-bit $\Delta\Sigma$ modulator optimized with $OSR = 8$.

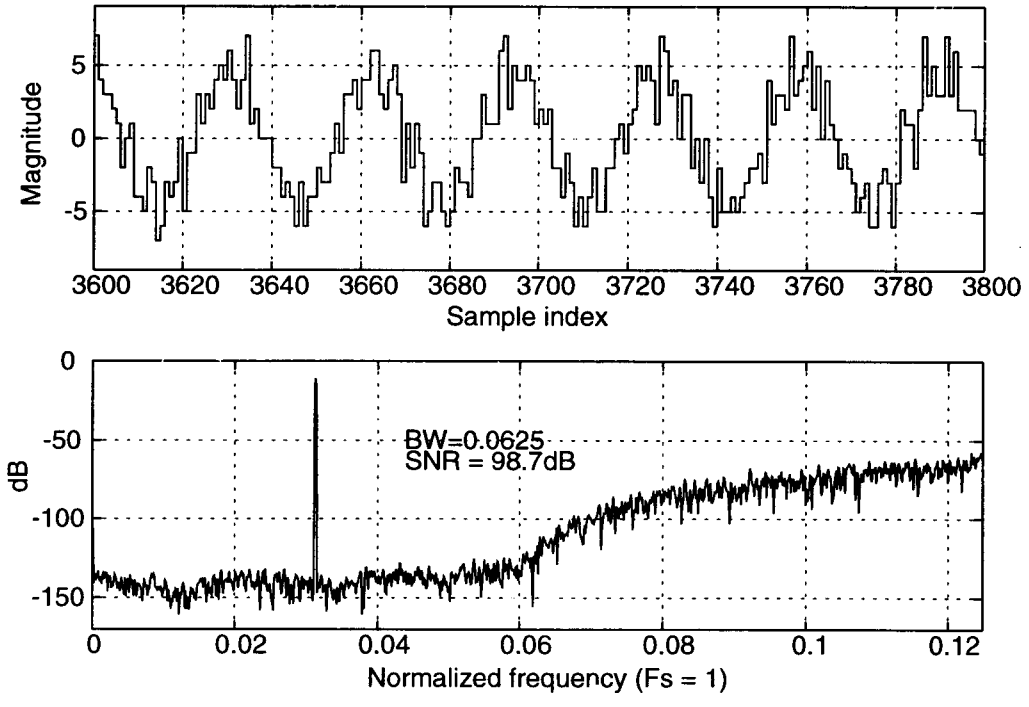


Figure A.1.11: Output waveform and spectrum of the 8^{th}-order modulator ($OSR = 8$).

Appendix II. Schematics of an 8-element 2nd-order ESL in VIEWDRAW Format

In Chapter 6, a 2nd-order ESL was implemented and used to drive the SC DAC under test. The design was accomplished using the POWERVIEW schematic entry tool, VIEWDRAW. A set of detail schematics of the 2nd-order ESL is attached. For descriptions of the design, please refer to Chapter 3 and Chapter 5.

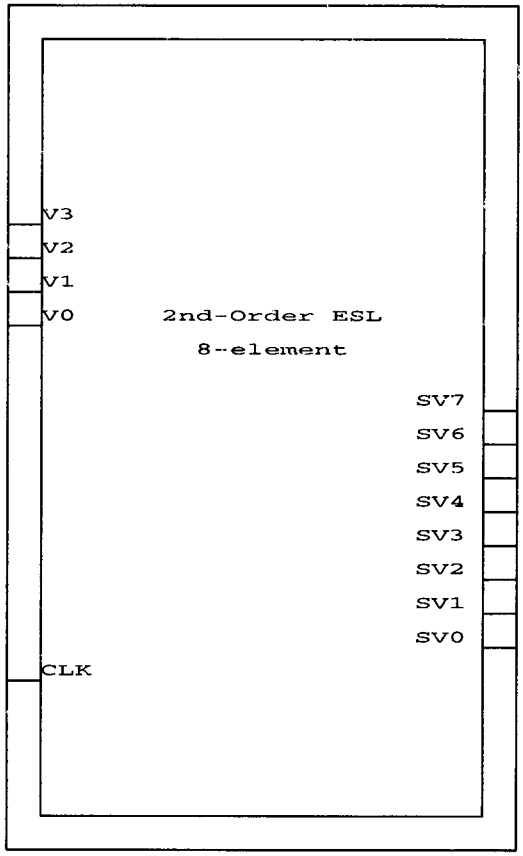


Figure A.2.1:The symbol of the 2nd-order ESL.

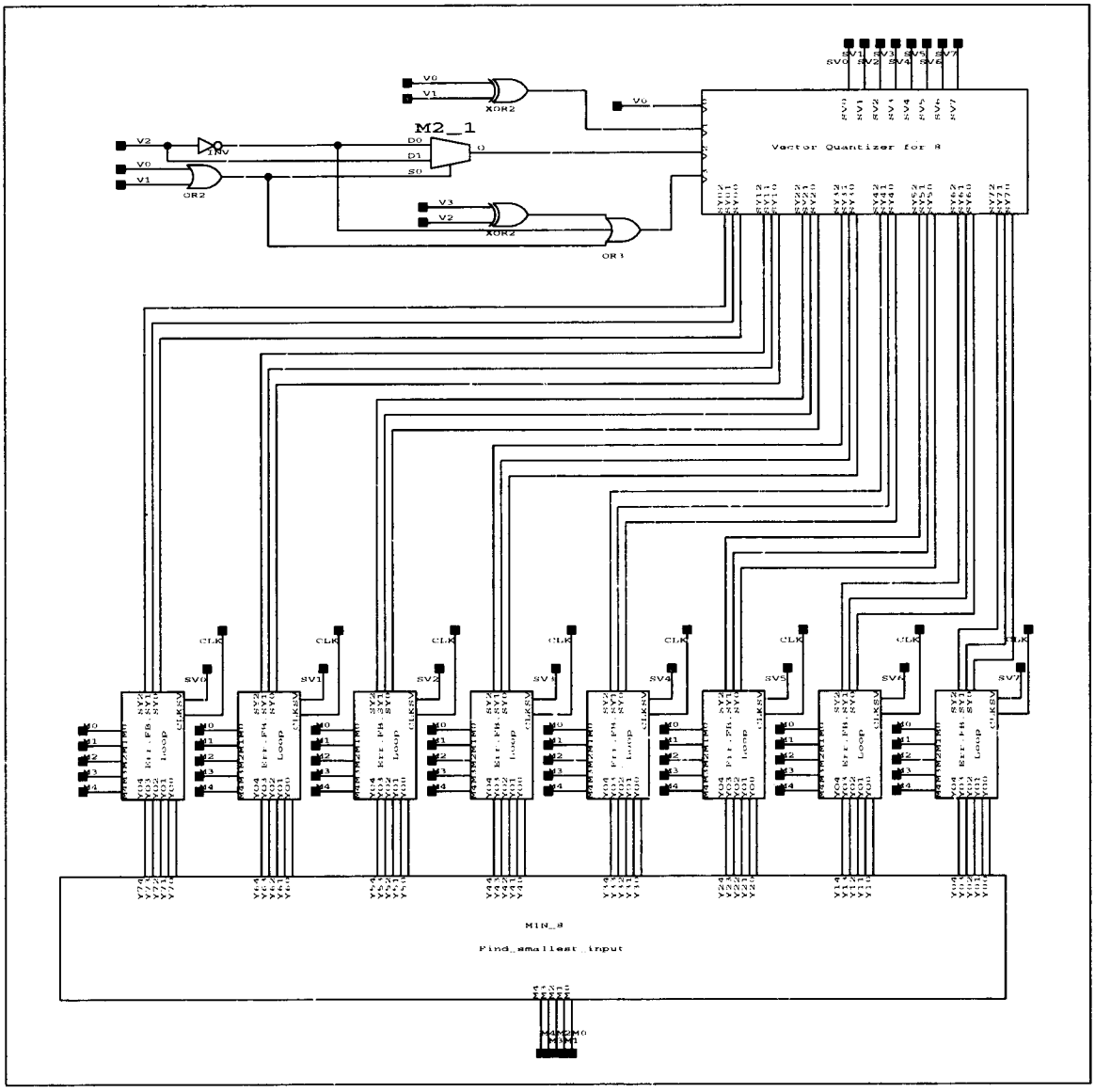


Figure A.2.2:Schematic of the 2nd-order ESL.

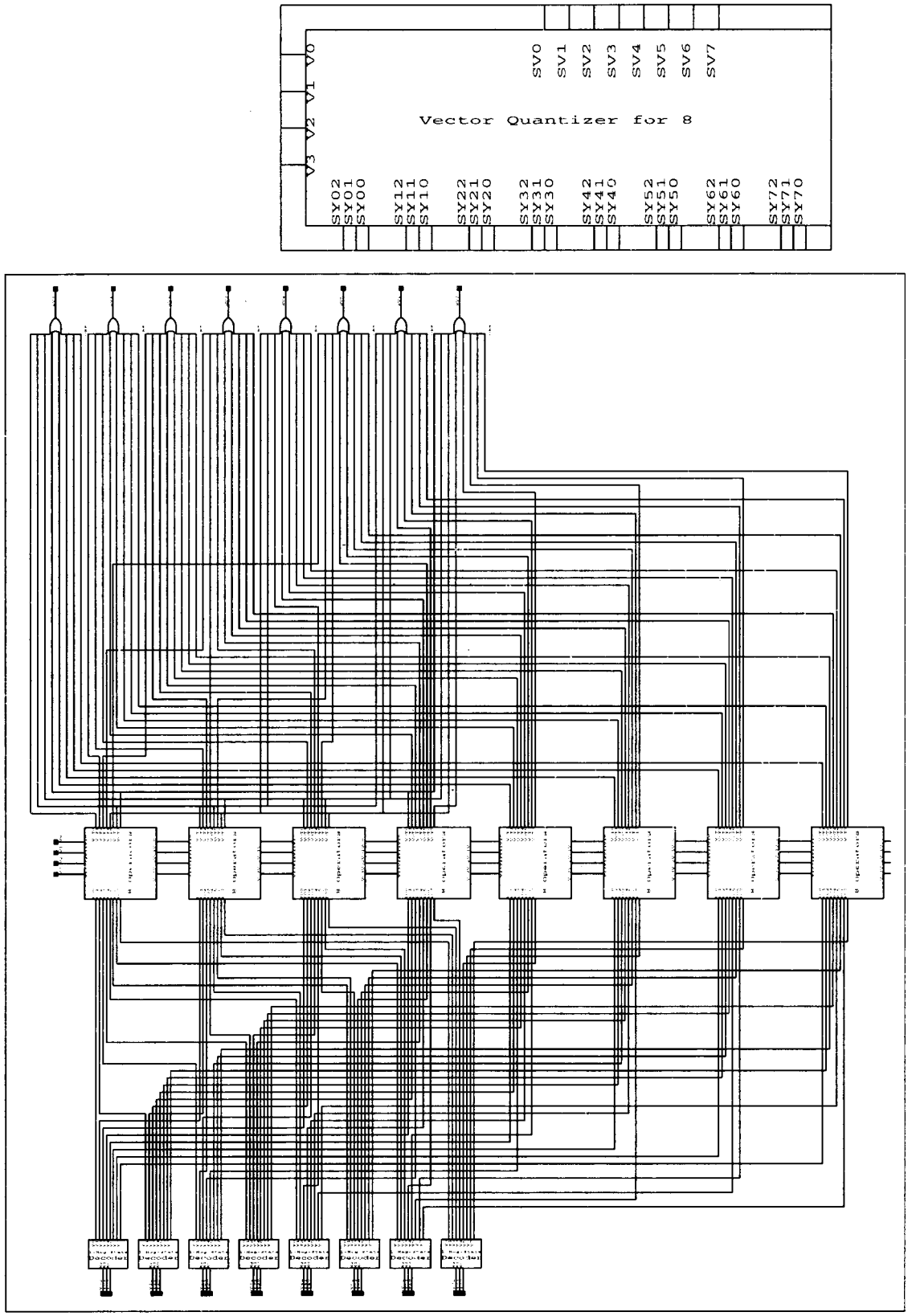


Figure A.1.3:Schematic and symbol of vector quantizer.

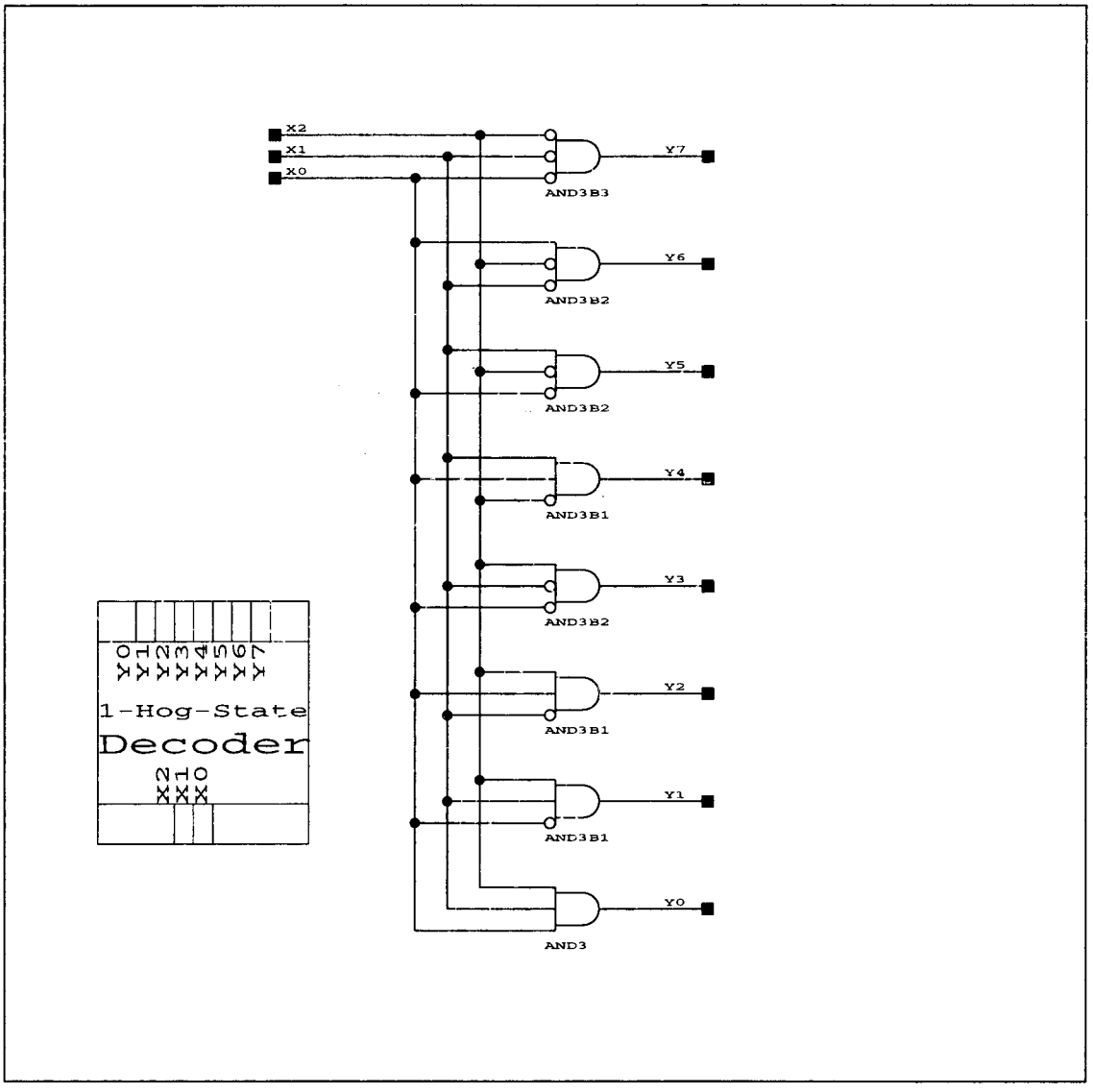


Figure A.1.4:Schematic and symbol of "single-hog-state" decoder.

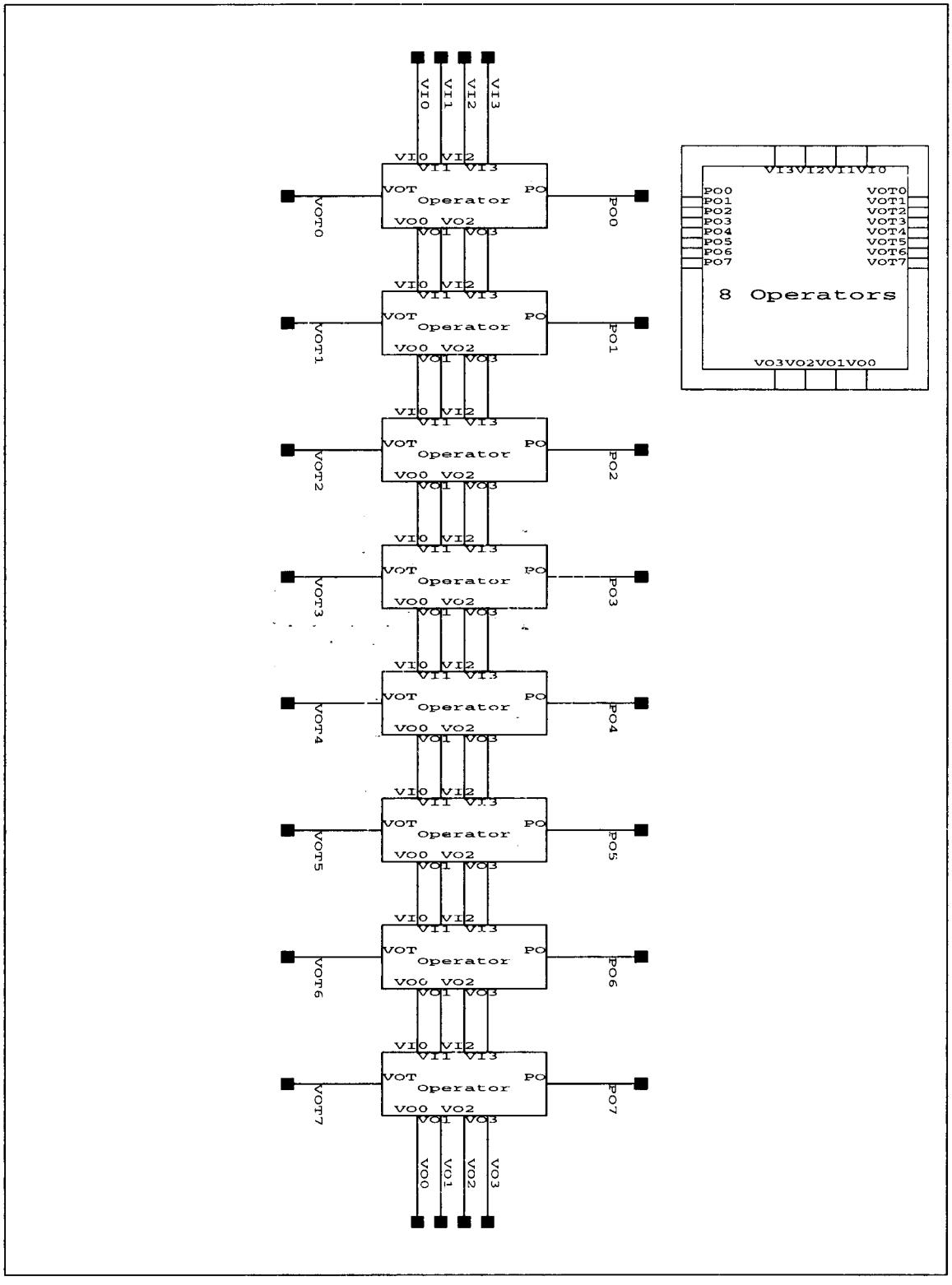
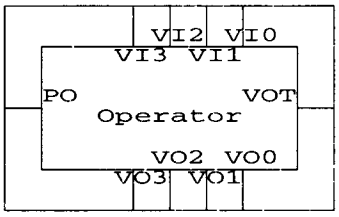


Figure A.1.5:Schematic and symbol of an operator group of 8.



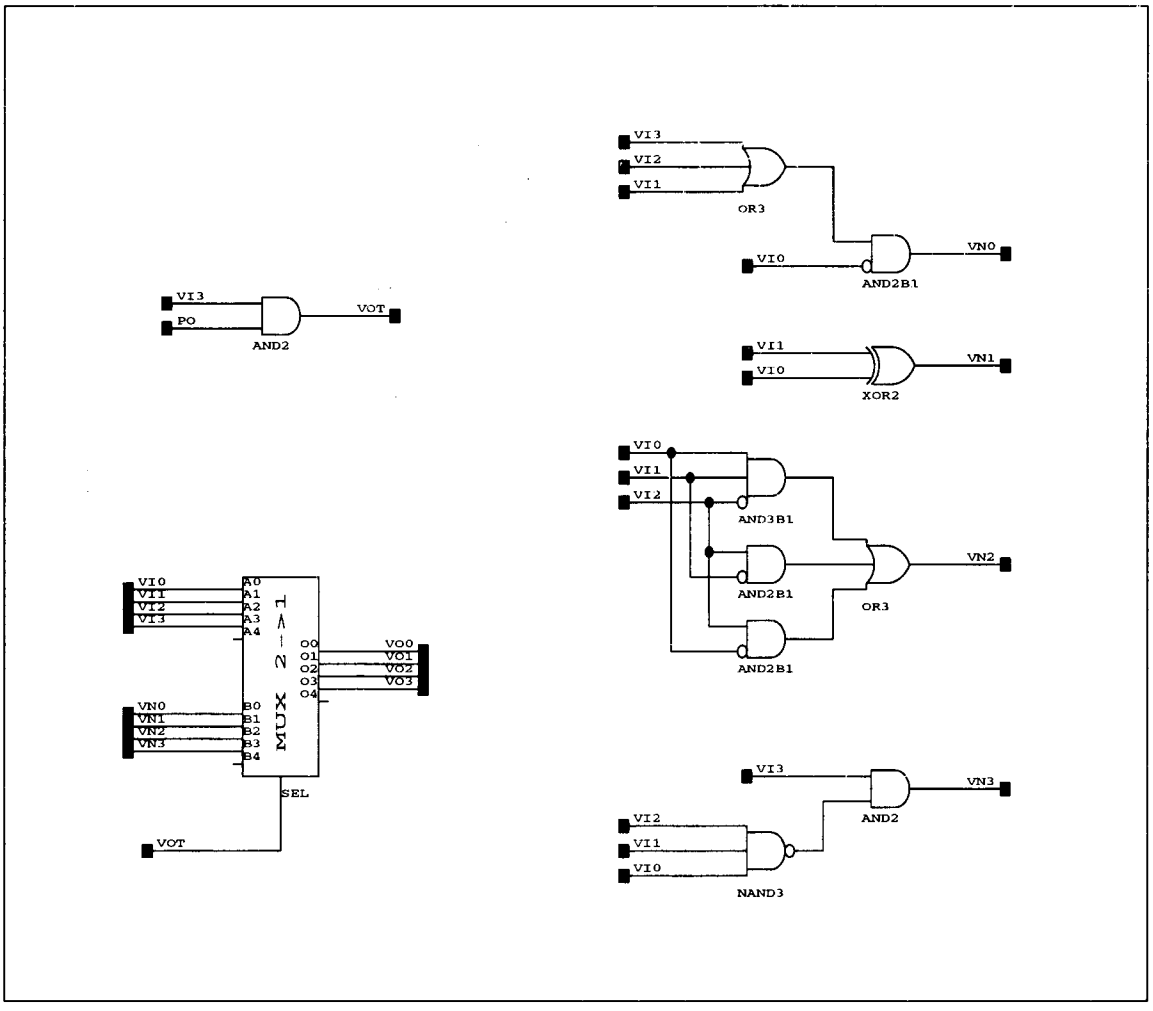


Figure A.1.6:Schematic and symbol of an operator cell.

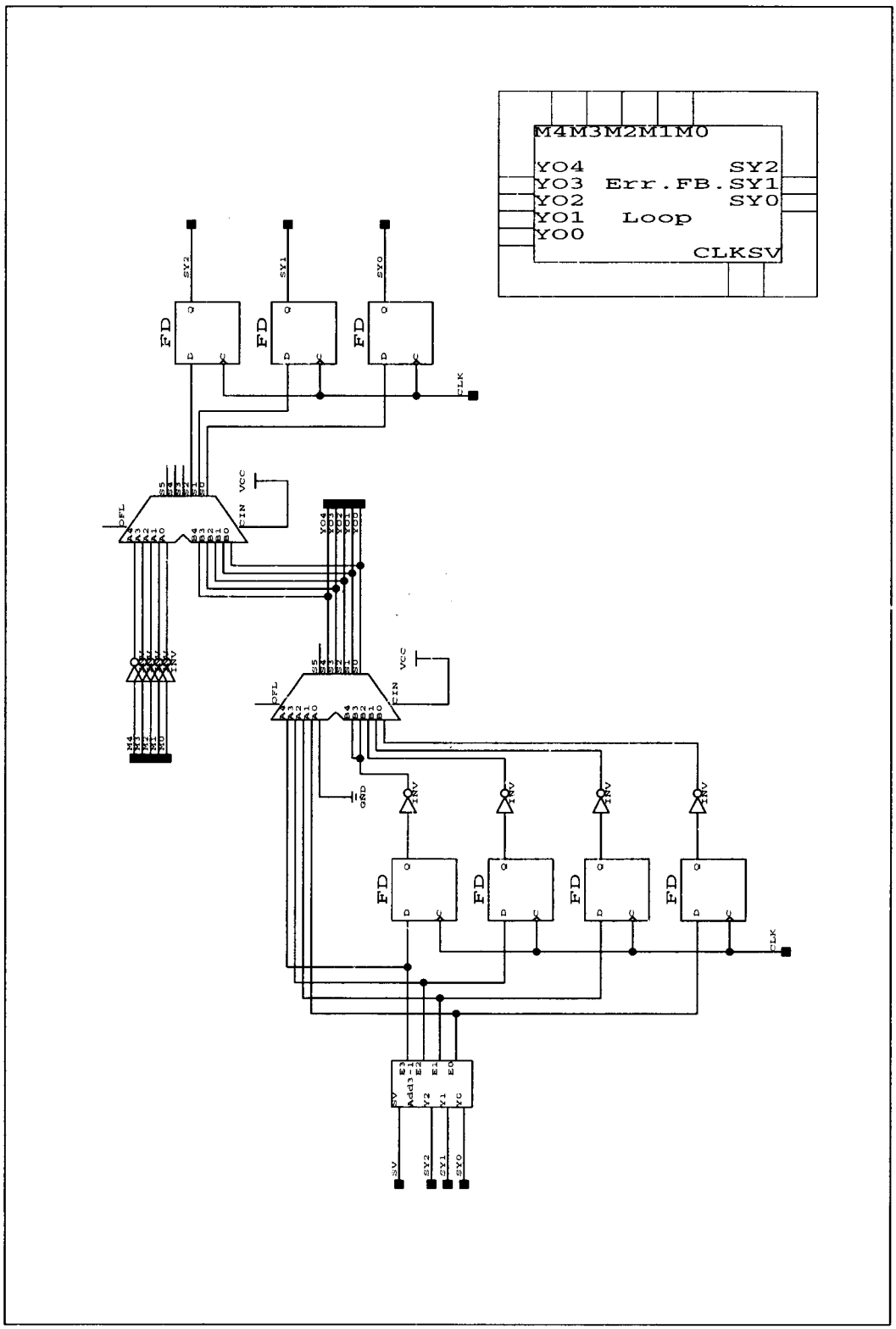


Figure A.1.7:Schematic and symbol of an error feedback loop.

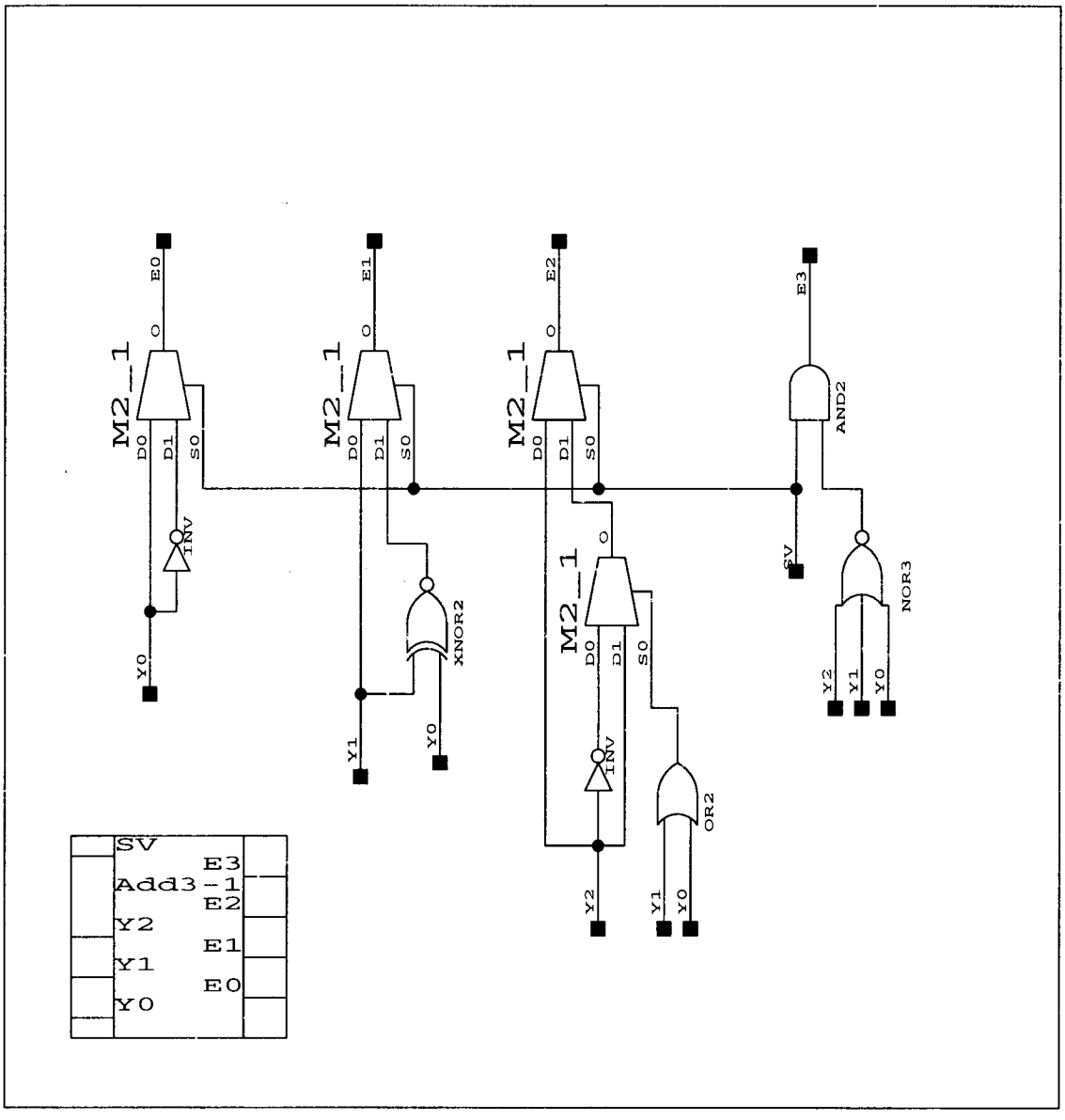
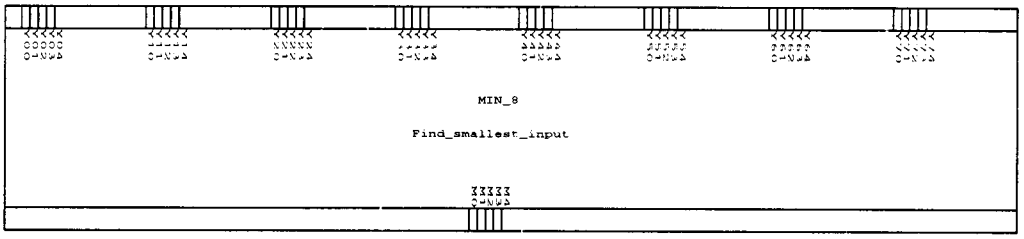


Figure A.1.8:Schematic and symbol of the "3-bit number minus 1"logic.



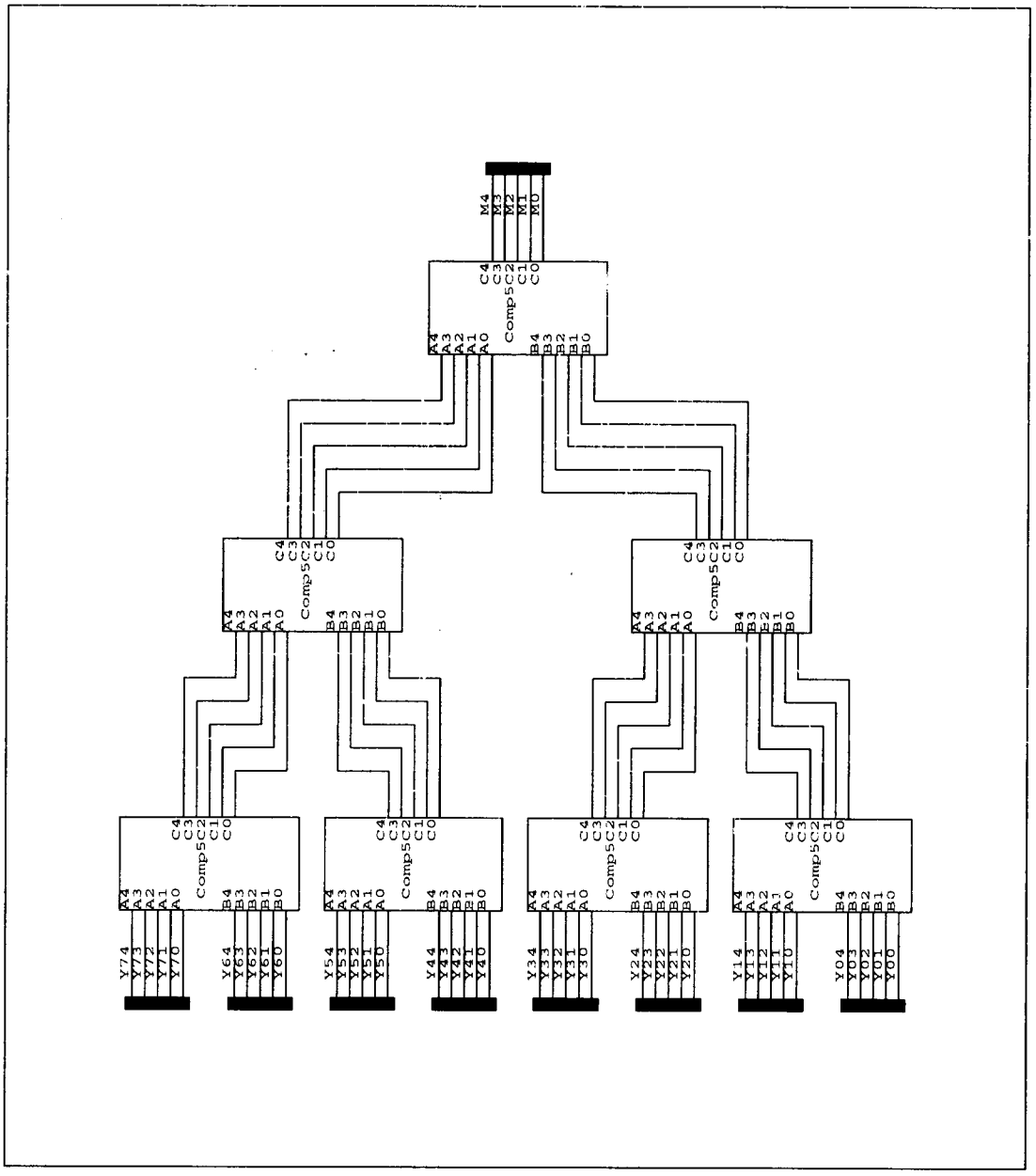


Figure A.1.9:Schematic and symbol of the 8-input minimum number selector.

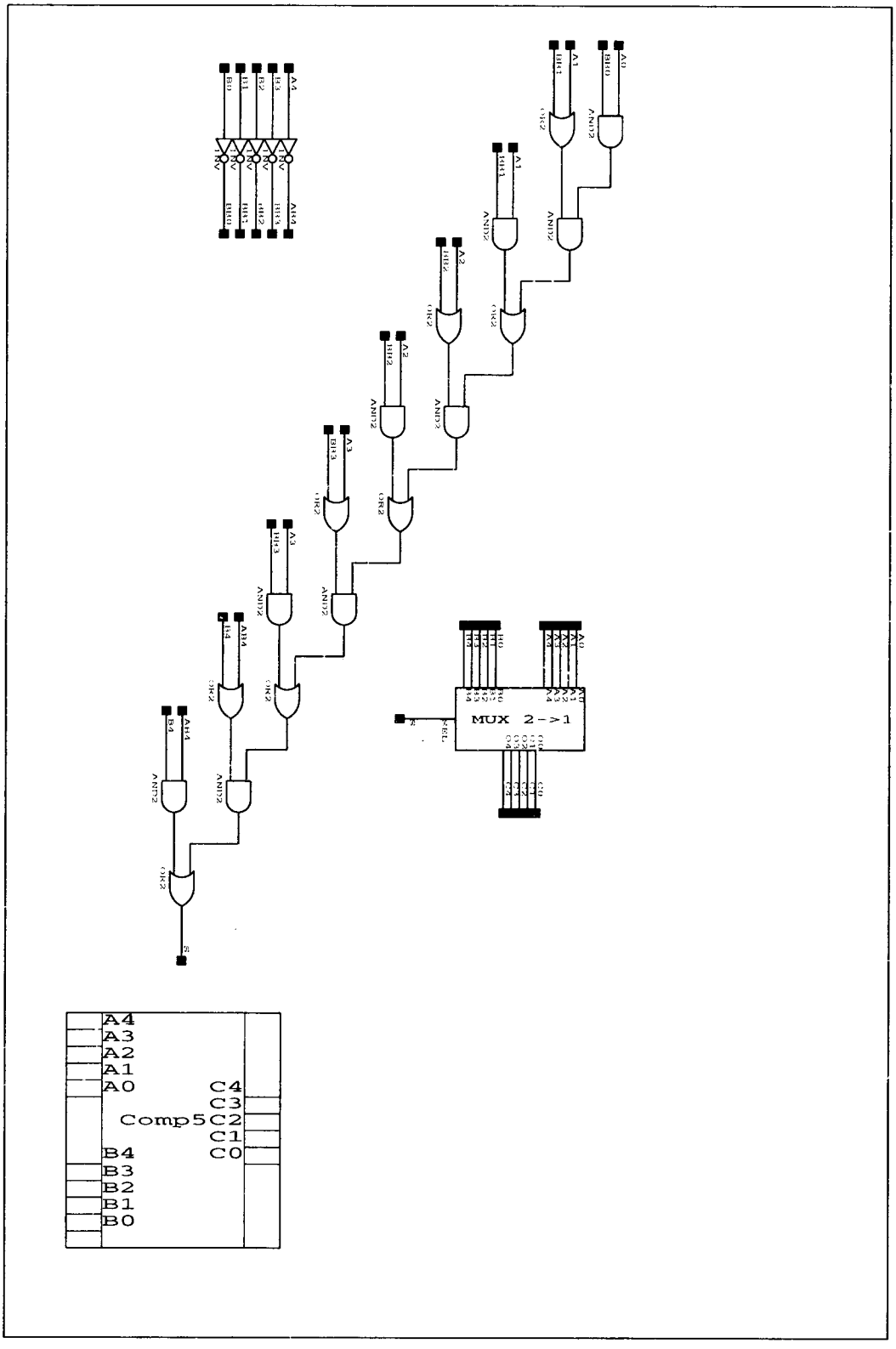


Figure A.1.10:Schematic and symbol of a 5-bit comparator.