UNIVERSITE DE NEUCHÂTEL
INSTITUT DE MICROTECHNIQUE

Oversampled digital leapfrog filters

THESE

PRESENTEE A LA FACULTE DES SCIENCES
POUR OBTENIR LE GRADE DE DOCTEUR ES SCIENCES

PAR

François Corthay

Neuchâtel
Juin 1992

IMPRIMATUR POUR LA THÈSE

Oversampled Digital Leapfrog Filters

de Monsieur François Corthay

UNIVERSITÉ DE NEUCHÂTEL

FACULTÉ DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel
sur le rapport des membres du jury,

Messieurs F. Pellandini, N. de Rooij,
P.-A. Farine et M. Kunt (EPF-Lausanne)

autorise l'impression de la présente thèse.

Neuchâtel, le 5 mai 1992

Le doyen :

A. Robert

Abstract

A new digital filter class is presented which is based on the hardware simulation of passive analog LC filters in a leapfrog manner. This filter class exploits a high oversampling in order to minimize the hardware of the devices and to restrict the side effects of the sampling on the frequency response. Also due to the oversampling, the filter state variables are coded on a single bit. The amplitude of signals is determined by the frequency of the true bits pulses. Likewise, coefficients are entered to the filter as pulse rates. The operators of the system are up-down counters, which act as integrators, and rate multipliers, which code the outputs of the counters back to a bit rate.

Two leapfrog synthesis methods have been proposed. They correspond to the ladder and the lattice structures. An analysis method allows to evaluate the effects of the quantization of both coefficients and variables. The integrated circuit implementation of a filter can be realized by the abutment of a small set of basic bit slices.

The proposed filters process pulse frequency modulated signals. The interfaces between this kind of signals and the analog or the digital world are performed by simple and reliable readily available building blocks. Due to the high oversampling, the oversampled digital leapfrog filters are limited to applications of band edge frequencies limited to some kHz. Domains of interest include the processing of pulse frequency modulated signals related to sensors and actuators such as quartz crystals and stepping motors. The limited signal bandwidth motivates the use of this kind of filters for applications such as the monitoring of thermal or mechanical processes up to the filtering of telephone signals.

Résumé

Une nouvelle classe de filtres numériques est proposée. Elle se base sur la simulation numérique d'un filtre analogique LC passif. Cette classe de filtres se distingue par un haut taux de suréchantillonnage qui permet de réduire le matériel nécessaire à sa réalisation et qui rend négligeables les distorsions de la réponse en fréquence dues à l'échantillonnage. Du fait de ce suréchantillonnage, les variables d'état du filtre sont codées sur un seul bit. L'amplitude des signaux est alors donnée par la densité temporelle des bits "1". Les coefficients eux aussi se présentent sous la forme d'un train d'impulsions. Les opérateurs du système sont des compteurs-décompteurs, qui agissent comme des intégrateurs, et des multiplieurs à taux programmable qui réencodent les sorties des compteurs en un train d'impulsions.

Deux méthodes de synthèse ont été développées. Ces structures sont dérivées des filtres en échelle et des filtres en traillis. Une méthode d'analyse permet d'évaluer les effets de la quantification des coefficients et des variables du filtre. La réalisation du filtre sous forme de circuit intégré peut se faire par le collage d'un ensemble réduit de tranches.

Ce type de filtres traite des signaux modulés en fréquence d'impulsions. Les interfaces entre ce type de signaux et aussi bien le monde analogique que le monde numérique sont réalisées par des blocs fonctionnels existant, simples et fiables. Du fait du haut taux de suréchantillonnage, la mise en oeuvre de ces filtres est limitée à des fréquences de coupure de quelques kHz. Parmi les domaines d'intéret, citons le traitement de signaux codés en fréquence associés à des capteurs et des actuateurs comme par exemple les quartz ou les moteurs pas à pas. La largeur de bande limitée de ces filtres permet leur mise en oeuvre pour des applications concernant des processus variant lentement comme des mesures thermiques ou mécaniques jusqu'au filtrage de signaux téléphoniques.

Table of contents

1 Introduction

Oversamplad Digital LeapFrog filtera (ODLFs) rely on the aimulation of an analog passive LC prototype in order to inherit its interesting propertiea. Tha aimulation of a reactance in its ladder expanaion ia performed by a chain of up-down countera and rata multipliera. The filter atate variablea and coefficients ara coded in tha form of a pulae frequency modulation.

In order to properly aimulata tha analog network, tha ODLFa require a high overaampling. Tha overaampling ratio ia determined by tha deaired signal to noiaa ratio or by tha allowed tranafer function deviation.

Tha work ia atructurad as followa: first tha ayatem and ita aignala ara presented, then tha aynthesis ahowa tha method used to derive an ODLF from ita analog paaaive prototype, tha analyaia atep allowa to determina tha required aampling rate to fulfill tha apecificationa and an integrated circuit realization method ia preaented. An exampla filter specification aet ia chosen to describe the completa deaign procedure of an ODLF.

1.1 Scope of ODLFs

1.1.1 Analog passive filter

Actual filtering techniques largely rely on the simulation of an analog passive LC filter [Nei81]. Indeed, although the advent of the transistor has widely enlarged the variety of signal processing systems, it pointed out a problem which had not occurred before: the first active filters suffered from a high sensitivity of the transfer function to their element values. Alternately, this problem disclosed the most interesting property of passive analog filters: they are extremely insensitive to variations of their element values.

The most important and most widely used passive analog filter is the doubly terminated reactance two-port [Tem77]. It is depicted in figure I.I. The two-port is exclusively made out of inductances and capacitances. One port, considered as the input port, is driven by a generator E_G of internal resistance R_G and the other port is considered as the output port and loaded with a resistor R_L.

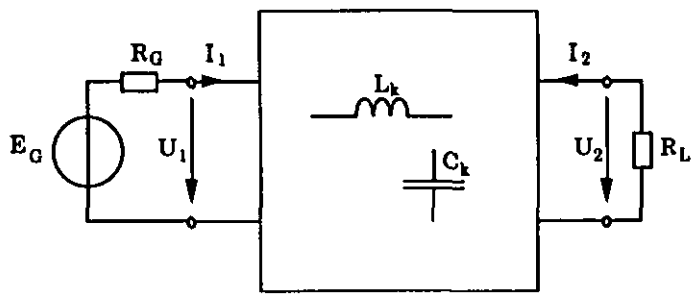


Figure 1.1 Lossless doubly terminated two-port.

The low sensitivity to the element values of the doubly terminated analog LC filters can be explained by considerations about the power transmission from the input signal generator to the output load [Tem77].

The most widely used passive reactance two-port is the ladder network. The ladder configuration is built from an alternative placament of a series and a shunt reactance. Figure 1.2 presents a special case of a ladder where the series reactances are inductances and the shunt reactances are capacitances.

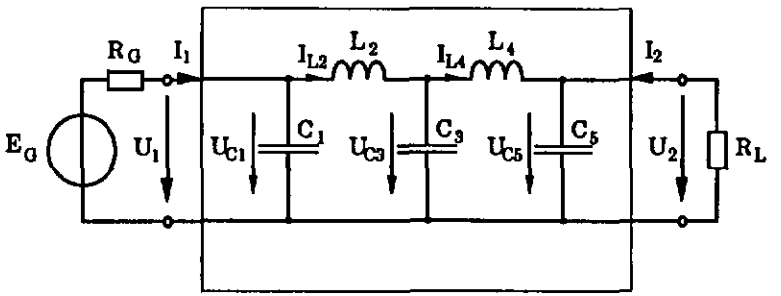


Figure 1.2 Doubly terminated LC ladder two-port.

In addition to a low passband sensitivity, the ladder two-port also proves to have a fine stopband behavior. This criterion favors the ladder configuration as the most interesting prototype candidate for an active filter design.

1.1.2 Active leapfrog filter

One classical active filter structure simulating a passive LC ladder prototype has been introduced by [Gir70]. In the proposed synthesis method, the reactances of an analog passive LC ladder prototype are simulated by integrators, in accordance to the relations between currents and voltages of the reactances:

$$\begin{cases} I_L = \frac{1}{sL} U_L \\ U_C = \frac{1}{sC} I_C \end{cases} \qquad (1.1)$$

The one-to-one correspondence between the elements of the passive filter and the integrators of the active system serves to duplicate not only the overall response of the filter but also the internal workings. This structural similarity ensures the active filter to inherit the interesting properties of the analog prototype. The synthesis method was named as LeapFrog (LF) due to the appearance of the feedback links in the schematic representation.

The LF structure simulating the ladder prototype of figure 1.2 is represented in figure 1.3. In the schematic representation, each integrator is followed by a multiplier setting the integrator gain. The multiplier coefficients are radian frequencies inversely proportional to the element value of the passive prototype.

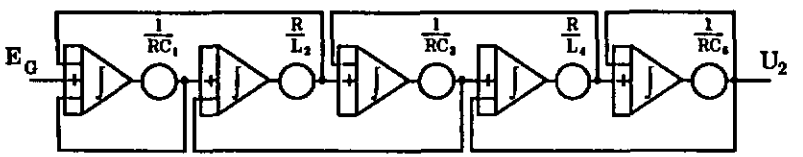


Figure 1.3 LF active filter.

Without the resistive generator and load, the ladder of figure 1.2 corresponds to the Cauer1 expansion of an LC impedance [Tem77]. This means that not only the *LC* ladder but also any reactance can be simulated by a LF structure.

1.1.3 Digital LF filter

The digital realization of the LF synthesis leads to different possible structures [Bru75]. In the Direct-transform Digital Integration (DDI) synthesis method, the integrators of the active leapfrog are realized by accumulators. The integrator gains of the active device are set by a digital multiplier.

Unlike the bilinear z-transform, the DDI z-transform does not map the s-plane imaginary axis onto the z-plane unit circle. As a consequence, the sampled transfer function is not a stretched replica of the time-continuous transfer function; the DDI z-transform results in the distortion of the transfer function in a manner similar to the standard z-transform.

The main effect of the DDI plane mapping consists in a band edge droop of amplitude depending on the ratio between the sampling rate and the band edge frequency [Bru75]. Additionally, the DDI z-transform does not produce a zero at the location z = -1. Thus, the stopband attenuation is limited by the periodicity of the transfer function.

Subsequently, the DDI synthesis method requires a high sampling rate compared to the band edges of the filter specifications. Yet its interest relies in the low coefficient sensitivity which leads to a worthy reduction of the multiplier coefficient wordlength.

1.1.4 ODLF

The non-idealities of the DDI structures depend on the sampling rate. Obviously, these effects vanish as the sampling rate tends towards infinity. Moreover, the increase of the sampling rate allows the coding of the internal filter variables with a reduced number of bits. The decrease of the number of bits of the signals as a function of the oversampling depends on the coding scheme.

ODLFs are DDI structures whose sampling rate is high enough to allow the coding of the filter variables on a single bit. The coding algorithm makes use of dithering. As the filter variables are represented with a single bit, the ODLF operators turn out to become even more moderate. The accumulators simplify to counters so that the differential input integrators of the analog active LF are implemented by Up-Down Counters (UDCs). As for the multipliers setting the integrator gains with their output coded to a single bit, they are efficiently implemented by Rate Multipliers (RMs). RMs are basically made out of a quantizer, which codes the bit-parallel input to a single bit, and a one bit multiplier which multiplies the quantizer output with the bit-rate input.

The ODLF realization of the analog active LF filter of figure 1.3 is thus obtained by replacing the integrators in the figure by UDCs and the multipliers by RMs. The corresponding structure is shown in figure 1.4.

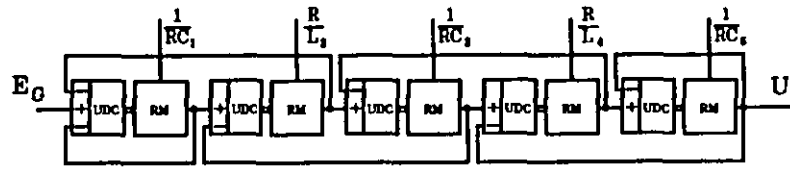


Figure 1.4 ODLF.

The ODLF state variables corresponding to the voltages and the currents of the analog passive prototype are found at the RM outputs. As they are coded on a single bit, they can be considered as pulse rates. In a similar manner, the ODLF coefficients are set by pulses of rate inversely proportional to the value of the reactance simulated by the UDC-RM set.

The relative placement of the UDC and the RM in the basic building block can be exchanged: the RM can be placed before the UDC. In this case, the RMs would have more than one bit-parallel input and the same number of single bit outputs. Moreover, the feedback lines would have to be replaced by buses. This dual configuration is more costly in terms of hardware and will generally not be considered in this work, although it could reveal of interest for systems having bit-parallel inputs and outputs.

1.2 Appliance of ODLFs

1.2.1 Oversampling ratio

ODLFs require a high oversampling in order to properly simulate the analog passive prototype. The minimal sampling rate f_s for signals limited to a baseband $[-f_b, f_b]$ is given by the sampling theorem:

$$f_s > 2 \cdot f_b \qquad (1.2)$$

In most of the common sampled filters, the sampling rate is then chosen as low as possible in order to reduce the filter order and the precision of the coefficients. Alternately, oversampled filters are characterized by a sampling rate several orders of magnitude higher than the baseband of the signal they process. The ratio between the chosen sampling rate and the minimal sampling rate is called the oversampling ratio:

$$osr = \frac{f_s}{2 \cdot f_b} \qquad (1.3)$$

The use of a high oversampling allows the coding of the signals with a limited set of values, usually with only two possible values. The frequency band between the signal baseband and the half of the sampling rate is occupied by a noise signal originated from the quantization of the signals.

In the ODLFs, the incremental behavior of the UDCs determine the minimal value for the oversampling ratio. Indeed, the UDCs act as integrators with a time constant proportional to the power of two of their number of bits. This means that assigning one supplemental bit to the UDCs requires to double the value of the sampling rate in order to maintain the frequency response of the filter unchanged.

1.2.2 Signal quantization

In common digital filters using fixed point arithmetic, the Signal to Noise Ratio (SNR), in dB, of a signal is proportional to the number of bits used to code the signal [Cou84]:

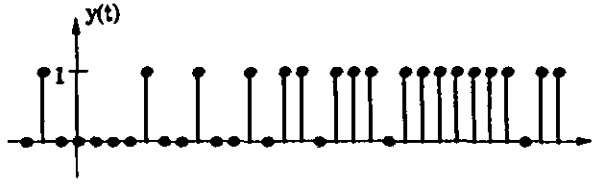
$$\xi_{q\,dB} \cong 6 \cdot n_{bits} \qquad (1.4)$$

Depending on the statistical properties of the signal, an offset has to be added to the preceding law.

In consequence, the SNR of a signal coded on a single bit seems useless for the use in digital signal processing. But as already mentioned, the SNR is improved by an appropriate oversampling algorithm. Oversampling quantization algorithms do not unequivocally assign one unique quantized output value to a set of input values: the choice of the quantized output value is also determined by the history of the signal quantization process.

Figure 1.5 provides an example of the binary quantization of a signal. The decision whether to assign a one or a zero to the quantized sample depends not only on the amplitude of the input signal but also on the decisions that have been taken before.

a) continuous time signal



b) signal coded to a single bit

Figure 1.5 Oversampled signal coding.

SNR improvemsnt due to oversampling is gensrally expressed in tsrms of bit per octave. Dithering bears 1 bit / octsve, which mesns that the SNR is incresaed by 6 dB se the sampling frequsncy ie movsd ons octsve higher or, in other terms, as the oversampling rstio is doubled. Single bit coding using error feedback loops resulte in mors efficisnt SNR enhancement. First order $\Sigma\Delta$ modulstors provids 1.5 bit / octsve and second order loops 2.5 bit / octave [Can85].

The UDCs require such a high oversampling so that they allow ths uss of the most simple quantizing algorithm. Hence ths SNR enhancement provided by the dithering inside tha RMs is sufficient to encode tha UDC outputs.

1.2.3 Transfer function deviation

Ths transfsr function dsviation of a DDI structurs compared to the lnitial filter approximation function has bsen shown [Bru75] to bs described by two non-ldealitiss: a bandedgs droop and a limitsd stopbsnd sttenustion. The other sources of transfer function deviation is ths quantizstion of the coefficients and vsriablss of the digital filter.

Dus to the high oversampling, both the bandedgs droop and the limited stopband sttsnuetion affsct ths trsnsfer function in a negligible manner. Howsvsr, the quantization of ths coefficients snd ths variables havs to bs examined thoroughly for the filtsr to meet the epecifications.

Cosfficisnt quantization changes ths shspe of the transfer function in a deterministic wsy. So cosfficient quantization can bs done in sn itersting loop as long as the obtsined transfer function still meets the specificationa. In contrast, the quantization of the vsriables doss not change the shape of the transfer function. This quantization is modeled aa ths superposition of noiss sources to ths filter varisbles and resulta in a deviation of the trsnafer function around its snslytic shape. The amplitude of the dsviation depends on the filter atructure snd on ths number of bits assigned to the operatora. Quantization of the variables is alao performed in sn iterating manner by adding bits to the operators until ths transfer function with its maximsl dsviation fits into the filter epecifications. For ODLFs, this is done at the expenss of a higher sampling rats.

1.2.4 Advantages and limitatlons

ODLFs sre digital devices simulating an snslog pasaivs filter. Ths ODLF cosfficients sre the inverses of the element vslues of the analog prototype. The simulation of tha internsl workings of tha paasivs filter enaures ths vary low sensitivity of ths psssband transfer function to coefficient values. The shape of the transfer function depends on tha relativs vslues of the coefficients. Ths amplituds of the coefficients determins ths magnituds of ths bsnd edge frequencies.

ODLFs csn bs considered ss a limiting csse of DDI filters. Due to the high overaampling, they srs insensitive to the transfer function deviation which ia the major limitation associated to the DDI ayntheais method.

From their DDI parent, they inherit the advantages of simplicity of design and minimal complexity in terms of hardware.

Furthermore, the high oversampling of ODLFs allows to simplify its operators to UDCs and RMs, leading to a drastic hardware reduction. The use of these basic operators make them a close relative to Quasi-Continuous Digital Filters (QCDFs) [Fer84], [Fer85], [Fer86]. The QCDFs also implement integrator based structures, but their signals are defined as pulse rates which are not synchronized to a sampling clock. This difference leads to a more difficult characterization of the QCDF signals and, above all, realization problems due to the asynchronous configuration of the pulses. As for the QCDFs, the ODLFs inputs and outputs are defined as pulse rates rather than binary numbers. This coding of the signals allows the use of inexpensive and robust Analog to Digital (A/D) and Digital to Analog (D/A) converters. Moreover, pulse frequency modulated signals are found in association with sensors and actuators. The coefficients, too, are pulse rates. As such, their value can be refined without this having any effect on the size of the corresponding multiplier.

The major limitation to the application of ODLFs consists in the high sampling rate. The sampling rate is proportional to the power of two of the required SNR, in dB, of the signals inside the filter. Hence, the use of high order ODLFs is best suited for filter requirements exhibiting sharp transition bands rather than for requirements with considerable stopband attenuation. Together with the simplicity of the operators, the high sampling rate refrains the relevance of time multiplexing for this kind of device.

1.3 Structure of the work

The specificity of the ODLFs has been introduced heretofore. The filter design task, leading from a system specification to a digital circuit realization, follows the same steps for any filter type and is sketched in Figure 1.6.

The same steps will be followed for the ODLF design. They are briefly presented hereafter and each one more thoroughly in a chapter of the work.

Chapter 2, "System and signals" presents the working of the ODLF operators and analyses the signals related to them. The chapter discloses the relation between the coding of operators outputs and the SNR of the corresponding signals.

Chapter 3, "Synthesis" shows how an ODLF can be derived from a ladder or a lattice passive LC analog prototype. The ladder synthesis method provides the most efficient implementation of an all-pole filter. For the more general case, the lattice synthesis is to be preferred.

Chapter 4, "Analysis" examines the effects of the quantization of the ODLF coefficients and signals. This step allows to determine the oversampling ratio necessary to ensure that the device meets the specifications.

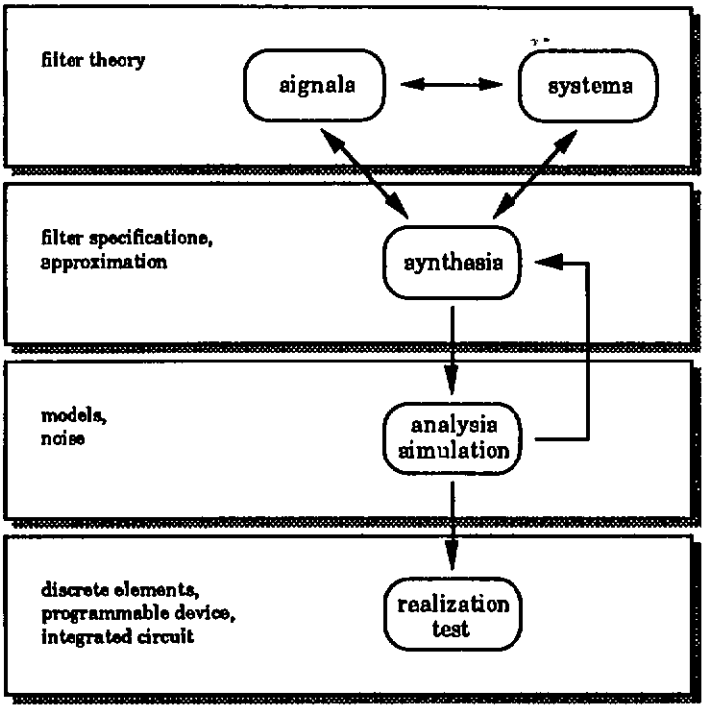


Figure 1.6 Filter design task.

Chapter 5, "Realization" puts forth the Very Large Scale Integration (VLSI) realization of ODLFs. The operators are built up out of bit slices abutted one to another. UDCs and RMs are also abutted together. Last, the feedback links are wired in the leapfrog manner.

Chapter 6, "Design example" describes a complete ODLF synthesis, all the way from the filter specifications to the VLSI realization.

2 System and signals

The UDC acts as an integrator of the LF structure. Its incremental behavior requires a high oversampling in order to generate a full scale signal.

The RM is used as a multiplier setting the integrator gain as well as a quantizer delivering an output coded on a single bit. The quantization algorithm makes use of dithering.

Besids integrators and multipliers, some LF structures require an adder or a subtracter. In ODLFs, this operator receives two bit rates as inputs and maintains data coded on only one bit at the output.

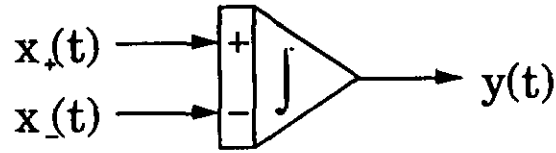
Input and output signals as well as coefficients of the ODLFs are coded as pulse rates. The rate is defined between 0 and the ODLF sampling rate and corresponds to an amplitude in the range [0, 1]. The coding and decoding of analog signals is simple and requires less hardware than A/D and D/A conversion.

2.1 The UDC as an integrator

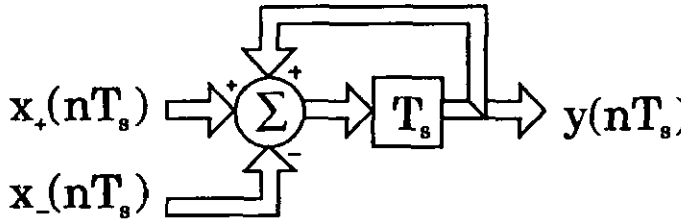
Inside the ODLF, the UDCs simulate the integrators of the analog LF structure. In this section, first the operator is described. From this, the transfer function of the UDC is determined in the time-discrete z-domain and, taking into account the high oversampling, the equivalent transfer function is expressed in the time-continuous s-domain. Last, the major limitation of the UDC as a signal processing operator is detailed in terms of an oversampling requirement.

2.1.1 The operator

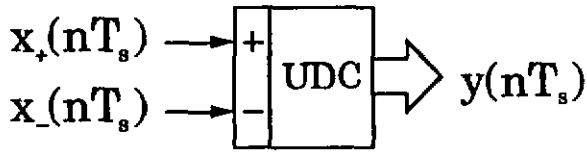
Figure 2.1 a) displays an integrator of the active LF structure. The LF integrators generally have a differential input. Using the DDI z-transform, the digital counterpart of this integrator is an accumulator with two inputs. In the special case where the inputs are coded on a single bit, the adder can be replaced by an incrementer/decrementer so the accumulator with two inputs is implemented by an UDC.



a) integrator with a differential input



b) digital accumulator simulating the integrator



c) accumulator with the inputs coded on a single bit

Figure 2.1 Up-Down Counter (UDC).

Figure 2.1 c) is the schematic representation of an UDC. The UDC inputs are ODLF state variables. These are coded on a single bit and can be interpreted as pulse rates. The UDC output is a bit-parallel binary number which will be modulated back to a single bit representation by a RM.

2.1.2 UDC transfer function

Here, the equivalent s-domain transfer function is found which describes the working of the UDC as an integrator. This is determined from the z-domain transfer function of the UDC under the assumption of a high oversampling.

The behavior of the accumulator of figure 2.1 b) is described in the z-domain by

$$z \cdot Y(z) = X(z) + Y(z) \tag{2.1}$$

where X(z) stands for the difference of the two inputs $X_+(z)$ and $X_-(z)$.

For the UDC, care has to be taken about the relative amplitude of the input and the output. Throughout this work, both are considered as normalized in the domain [0, 1]. In this case, an input $x(nT_s)$ of one will cause the incrementing of the output $y((n+1)\cdot T_s)$ by one Least Significant Bit (LSB) which corresponds to $1 / 2^{n_{bits}}$ of the total amplitude of $y((n+1)\cdot T_s)$. The UDC is thus characterized by

$$z \cdot Y(z) = \frac{X(z)}{2^{n_{bits}}} + Y(z) \tag{2.2}$$

where n_{bits} stands for the number of bits of the UDC. The transfer function is then given by

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{2^{n_{bits}} (z - 1)} \tag{2.3}$$

The complex variable z is defined as e^{sT_s}, where T_s is the sampling period. In the case of highly oversampled devices, sT_s is very small and the exponential can be approximated by the two first terms of its Taylor series

$$z = e^{sT_s} \approx 1 + sT_s \tag{2.4}$$

As a matter of fact, the preceding approximation corresponds to the definition of the DDI z-transform. Using this approximation, the transfer function of the UDC in the s-domain becomes

$$H(s) \approx \frac{1}{2^{n_{bits}} \cdot T_s \cdot s} = \frac{f_s}{2^{n_{bits}}} \cdot \frac{1}{s} \tag{2.5}$$

Under the above assumption of high oversampling, the UDC transfer function has been found as the one of an integrator with a fixed gain. This model of the UDC, symbolized in figure 2.2, will be used for the ODLF synthesis.

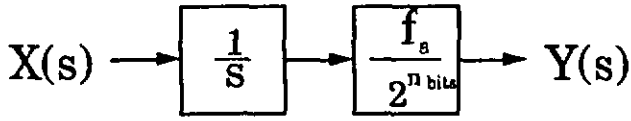


Figure 2.2 Transfer function of the UDC as an integrator.

The integrator time constant is proportional to the power of two of the number of bits of the UDC and inversely proportional to the sampling rate; this means that adding one bit to every UDC in a filter and doubling the sampling rate leaves the transfer function unchanged. The following discussion about the oversampling required by the UDC shows that the frequency $f_s / 2^{nbits}$ is greater than the bandwidth of the filter state variables, but remains of same order of magnitude.

2.1.3 UDC oversampling

The use of the UDC as a signal processing operator is restricted by its incremental behavior which leads to a limited bandwidth of its output. This constraint is a function of the sampling rate and and can so be expressed as a requirement in terms of oversampling.

This can be analyzed in the case of a full scale sine wave UDC output of frequency f_b equal to the maximal value allowed by the signal bandwidth. The limitation of the UDC arises from the fact that the values of two contiguous output samples differ at most of 1 LSB. In order to generate the given signal, the sampling period T_s must be small enough so that the maximal difference between two samples of the signal is smaller or equal to 1 LSB. Figure 2.3 presents the situation graphically.

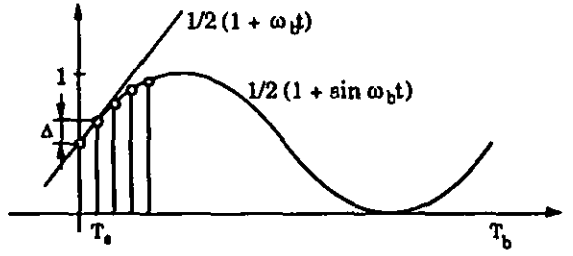


Figure 2.3 Sampled sine wave.

If Δ is the maximal difference between two samples of the sine wave of frequency f_b, it corresponds to

$$\Delta = \frac{1}{2}\omega_b T_s = \pi \frac{f_b}{f_s} \qquad (2.6)$$

The oversampling ratio is

$$osr = \frac{f_s}{2f_b} = \frac{\pi}{2\Delta} \qquad (2.7)$$

So the condition that Δ is smaller or equal to 1 LSB, this is $1 / 2^{nbits}$, requires an oversampling ratio of

$$osr \geq \frac{\pi}{2} 2^{nbits} \qquad (2.8)$$

The equation (2.8) points out the major limitation of ODLFs. The use of incremental devices requires a very high sampling rate compared to the signal bandwidth. This is compensated by the small hardware required by the operators.

2.2 The RM as a multiplier and quantizer

In the ODLF, the utilization of RMs is twofold. The RMs set the gain of the integrators of the analog LF structure and they code the UDC bit-parallel outputs back to a single bit. In this section, first the operator is described. From this, the transfer function of the RM is determined. Last, the quantization noise of the device is analyzed.

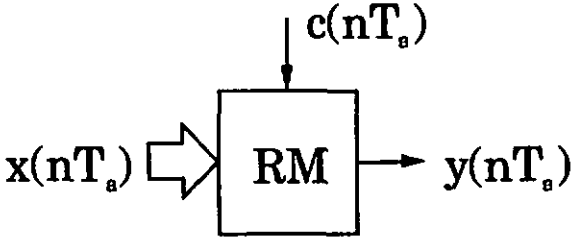
2.2.1 The operator

The RM has one bit parallel input and one bit rate input. The output is a bit rate corresponding to the bit rate input decimated according to the value of the bit parallel input. The RM exists as a logic integrated circuit such as the CD4089. As an ODLF operator, it is used together with an UDC. The bit parallel input comes from the UDC output whereas the bit rate input represents a filter coefficient.

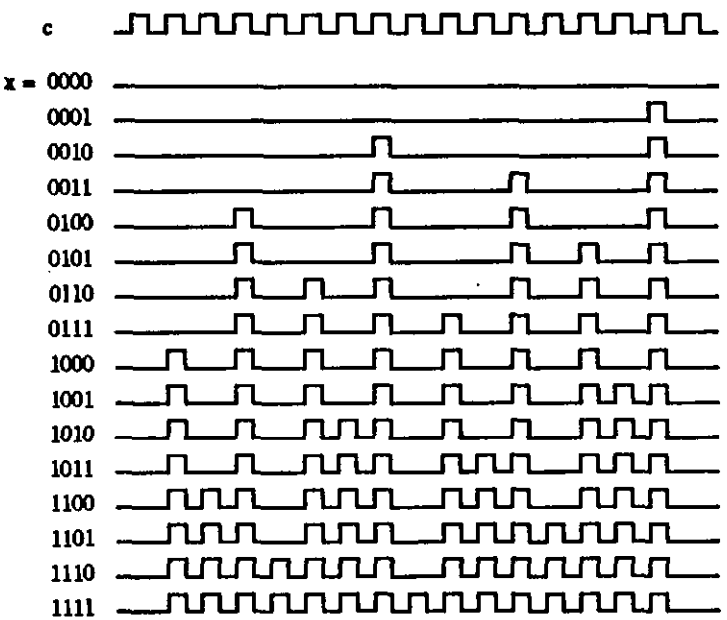
Figure 2.4 displays a RM and its behavior in the case of constant inputs. In figure 2.4 b), the coefficient pulse rate c is shown on the top. Underneath, the output of a 4 bit RM is displayed for every possible value of the input x. With constant inputs, the RM output is periodic of period 2^{nbits} cycles. With a zero value of x, the RM output remains zero. With z corresponding to 1/2, coded as 1000, the RM output pulse rate it the half of the input coefficient rate. The largest value of x, coded as 1111 and corresponding to the amplitude of $1 - 2^{-nbits}$, leads to an output rate where only every 1 of 2^{nbits} pulses of c is inhibited.

The logic waveforms of the RM described in this work are slightly different then the working of the CD4089. The proposed RM has been designed to show regular transitions for small variations of the bit-parallel input. Moreover, in ODLFs, the coefficient pulse rate is sampled, which means that it will not necessarily look as regular as the one presented in figure 2.4 b). For example, the pulse rate of a coefficient equal to c = 3/4 has the repetitive pattern of 3 samples of amplitude 1 followed by one sample of amplitude 0. With this, the period of the RM output with a constant input turns to be of the length of $2^{nbits}/c$ cycles. However, with a proper scaling of

the ODLF variables, the coefficient c is always greater than 1/2, which leads to a maximal period of $2^{(n_{bits}+1)}$ cycles.



a) schematic representation



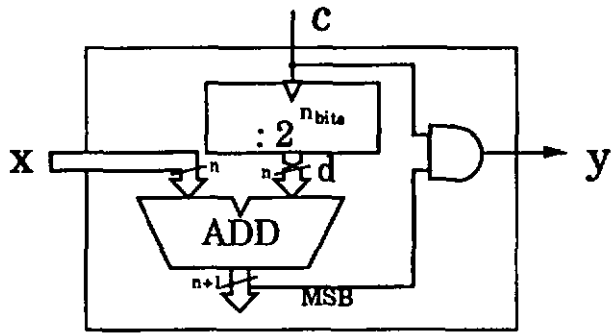
b) time responses of a 4 bit RM in the case of constant inputs

Figure 2.4 Rate Multiplier (RM).

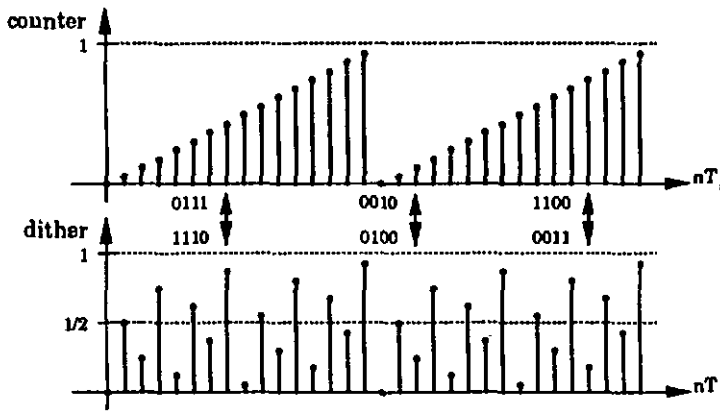
Figure 2.4 b) tends to indicate that the computation of the rate multiplication requires $2^{n_{bits}}$ pulses of c to provide a readable output. However, it should be noted that each individual RM output sample does makes sense at every clock cycle. It corresponds to the bit parallel input x quantized to a single bit. The length of the dither pattern determines the period over which the quantization errors towards 0 and towards 1 should have compensated and provides the base frequency of the lines spectrum of the dither pattern.

2.2.2 RM transfer function

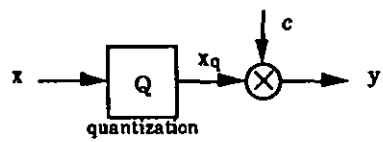
Here, the transfer function of the RM is found. It is determined from the working of the operator, depicted in figure 2.5.



a) functional diagram



b) dither signal



c) block diagram

Figure 2.5 Internal working of the RM.

The RM first quantizes the bit-parallel input $x(nT_s)$ to a single bit representation $x_q(nT_s)$. Then the signal $x_q(nT_s)$ and the coefficient $c(nT_s)$,

which are both coded on a single bit, are multiplied by a mere AND gate to deliver the RM output $y(nT_s)$. The corresponding block diagram is represented in figure 2.5 c).

The quantization is performed by adding a dither signal to the input $x(nT_s)$ and by keeping only the Most Significant Bit (MSB) of the sum. The dither pattern is a high frequency periodic signal $d(nT_s)$ which spectrum is distinct from the one of the input signal $x(nT_s)$ except for a DC component of 1/2, as sketched in figure 2.6 b). So, in a signal processing point of view, the addition of the dither pattern consists in the superposition of a DC value of 1/2 to the signal in its baseband. The sum of two samples, coded on n_{bits} bits in the interval [0, 1], is coded on $(n_{bits}+1)$ bits in the interval [0, 2]. Taking the MSB of the sum consists in comparing the sum to 1. In summary, the RM quantizer adds a DC component of 1/2 to the input $x(nT_s)$ in its baseband and compares the sum to 1; this corresponds to comparing the input $x(nT_s)$ to 1/2 which is actually is rounding quantization.

In the above mentioned process, a quantization error is added to the input signal $x(nT_s)$ but the amplitude of $x(nT_s)$ is left unchanged. The following multiplication is exact so the global transfer function of the RM does not show any scaling coefficient. Thus without taking into account the nonlinear effects of quantization, the transfer function of the RM can be expressed as

$$\frac{Y(s)}{X(s)} = c \qquad (2.9)$$

The quantization inside the RM is the main source of noise inside the ODLFs. Another quantization noise occurs inside the adder of the lattice structure.

2.2.3 RM signal quantization

The description of the working of the RM refers to a quantization operation. The effect of this quantization is analyzed now in terms of SNR and compared to the quantization of the bit parallel coding of the UDC output.

The quantization of the RM input to a single bit is a non-linear operation. As such, it can not be described in terms of a transfer function. The engineering approach to this problem is to model the quantization process by the insertion of a noise source at the place where the quantization occurs. The noise signal is defined as the difference between the signals after and before the quantization. Its spectral density is estimated from the quantization algorithm.

Hypothesis have to be made about the quantization noise sources. Their spectral shape are generally assumed to be white and as such the quantization noise is presumed to be uncorrelated to the signal. For digital signals coded with a fixed point representation, the energy of the quantization noise is estimated from equation (1.4)

$$\xi_{q\,dB} \simeq 6 \cdot n_{bits} \qquad (2.10)$$

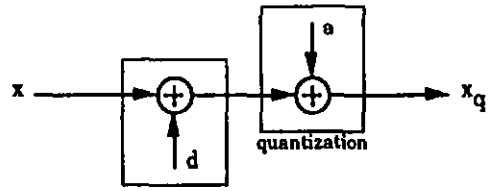
where n_{bits} stands for the number of bits used for the representation of the values of the samples. However, quantization to a single bit is a severe procedure and requires the two afore mentioned prerequisites to be reconsidered.

Quantizing a signal to a single bit using rounding or truncation obviously leads to a noise signal with a spectral content not white but very much correlated to the original signal. RMs overcome this problem using dithering. Dithering consists in adding a chaotic signal, called dither signal, to the input signal before the quantization operation. In this way, the quantization noise becomes quite independant of the input signal. Besides, the spectral content of the dither signal is high enough for that signal not to interfere with the input signal. Figure 2.5 b) illustrates how the dither signal is generated in the RM: each pulse of the bit rate input increments a counter and the dither signal is the counter output where the weight of the bits is reversed (the MSB is considered as LSB and vice-versa). This algorithm requires very little hardware and still has a high frequency content. As a matter of fact, the LSB of the counter, the MSB of the dither pattern, changes at each coefficient pulse so, as pictured in figure 2.5 b), the dither signal swings over and under 1/2 at each change. Figure 2.6 b) provides the expected spectre of the signals in the dithering process.

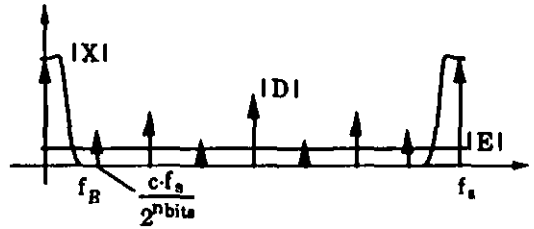
For a signal quantized to a single bit, the quantization error can be of same amplitude as the signal. However if oversampling is used, the signal remains limited to its baseband $[-f_b, f_b]$ whilst the noise spreads over a larger frequency band $[-f_s/2, f_s/2]$. In essence, the SNR in the whole frequency band remains unchanged with oversampling, but in the signal baseband the SNR is improved proportionally to the oversampling. As it is equal to 1 for an oversampling ratio of 1, the SNR of a signal coded on a single bit using dithering is equal to the oversampling ratio.

$$\xi_q = osr \qquad (2.11)$$

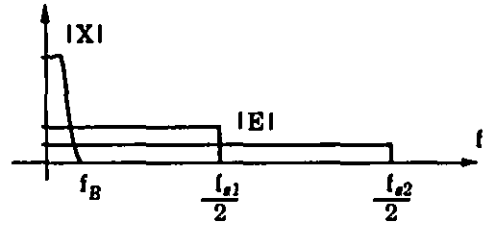
Figure 2.6 c) delineates the SNR enhencement due to oversampling. For the same signal X(s) limited to a baseband $[-f_b, f_b]$, if the sampling rate is doubled from f_{s1} to f_{s2}, the noise power is stretched in a twice as large frequency band and its amplitude is thus reduced by a factor of two throughout the frequency band.



a) dithering



b) spectra of the signals



c) oversampling and quantization

Figure 2.8 RM signal quantization.

The SNR of the RM dithering can be compared to the SNR associated to the signal coding of the UDCs. Equation (2.8) states that the incremental behavior of the UDC requires an oversampling ratio greater than $2^{n_{bits}}$. With this, the SNR, in dB, of the dithering can be rewritten as

$$osr \geq 2^{n_{bits}} \Rightarrow \xi_{q\,dB} = 20 \cdot \log_{10}(osr) \geq 6\, n_{bits} \qquad (2.12)$$

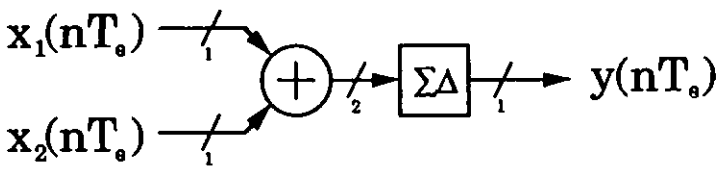
Equation (2.12) shows that the SNR of the dithering is of same order of magnitude and is even greater than the SNR of the bit-parallel coding associated to the UDCs in the ODLFs. It should be noted that the UDCs, although possessing a quantized output are no source of quantization, left out the effects of overflow. Moreover, although the UDCs also are oversampled, they take little advantage of the oversampling to code their output due to their integrator-type transfer function.

A more efficient single bit coding scheme than dithering is used by ΣΔ modulation [Cha90]. This method consists in feeding the quantized output back into the quantizer. As a result, the transfer function of the quantization noise to the quantizer output is no more equal to unity. The shape of the noise transfer function is of highpass type. For this reason, more of the noise power is retrieved out of the signal baseband providing so a better SNR in this band. Nevertheless, together with the oversampling required by the incremental behavior of the UDCs, the SNR supplied by dithering is sufficient to maintain the quality of the RM inputs, as can be seen by equation (2.12). Dithering coding scheme results into a smaller hardware and into less high fraquency noise which can induce limit cycles in the LSBs of the counters.

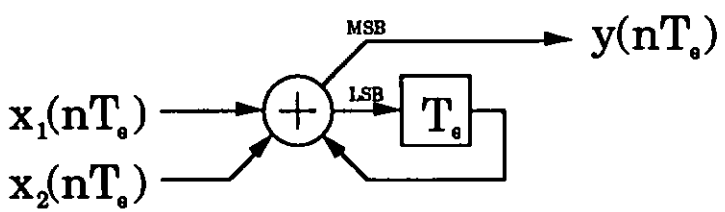
2.3 Bit rate adder and subtracter

The need for an adder or subtracter can occur in ODLF structures. For example the first UDC of the ladder ODLF depicted in figure 1.4 has one positive and two negative outputs; this can be implemented by an adder calculating the sum of the negative inputs followed by an UDC with one positive and one negative input such as the other UDCs of the chain. Also in the lattice structure, such as the ODLF of figure 3.11, the output is determined as the difference between the two branches.

The sum or the difference of two signals coded on one bit normally require two bits to maintain all the information of the input data. However, as the bit rates are known to be oversampled, it is possible to code the output back on a single bit. This is done by placing a first order ΣΔ modulator at the adder or subtracter output [OLe90].



a) bit rate adder structure



b) bit rate adder realization

Figure 2.7 Bit rate adder.

The transfer function of the bit rate adder shows a scaling of 1/2. Indeed, the output of the adder is in the range [0, 2] but is assumed to be normalized in the range [0, 1] by the ΣΔ modulator.

$$y = \frac{1}{2} \cdot (x_1 + x_2) \cdot \qquad (2.13)$$

The bit rate subtracter is realized by placing an inverter at the negative input of the operator [OLe90]. Care should be taken of the fact that this operator implies the use of signed arithmetic. Here too, the output is scaled by 1/2.

2.4 ODLF input and output signals

2.4.1 ODLF input

The ODLF input is fed into an UDC and is thus in the form of a pulse rate. From the analog world, a signal can be shaped into a pulse rate using a voltage to frequency converter which requires less hardware then an A/D converter. Actually, some A/D converters make use of a voltage to frequency converter as a building block. Besides, information also happens to be found directly in the form of pulse rates, as it is the case for quartz sensors. Finally, a digital signal is transformed to a pulse rate by a rate multiplier or eventually a ΣΔ modulator.

As a pulse rate, the ODLF input is defined positive. Eventually, an offset has to be added to the original signal in order to encode it. The pulse rate is synchronized to the ODLF clock and then sampled. After the sampling, a series of zeros corresponds to the amplitude 0 and a series of ones corresponds to 1. Any value between these bounds is coded by an series of alternating zeroes and ones. The amplitude is determined by the relative amount of zeros and ones. In summary, the input signal is coded as a pulse rate varying between 0 and the ODLF sampling frequency. This rate is considered as the signal amplitude ranging in the domain [0, 1].

2.4.2 ODLF output

The shape of the ODLF output depends on the chosen structure. The output of a ladder *ODLF*, as the one of figure 1.4, is in the form of a pulse rate. The output can also be read from the last UDC of the chain; this data is proportional to the bit rate output. The lattice ODLF output is the difference between two pulse rates. This value can be coded on two bits and eventually modulated back onto a single bit.

The *D/A conversion of a signal coded on a single bit* is done by switching between a reference voltage and a zero voltage. This analog output should then be lowpass filtered in order to separate the signal from the quantization noise. A pulse coded output can be of interest for the transmission of a signal in a noisy environment. It may also be of interest in some applications such as stepping motor control. If a digital output is required, it can be obtained by the decimation of the binary pulses.

3 Synthesis

The ODLF synthesis relies on the simulation of an analog passive filter. The relevant feature of the analog passive filter is its low sensitivity to variations of its element values.

In the case of all-pole filters, the LC ladder prototype leads to a straightforward active filter implementation characterized by a low sensitivity to coefficient values also in the stopband.

If the filter owns not only poles but also zeroes, the simulation of the LC lattice prototype reveals to be preferable. This structure also allows to easily generate a complementary filter output. An alteration of this structure allows the simple generation of allpass filters.

3.1 Analog passive reference prototype

Actuel filtering techniquee largely rely on the simulation of an analog passive LC filter [Nei81]. Indeed, although the advent of the transistor has widely enlarged the variety of signal processing systems, it pointed out a problem which had not occurred with analog filters: the first operational amplifier active filters suffered from a high sensitivity to their element values. The low sensitivity to the element values of the doubly terminated analog LC filters can be explained by considerations about the power transmission from the input signal generator to the output load.

The most important and most widely used passive analog filter is the doubly terminated reactance two-port [Tem77]. It is depicted in figure 3.1. The two-port is made out of inductances and capacitances. One port, considered as the input port, is driven by a generator E_G of internal resistance R_G and the other port is considered as the output port and loaded with a resistor R_L.

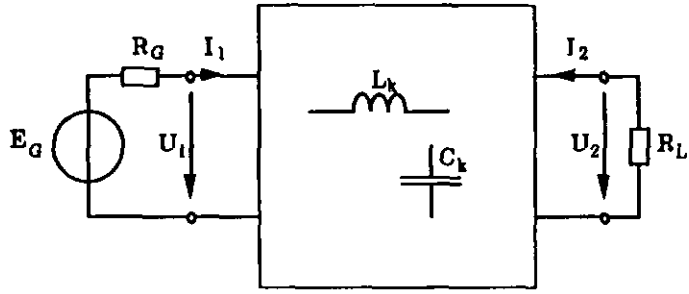
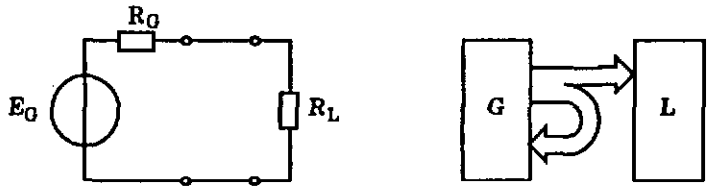


Figure 3.1 Lossless doubly terminated analog filter.

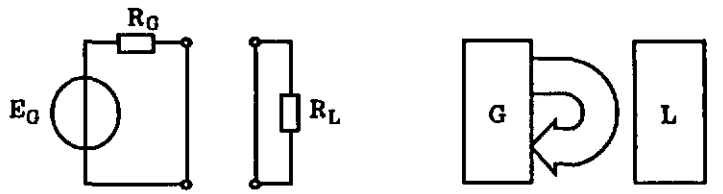
The LC network of the two-port is lossless: it does not dissipate any power. The power of the generator is dissipated in its internal resistance R_G and in the load R_L. If the input signal E_G lies in the filter passband, the two-port drives the power of the generator to the load, as shown in figure 3.2 a). Then, as it is well known, the maximum ratio of the power delivered by the generator can be dissipated in the load resistance if the source and the load resistances are matched:

$$R_L = R_G \qquad (3.1)$$

In this case, the generator and the load are said to be matched and the transmitted power is equal to the half of the maximum amount of power delivered by the generator. If the input signal E_G lies in the filter stopband, the two-port acts roughly as a short circuit on the generator and all of its power is dissipated in the internal resistance. This case is represented in figure 3.2 b).



a) power transmission in the passband



b) power transmission in the stopband

Figure 3.2 Power transmission from the generator to the load.

The lossless doubly terminated two-ports have a very little sensitivity to variations of their element values in the passband [Tem77]. Indeed, the power transmission is maximal when the filter elements have their nominal values. If one element value is either increased or decreased, the perfect match between the parts of the system will be disturbed and the output will be decreased. So at the frequencies where the power transfer is maximal with the nominal values of the elements, the sensitivity to changes of the elements around their nominal values is zero. Thus, a matched system consisting of a lossless two-port and its resistive terminations has a natural immunity against the variations of any of its elements in the passband. Singly terminated or unterminated networks do not have this property.

Two types of analog devices will be considered hereafter as prototypes: the ladder and the lattice LC filters. Both lead to the synthesis of LF filters which are canonical in regard to the number of integrators and of multiplying coefficients.

3.2 Active ladder synthesis

The most commonly used passive reactance two-port is the ladder filter. Figure 3.3 displays a fifth order lowpass ladder filter. The relation between this network and the model described in figure 3.2 is straightforward: for a DC signal, the inductances act as short circuits and the capacitances as open circuits so the device corresponds to the circuit of figure 3.2 a); for very high frequency signals, the inductances act as open circuits and the

capacitances as short circuits so the device corresponds to the circuit of figure 3.2 b).

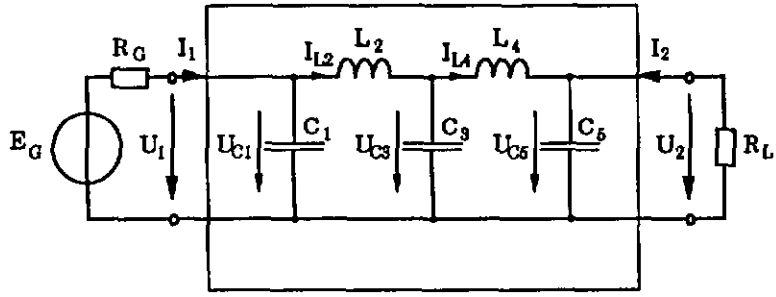


Figure 3.3 LC ladder two-port.

The simulation of the passive analog ladder filter using integrators has been described by [Gir70]. It is called the active ladder or LeapFrog (LF) synthesis. The internal workings of the analog prototype are described by the integral equations relating the voltages and the currents of the capacitances and the inductances:

$$\begin{cases} sC_k \cdot U_{Ck} = I_{Ck} = I_{L(k-1)} - I_{L(k+1)} \\ sL_k \cdot I_{Lk} = U_{Lk} = U_{C(k-1)} - U_{C(k+1)} \end{cases} \tag{3.2}$$

The equations are slightly different for the first and the last integrator. Equation (3.3) represents the state-space description of the ladder of figure 3.3:

$$\begin{cases} sR_G C_1 \cdot U_{C1} = E_G - U_{C1} - R_G I_{L2} \\ s\frac{L_2}{R_G} \cdot R_G I_{L2} = U_{C1} - U_{C3} \\ sR_G C_3 \cdot U_{C3} = R_G I_{L2} - R_G I_{L4} \\ s\frac{L_4}{R_G} \cdot R_G I_{L4} = U_{C3} - U_{C5} \\ sR_G C_5 \cdot U_{C5} = R_G I_{L4} - \frac{R_G}{R_L} U_{C5} \end{cases} \tag{3.3}$$

$$H(s) = \frac{U_2}{E_G} = \frac{U_{C5}}{E_G}$$

The state space set of equations (3.3) is built by a chain of integrators and the LF feedback loops, as shown in figure 3.4 for the case $R_G = R_L = R$.

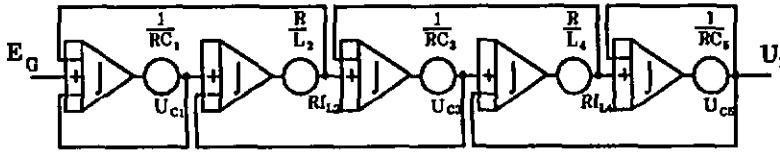


Figure 3.4 Ladder LF filter.

The ODLF realization of the LF structure is similar to the device of figure 3.4: the integrators are replaced by UDCs and the coefficients setting the integrator gain are realized by RMs.

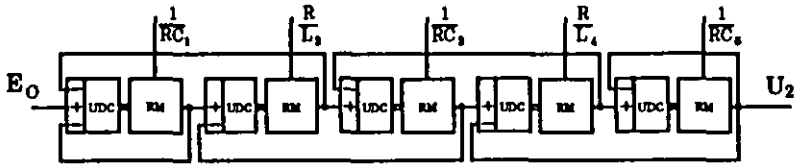


Figure 3.5 Ladder ODLF.

As has already been stated, the greatest power transfer is done when the generator and the load are matched, this is when $R_G = R_L$. Nevertheless, it appears that for an all-pole ladder this requirement cannot always be satisfied. This is due to the fact that an equally terminated all-pole ladder has a maximal transfer function amplitude of 1/2 and the DC amplitude also is 1/2. This condition that the transfer function amplitude is never greater than its DC value is met by Bessel, Butterworth and odd order Chebyshev functions but not by even order Chebyshev functions.

The ladder LF synthesis allows to realize all-pole lowpass filters in a straightforward manner. In this case, all shunt impedances of the passive prototype are capacitances and all series impedances are inductances. The determination of the element values C_k and L_k from the filter poles locus is the most difficult point of the synthesis. For usual filter types, the element values can be obtained by filter design programs or found in tables [Saa79]. This difficulty is the price to pay for a decreased sensitivity of the generated filters.

The generic case of ladder filters owning not only poles but also zeroes is not trivial to simulate using active devices. Various methods have been proposed, however none of them emerged as a definite solution. It can be noted that the analog ladder prototype owning zeroes is no more canonical in terms of number of elements, so the active device simulating the analog ladder also will require more hardware than required by the order of the filter. For the synthesis of inverse Chebyshev or elliptic (also named Cauer) filters, a method based on the simulation of a doubly loaded LC lattice two-port will be presented here.

3.3 Active lattice synthesis

Compared to the ladder, the lattice structure suffers of a greater sensitivity in the stopband. However, the lattice synthesis allows to realize transfer functions owning not only poles but also zeroes with the smallest hardware, as they are canonical in terms of integrators. In this section, first the analog lattice is presented. A realization method is presented for biasing the filter state variables, in order to synthesize ODLFs using unsigned arithmetic. The relation between the lattice branches and allpass

filters is established. From this, the analogy between the active lattice and the Lattice Wave Digital Filter (LWDF) structure is presented. This analogy allows to specify a doubly complementary transfer function for the active lattice.

3.3.1 Active lattice

The synthesis of a canonical active filter structure from an analog passive lattice two-port is performed in a straightforward manner.

Lattice filters, such as the one depicted in figure 3.6, are seldom used as analog devices because they require the two impedances Z_a and Z_b to be replicated in the mirror branches. Because of this, the number of elements is twice as large as the one of a canonical structure and the filters are very sensitive to a mismatch between these mirror impedances.

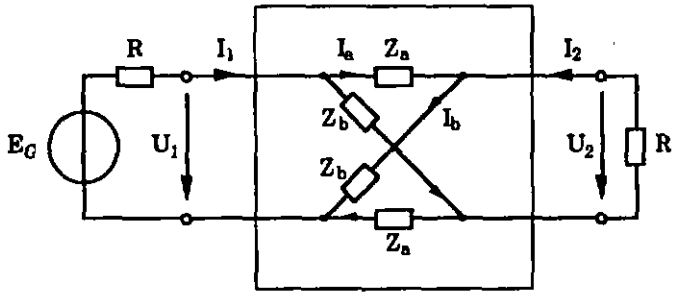


Figure 3.6 Symmetrical lattice.

Active filters can achieve the simulation of an analog passive symmetrical LC lattice terminated in equal resistances while remaining canonic in regard to the number of elements. For doing so, they take advantage of the symmetries inside the filter. Indeed, in the two-port of figure 3.6, the voltages and the currents associated to the mirror impedances are identical.

Furthermore, the voltages and the currents associated to the impedances Z_a and Z_b are independent of each other. This fact allows to realize an equivalent network made out two subnets. Indeed, as the voltages on the impedances Z_a and Z_b are independent of each other, one can replace Z_b in figure 3.6 by an open circuit to obtain the subnet corresponding to Z_a. The same mechanism can be used for the subnet corresponding to Z_b. The lattice filter output is the difference between the voltages on Z_a and Z_b, as can be seen in figure 3.6. The network equivalent to the lattice is represented in figure 3.7.

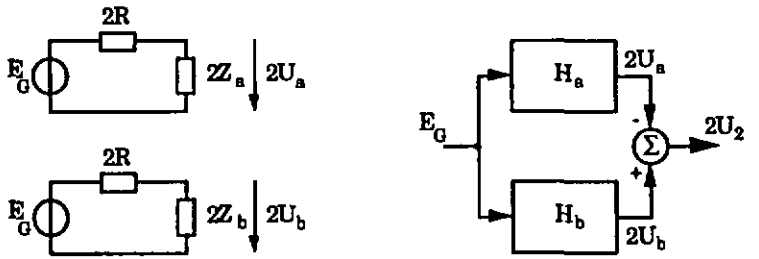


Figure 3.7 Decomposition of the symmetrical lattice into two branches.

In this way, the lattice filter simulation can be realized by two identical generators each loaded by one of the impedances and a comparator providing the filter output, as represented in figure 3.7. Each resistive generator and its load impedance is designated as a branch.

The transfer functions associated to the branches are written as

$$\begin{cases} H_a(s) = \dfrac{2U_a}{E_G} = \dfrac{Z_a}{Z_a+R} \\ H_b(s) = \dfrac{2U_b}{E_G} = \dfrac{Z_b}{Z_b+R} \end{cases} \tag{3.4}$$

and the overall filter transfer function is obtained from difference of the partial transfer functions

$$H(s) = \frac{2U_2}{E_G} = H_b(s) - H_a(s) \tag{3.5}$$

The realization of the filter as the parallel connection of two blocks leads to the apparition of zeroes in the transfer function. The factor 2 in equation (3.5) has been inserted for reasons of normalization: the maximal voltage on the load of an LC network terminated in equal resistances is one half of the input.

The Cauer1 realization of the impedances Z_a and Z_b provides a circuit in the form of a ladder [Tem77]. A corresponding branch is shown in figure 3.8. It corresponds to a singly loaded ladder. The active filter realization of the branch will correspond to the LF chain of integrators of figure 3.4 where the feedback link, corresponding to the load resistor, on the last integrator of the chain is removed.

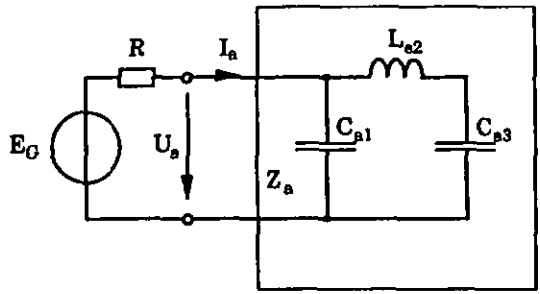


Figure 3.8 Realization of one branch.

If the Cauer1 realization starts with a capacitance, the output of the branch (the voltage on the impedance) is equal to the voltage on the first capacitor C_{a1}. This is actually a filter state variable: it corresponds to the output of the first integrator of the chain.

3.3.2 Active filter with positive variables

In order to work with unsigned arithmetic inside the ODLF, the state variables of the active filter must remain positive.

If the branch of figure 3.8 is biased with a positive constant input U_0, the voltages on the capacitances will have the same bias but the currents in the inductances will all be zero. In this case, a variation of the input could cause these currents to become negative. To avoid this, a bias current is forced in the inductances by a current source I_0 as shown in figure 3.9. For an even order branch, a voltage bias RI_0 is applied on the capacitances.

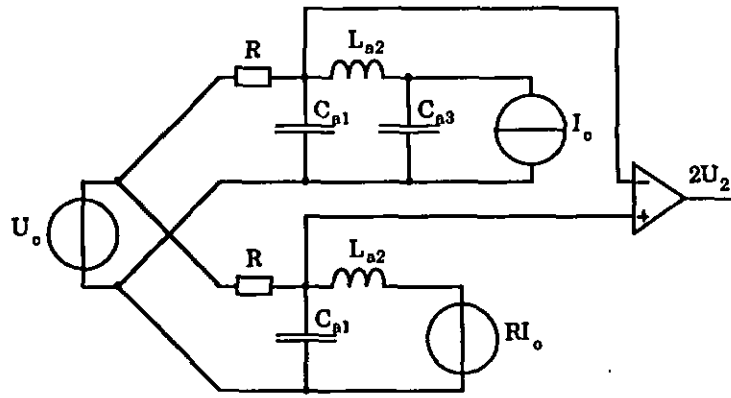


Figure 3.9 Lattice filter with positive state variables.

In the active filter, the bias is simulated by a constant value at the negative input of the last integrator of the chain.

The active filter simulating the enelog equally terminated passive LC symmetrical lattice is realized by two branches and subtracter comparing the outputs of the branches. Each branch consists of a LF chain simulating a singly terminated ladder.

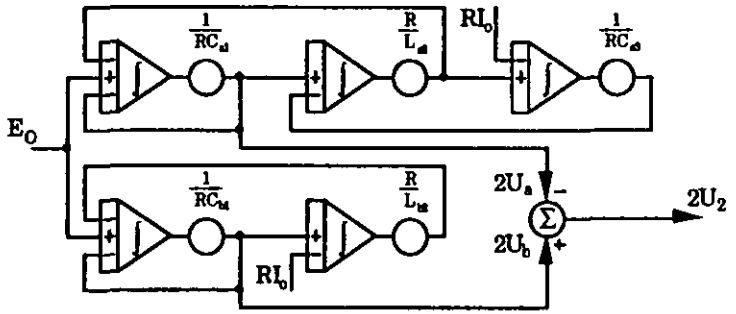


Figure 3.10 Active lattice filter.

The ODLF corresponding to the active lattice of figure 3.12 is represented in figure 3.11.

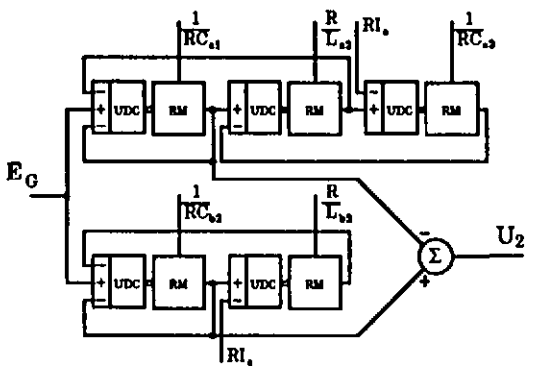


Figure 3.11 Lattice ODLF.

The lattice ODLF of figure 3.11 is of same order as the ladder ODLF of figure 3.5, but this structure can implement transfer functions with smaller transition bands by a proper placement of the zeroes it possesses. In the passband, the sensitivity of the active lattice is comparable to the one of the active ladder. In the stopband however, the sensitivity of the active lattice is much greater.

3.3.3 Allpass active filter

One branch of the activa lattica filter can serve to realize an allpass function. The interest of this is twofold: this propriety can serve to realize phase equalizers and the relation between a lattice branch and an allpass

filter allows to determine a relation between active lattice filters and LWDFs.

Allpass filters are widely used in signal processing. They allow a modular approach to filter synthesis. A filter can be realized by the cascade of two devices: the first one satisfies only the amplitude requirements of the filter and the second one, an allpass filter, matches the phase response of the system to the filter requirements.

An allpass filter can be realized as a lattice filter where $Z_a \cdot Z_b = R^2$ [Tem77]. However, the structure proposed here uses a single branch of the active lattice filter. This corresponds to a singly loaded ladder filter and is a subclass of the orthonormal ladder filters presented by [Joh89]. It presents the advantage to remain an allpass filter independently of coefficient changes.

The zeroes of an allpass filter are the opposites of its poles. Because of this, the coefficients of the numerator N(s) of the transfer function H(s) of an allpass filter are equal to the corresponding coefficients of the denominator D(s) except for sign reversals: the coefficients of the odd powers of s of the numerator are the opposites of the coefficients of the odd powers of s of the denominator. This propriety can be written as:

$$H(s) = \frac{N(s)}{D(s)} = \frac{D(-s)}{D(s)} = \frac{D_e(s) - D_o(s)}{D_e(s) + D_o(s)} \qquad (3.6)$$

where $D_e(s)$ and $D_o(s)$ are respectively the even and the odd part of D(s). Equation (3.6) can be rewritten as:

$$H(s) = \frac{\frac{D_e(s)}{D_o(s)} - 1}{\frac{D_e(s)}{D_o(s)} + 1} \qquad (3.7)$$

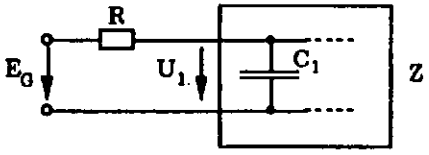
The quotient $D_e(s) / D_o(s)$ corresponds to an LC impedance: the numerator is even, the denominator is odd and the degrees of the numerator and of the denominator differ by one. This impedance Z, normalized to a resistance R, can be written as:

$$Z = R \cdot \frac{D_e(s)}{D_o(s)} \qquad (3.8)$$

so equation (3.7) becomes

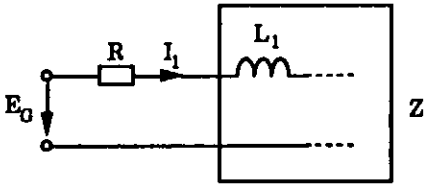
$$H(s) = \frac{Z - R}{Z + R} \qquad (3.9)$$

This transfer function is obtained by taking the difference between the filter input E_G and the double of the first state variable, as can be seen from figure 3.12.



$$\frac{E_G - 2U_1}{E_G} = \frac{R - Z}{R + Z} \qquad (3.10)$$

a) branch starting with a capacitance



$$\frac{E_G - 2RI_1}{E_G} = \frac{Z - R}{Z + R} \qquad (3.11)$$

b) branch starting with an inductance

Figure 3.12 Allpass filter.

A change in one of the filter coefficients corresponds to a change in the element values thus generating a different impedance than the expected one. Still the system transfer function, given by equation (3.9) remains an allpass function.

3.3.4 Relationship to LWDFs

LWDFs show a structure similar to the one of active lattice filters. The LWDFs output is made out of the difference of two allpass branches. There is a one to one correspondence between the transfer functions of the branches of the LWDFs and of the active lattices filters.

LWDFs also simulate a passive analog lattice filter. In a similar way as the one described above, the lattice is split into two branches and the output is made up by the comparison of the two branches [Fet86]. For the LWDFs, the transfer function of the individual branches is given by

$$\begin{cases} S_a(\psi) = \dfrac{g_a(-\psi)}{g_a(\psi)} = \dfrac{Z_a - R}{Z_a + R} \\ S_b(\psi) = \dfrac{g_b(-\psi)}{g_b(\psi)} = \dfrac{Z_b - R}{Z_b + R} \end{cases} \qquad (3.12)$$

where ψ is given by $\psi = (z-1) / (z+1)$ and is thus closely related to the variable z. The transfer functions of the individual branches are of allpass type and correspond to the allpass filter described by equation (3.9). The

overall filter transfer function is made out of the difference of the transfer functions of the branches:

$$H(\psi) = S_{21}(\psi) = \frac{S_b(\psi) - S_a(\psi)}{2} \qquad (3.13)$$

The comparison of equations (3.4), (3.5) and (3.12), (3.13) puts forth the similarity of the two approaches. The relation between the two approaches is given by the relation between the transfer functions of the two types of branches:

$$H_a = \frac{Z_a}{Z_a+R} = \frac{S_a + 1}{2} \qquad (3.14)$$

The transfer functions of the two types of branches differ by an additional constant term and a factor of two. The comparison of the two branches cancels out the constant and the factor of two is found in the LWDF transfer function, equation (3.13), so that fortunately both overall transfer functions are identical.

The design of an active lattice can take advantage of this relationship. To determine the transfer functions of the branches, the active lattice designer can use algorithms developed for LWDFs such as [Gaz85] and obtain the desired active lattice functions using equation (3.14). Additionally, a complementary transfer function is obtained for active lattice filters in the same way as for LWDFs.

3.3.5 Doubly complementary transfer functions

The active lattice structure allows to directly obtain a complamentary filter output. This feature allows the design of a highpass filter from a lowpass reference prototype as well as the realization of bandsplit filters.

Two filters are commonly said to be a complementary pair if the passband of one matches the stopband of the other and vice versa [Mit85]. Complementary filters are used to separate the information of a signal from different frequency bands.

The two filters are said to be allpass complementary if their transfer functions $H(j\omega)$ and $\overline{H}(j\omega)$ satisfy the following condition:

$$| H(j\omega) + \overline{H}(j\omega) | = 1 \qquad (3.15)$$

This condition indicates that it is possible to reconstruct the original signal from the complementary filter outputs except for a phase distortion.

Square magnitude complementary filters are defined by

$$| H(j\omega) |^2 + | \overline{H}(j\omega) |^2 = 1 \qquad (3.16)$$

which means that the energy of the original signal is separated into the two complementary outputs.

The two filters are said to be doubly complementary if both conditions are met. LWDFs offer an efficient and natural implementation of doubly complementary filters as a parallel connection of two allpass branches.

In a similar way, the output of the active lattice filter is given by the comparison of two branches, as specified by equation (3.5):

$$H(z) = H_b(z) - H_a(z) \tag{3.17}$$

and the doubly complementary filter transfer function is given by

$$\overline{H}(z) = 1 - H_a(z) - H_b(z) \tag{3.18}$$

As it is the case for LWDFs, the generation of the doubly complementary output requires a very small amount of overhead. The complementary output can be used for the design of a highpass filter. Indeed, a lowpass filter can be designed instead and the complementary output realizes the highpass function.

The synthesis procedure of a lattice ODLF is similar to the one used for a LWDF. The transfer function poles are distributed among the lattice branches. The reactance functions Z_a and Z_b are derived from the values of the poles of their branch. The filter output is then realized by the comparison of the two branches. So far, no care has been taken about the transfer function zeroes. As a matter of fact, the lattice design does not provide any straightforward possibility to decide of the locations of the zeroes. The zeroes of the overall transfer function arise from the parallel connection of the two branches and are a function of the values of the poles. As such, lattice filters are limited to a restricted set of filter types. Fortunately, most amplitude approximation functions, such as elliptic, Chebyshev and Butterworth, are realizable with this synthesis procedure.

One further restriction is that lowpass and highpass transfer functions can only be realized with odd order filters whereas bandpass and bandstop (also called notch) filters are limited to even order devices. This can be understood by the examination of the phase response of an even and of an odd allpass LWDF branch.

The higher sensitivity of the active lattice in the stopband prone for the use of high order devices in the case of sharp cutoff or tight passband specifications. Analysis allows to determine the transfer function deviation in the stopband due to signal quantization and to choose the oversampling ratio and the number of bits required for the counters.

4 Analysis

The analysis of the transfer function deviation and of the scaling of the signals inside ODLFs is realized with the help of the state-space description of the filter.

The state-space matrices of the ladder and the lattice filters are provided for the case of a fifth order filter. Due to the regularity of the structures, the general form of the state-space matrices are derived by analogy.

The quantization of the filter coefficients leads to a deterministic distortion of the transfer function. The state-space description enables to estimate the transfer function of the filter with quantized coefficients. Furthermore, the state-space matrices can be used to examine the sensitivity of the transfer function to coefficient values.

The quantization of the filter variables is analyzed for large scale and small scale effects. Large scale quantization errors arise from overflow and are avoided by proper signal scaling. Small scale quantization errors result in a distortion of the transfer function.

4.1 State-space description

The state-space description has been mainly developed for control theory. This description does not only comprise the information relative to the transfer function of a device, it also scopes the internal workings of the system. The state-space description of a system describes three types of signals:

- inputs, which are ordered in a column vector X(s),
- state variables, a set of internal signals, ordered in a vector W(s),
- outputs, which are ordered in a column vector Y(s),

The description of a time-continuous system is given in the Laplace domain by equations (4.1) and presented graphically in figure 4.1.

$$\begin{aligned} s \cdot W(s) &= A \cdot W(s) + B \cdot X(s) \\ Y(s) &= C \cdot W(s) + D \cdot X(s) \end{aligned} \qquad (4.1)$$

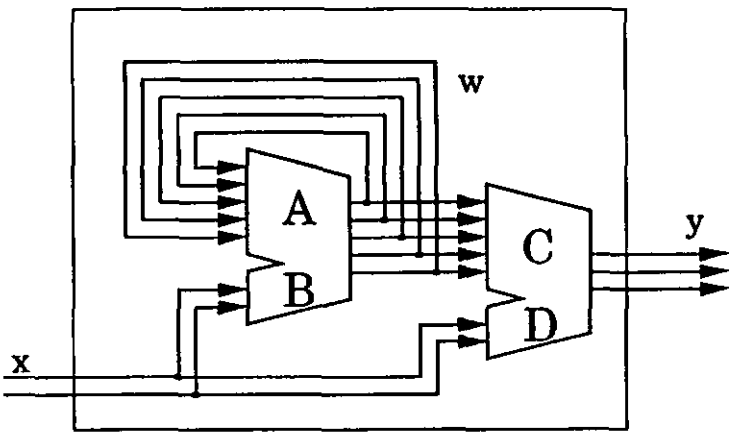


Figure 4.1 State-space description.

The poles of the system are the eigenvalues of the matrix A. The zeroes depend on all state matrices. In the case of a Single Input Single Output (SISO) filter of order n, the sizes of the vectors are the following:

- X(s) contains only one element and will be written as X(s),
- W(s) contains n elements,
- Y(s) contains only one element and will be written as Y(s).

Consequently:

- A is a n by n matrix,
- B is a n element column vector,
- C is a n element row vector,
- D is a scalar.

If not explicitly specified, the filter state-space analysis will be presented for SISO systems. The derived results can be extended to the Multiple Input Multiple Output (MIMO) case, only the formulation is much more cumbersome.

At present, the state-space matrices A, B, C, D of the ODLFs simulating the analog passive ladder and lattice filters will be determined.

4.2 Ladder and lattice state-space matrices

The state-space matrices are derived here for the more general case of Integrator based active filters. The ODLFs are considered as an active filter subclass where the integrators are realized by UDCs and the integrator gains are set by RMs.

4.2.1 Ladder active filter

The state-space description of an active filter simulating an analog passive ladder filter terminated in equal resistances R can be obtained from equations (3.3) in the particular case of a 5^{th} order device. They can be rewritten in a matrix form as:

$$s\cdot\begin{pmatrix} U_{C1} \\ RI_{L2} \\ U_{C3} \\ RI_{L4} \\ U_{C5} \end{pmatrix} = \begin{pmatrix} -c_1 & -c_1 & 0 & 0 & 0 \\ c_2 & 0 & -c_2 & 0 & 0 \\ 0 & c_3 & 0 & -c_3 & 0 \\ 0 & 0 & c_4 & 0 & -c_4 \\ 0 & 0 & 0 & c_5 & -c_5 \end{pmatrix}\begin{pmatrix} U_{C1} \\ RI_{L2} \\ U_{C3} \\ RI_{L4} \\ U_{C5} \end{pmatrix} + \begin{pmatrix} c_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} E_G \qquad (4.2)$$

$$U_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} U_{C1} \\ RI_{L2} \\ U_{C3} \\ RI_{L4} \\ U_{C5} \end{pmatrix} + 0\cdot E_G$$

where c_k is either $1/RC_k$ or R/L_k.

The state-space matrices are easily obtained by the comparison of equations (4.1) and (4.2). The matrices are given for the particular case of a 5^{th} order device but their general form can be derived by analogy.

The state variables have been chosen as the currents in the inductances, multiplied by the value of the termination resistances, RI_{Lk}, and the voltages on the capacitances, U_{Ck}. For the ODLF, this defines the state variables to be the RM outputs. Alternately, the state variables could be defined as the relative values of the currents and voltages to the corresponding element values, I_{Lk}/L_k and U_{Ck}/C_k. This would lead to different coefficients in the state-space matrices and would cause the UDC outputs to be the ODLF state variables. Important to notice is that the state variables are not the only internal filter signals but there exist also

auxiliary variables used to build the state variables. These signals also have to be taken into account for the signal quantization analysis.

4.2.2 Lattice active filter

The state-space description of the active lattice filter of figure 3.11 is given by equations (4.3). Two outputs are provided: the lattice filter direct output and the doubly complementary output.

$$s\cdot\begin{pmatrix} U_{Ca1} \\ RI_{La2} \\ U_{Ca3} \\ U_{Cb1} \\ RI_{Lb2} \end{pmatrix} = \begin{pmatrix} -c_{a1} & -c_{a1} & 0 & 0 & 0 \\ c_{a2} & 0 & -c_{a2} & 0 & 0 \\ 0 & c_{a3} & 0 & 0 & 0 \\ 0 & 0 & 0 & -c_{b1} & -c_{b1} \\ 0 & 0 & 0 & c_{b2} & 0 \end{pmatrix} \begin{pmatrix} U_{Ca1} \\ RI_{La2} \\ U_{Ca3} \\ U_{Cb1} \\ RI_{Lb2} \end{pmatrix} + \begin{pmatrix} c_{a1} \\ 0 \\ 0 \\ c_{b1} \\ 0 \end{pmatrix} E_G$$

$$\begin{pmatrix} 2U_2 \\ 2\overline{U_2} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} U_{Ca1} \\ RI_{La2} \\ U_{Ca3} \\ U_{Cb1} \\ RI_{Lb2} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} E_G \qquad (4.3)$$

where c_k is either $1/RC_k$ or R/L_k.

Here again, the state-space matrices are easily obtained by the comparison of equations (4.1) and (4.3). The matrices are given for the particular case of a 5^{th} order device but their general form can also be derived by analogy.

4.3 Quantization of the coefficients

Analysis of coefficient quantization allows the filter designer to find an optimal coefficient set in order to minimize the filter hardware.

4.3.1 Alteration of the transfer function due to coefficient quantization

The quantization of the filter coefficients leads to a deterministe modification of the transfer function. The examination of the disturbance due to coefficient quantization requires an algorithm to evaluate the transfer function from the filter coefficients. The algorithm used here is to build up the state-space matrices with the coefficient set and to use the state-space theory to evaluate the transfer function.

The coefficients are only quantized once: during the design process. Hence, the effect of coefficient quantization is to disturb the transfer function from its ideal form, usually in an invariable manner [Jac87]. A case where coefficient quantization leads to a signal dependent alteration of the filter response is distributed arithmetic [Kam77]. The transfer function of the filter with quantized coefficients can be obtained from the state-space matrices. This transfer function is compared to the filter specifications.

The transfer function of the state-space system is shown [Sch69] to be:

$$H(s) = C\,(sI-A)^{-1}\,B + D \qquad (4.4)$$

where I is the identity matrix.

Coefficient quantization is an optimization task and is thus done by recursive methods. The coefficients are set to some quantized values, as for example the rounded value of their binary representation. From these, the state-space matrices are built and the transfer function is evaluated. If the filter specifications are met, the set of coefficients is regarded as one possible solution. The search process is repeated in order to find an optimal solution in terms of hardware.

The ODLF coefficients are pulse rates. A practical realization method is to generate them from a common reference frequency. Because of this, optimization is concerned of finding a set of coefficients which are most easily derived from the reference frequency. The search will be oriented to obtain a set of common dividers of a same value or at least of fractional parts of a same value. Also of great interest is to obtain identical coefficients. The most direct approach is to generate the ODLF coefficients using RMs clocked by the same reference frequency. In this case, the coefficients are rounded to the nearest value of their binary representation and fed to the bit-parallel RM inputs. However, algorithms like simulated annealing or even, in the case of low order filters, trial and error can lead to far more economical solutions. Interesting to point out is that the optimization is meant to reduce the hardware used for the generation of the coefficients and not the hardware of the filter itself, as it is generally the case for digital filters.

4.3.2 Sensitivity of the transfer function to coefficient values

The estimation of the sensitivity of the transfer function is a valuable tool for the comparison of filter structures. The state-space description also provides algorithms for the evaluation of the sensitivity.

The most common measure of the sensitivity of the transfer function H(s) to the value of the coefficient c, called classical sensitivity, is defined by:

$$S_c^{H(s)} = \frac{\partial H(s)}{\partial c}\,\frac{c}{H(s)} = \frac{\partial(\ln H(s))}{\partial(\ln c)} \qquad (4.5)$$

This corresponds to a relative tolerance measure: it indicates how a relative change $\partial c/c$ in the coefficient leads to a relative change $\partial H/H$ in the transfer function.

For SISO systems, the partial derivative of equation (4.5) can be obtained from the state-space matrices as presented in [Sne86]. For this, two dual sets of intermediate transfer functions have to be defined. The first set contains the transfer functions from the filter input to the integrator outputs and are referred as the signal gains.

$$F(s) = \frac{W(s)}{X(s)} = (sI-A)^{-1}\,B \qquad (4.6)$$

The second set contains the transfer functions from the integrator inputs to the filter output and are referred as the noise gains.

$$G(s) = C\,(sI-A)^{-1} \tag{4.7}$$

From these function, the partial derivative of the transfer function with respect to the matrices coefficients is given by

$$\begin{cases} \dfrac{\partial H(s)}{\partial A_{ij}} = G_i(s)\cdot F_j(s) \\ \dfrac{\partial H(s)}{\partial B_i} = G_i(s) \\ \dfrac{\partial H(s)}{\partial C_i} = F_i(s) \\ \dfrac{\partial H(s)}{\partial D} = I \end{cases} \tag{4.8}$$

The sensitivity of a given structure to variations of its coefficients can be estimated by the standard deviation in the transfer function for standard deviations of 1 percent of the nominal component values [Joh89].

$$\sigma\,|H(j\omega)| = 0.01\cdot\left[\sum_{c = A_{ij}\,B_i\,C_i\,D}\left(\frac{\partial\,|H(j\omega)|}{\partial c}\,c\right)^2\right]^{1/2} \tag{4.9}$$

In the passband, the transfer function deviation is measured as the relative value of the expected transfer function to the ideal response. It is given in dB [Joh89] by

$$D(\omega) = 20\cdot\log_{10}\left(1+\frac{\sigma\,|H(j\omega)|}{|H(j\omega)|}\right) \tag{4.10}$$

In the stopband, an expressive deviation measure is the expected transfer function, this is the sum of the ideal transfer function and the deviation. It is given in dB [Joh89] by

$$H_\sigma(\omega) = 20\cdot\log_{10}\left(|H(j\omega)| + \sigma\,|H(j\omega)|\right) \tag{4.11}$$

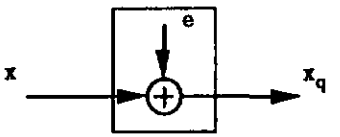
The sensitivity measures are of greater importance in the analog than in the digital domain because of the variations of the element values with temperature or because of aging. For digital filters, these measures provide an indication about the sensitivity of the structures to the small scale quantization of the variables, as this quantization usually takes place in the multipliers.

4.4 Quantization of the internal variables

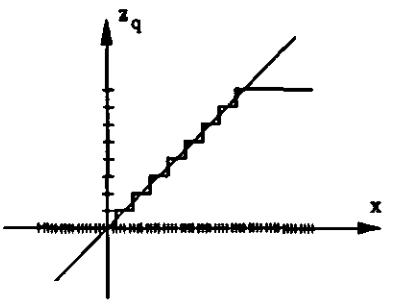
The analysis of the quantization of the signals inside a digital filter is meaningful for the designer to ensure stability and the required SNR of the device. Stability is ensured by a proper scaling of the signals to prevent from large scale quantization effects whilst the SNR is determined by the small scale quantization effects.

As opposed to coefficient quantization, the quantization of the signals inside a filter is no more deterministic. Strictly speaking, the resulting system is no more linear [Jac87].

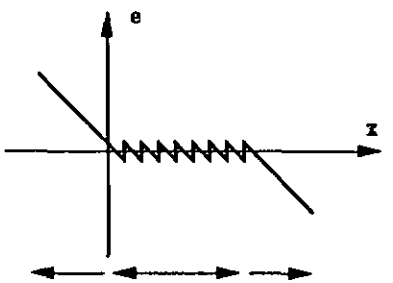
The effect of signal quantization is modeled as linear perturbations by additive quantization noise: a quantization error e is added to the signal x producing the quantized signal x_q. An example of quantization law with saturation is shown in figure 4.2 b). As can be seen in figure 4.2 c), two main regions of operation can be defined: the quasi-linear operation region and the overflow region [Cla84].



a) quantization model



b) quantization law



c) quantization error

Figure 4.2 Modeling of signal quantization.

The topic of the quantization of the filter internal variables is thus separated into two fields: quantization range which deals with overflow, and quantization step which analyses the SNR in the quasi-linear operation region.

Inside the ODLFs, the two specific situations occur at different locations: overflow might arise at the output of the UDCs whilst quantization is happens inside the RMs. The quantization model looks somewhat like the one described in figure 4.2, but with specific attributes:

- The overflow characteristic is due to the saturation of the UDCs and is identical to the one presented in figure 4.2. Alike most adders or accumulators of usual Infinite Impulse Reaponse (IIR) digital filters, the UDCs are not a source of small scale quantization errors.
- Since the RM coefficient values are smaller than unity, this kind of operator can never be the source of overflow. Yet the small scale quantization errors arise inside the RMs, as it is the case for multipliers in digital filters. In the quasi-linear operation region, the quantization law cannot be described in a deterministic manner as in figure 4.2 b). Indeed, because of the dithering, the quantization error of a RM can be different for input samples of same amplitude at different times. The quantization error of the RMs has been described in chapter 2, paragraph.2.2.3.

4.4.1 Quantization range

The analysis of large scale quantization effects requires an estimation of the range the signals inside the filter will use. Three estimators are presented hereafter.

Overflow occurs when the signal grows larger than the maximum quantization value. Overflow of internal variables produces errors of same order of magnitude as the Most Significant Bit (MSB) of the digit holding the variable and causes large damage to the filter output. Its effects can be altered by influencing the overflow characteristic. The most common overflow characteristic is saturation arithmetic as presented in figure 4.2.

In order to prevent overflow, the designer needs to estimate the dynamic range of the internal filter variables. The choice of a measure for the magnitude of the signal

$$W_k(z) = H_k(z) \cdot X(z) \qquad (4.12)$$

where $H_k(z)$ is the transfer function from the input to the internal variable w_k, depends strongly on the filter application and in particular on what is known about the input signal x(n) [Sne86]. Based upon the a-priori spectral characteristics of the input, three figures can be used to determine the amplitude range of the internal variable w_k:

- $\|W_k\|_\infty$: the peaking in the frequency response from X to W_k,
- $\|W_k\|_2$: the Root Mean Square (RMS) amplitude of W_k,
- $\|W_k\|_1$: the maximal bound for the amplitude of W_k.

These estimations correspond respectively to the L_∞ norm, the L_2 norm and the L_1 norm commonly used in digital signal processing. Due to saturation and slew-rate limiting of the active integrators, the L_∞ and the L_2 norm have also been adopted in analog signal processing [Sne86].

If the filter input is known to consist mainly of a sinusoid of peak amplitude x_{max}, the peaking $\|W_k\|_\infty$ in the frequency response from the input to the internal variable W_k provides a natural estimation of the signal range.

$$\|W_k\|_\infty = X_{max} \cdot \max(|H_k(f)|) \qquad (4.13)$$

This figure is easy to evaluate using the state-space approach. The transfer functions from the input to all state variables is obtained from equation (4.4) by replacing C with an identity matrix and D with a zero vector.

Assuming that the input power spectrum is white with density X_{max}^2, the RMS amplitude $\|W_k\|_2$ corresponds to the square root of the power transmitted to the internal variable w_k. This value can be estimated in the frequency or in the time domain due to the identity of Parseval.

$$\|W_k\|_2 = X_{max}\left[\int_{-\infty}^{\infty} |H_k(f)|^2\, df\right]^{1/2} = X_{max}\left[\int_{-\infty}^{\infty} h_k^2(t)\, dt\right]^{1/2} \qquad (4.14)$$

where $h_k(t)$ is the impulse response of the transfer function $H_k(s)$ from the input to the internal variable w_k.

The maximal bound $\|W_k\|_1$ for the amplitude of the internal variable w_k is a sufficient and necessary condition to preclude overflow [Jac87]. It corresponds to the maximal possible value of the convolution between the step response and the filter input.

$$\|W_k\|_1 = \int_{-\infty}^{\infty} X_{max} \cdot |h_k(t)|\, dt \qquad (4.15)$$

However, this estimation is very pessimistic about the performances of the system. The choice of this figure to set the number of bits of the variables or the oversampling ratio is overly conservative and leads to the sacrifice of available dynamic range.

In the ODLFs, the locations where overflow can occur are the UDCs. These variables are proportional to the state-space variables by a factor of $1/c_k$. The transfer function $H_k(s)$ is obtained from equation (4.4) where the matrix C is a row vector where only the k^{th} element is non-zero and equal to $1/c_k$, and where the matrix D is the scalar 0.

An efficient way to determine the L_2 norm for singly terminated ladder state variables, which are represented in the ODLFs by the RM outputs, is provided in [Joh89].

$$\|W_k\|_2 = \sqrt{\frac{\pi}{X_k}} = \sqrt{\pi \cdot c_k} \qquad (4.16)$$

where X_k is the reactance (inductance L_k or capacitance C_k) of the ladder element. This allows a straightforward determination of the L_2 norm from the ODLF coefficients. This evaluation can evidently be used for the ladder branches of the lattice filter realization. The a-priori knowledge of the L_2 norm could be used to scale the ODLF coefficients to have a maximum norm of 1, which means a maximal coefficient of $1/\pi$. However, this leads to the limitation of the RM outputs to one third of their possible range and anyway this scaling has no effect a on the L_∞ norm which depends on the relative value of the elements. Scaling the coefficients to have a maximal value of 1 allows a maximal range and gives the maximal L_2 norm of $\sqrt{\pi} \cong 1.77$.

The estimation of the signal ranges of each internal signal allows the designer to determine the amplitude they require for a proper operation of the filter. Conversely, the internal filter signals can be scaled to make a maximal use of their dynamic range.

4.4.2 Signal scaling

In order to maximally exploit the range they are attributed, signals have to be scaled. In the case of ODLFs, the scaling of the UDC outputs is done by adding LSBs to the counter. This provides an automatic scaling of the RM outputs as a proper scaling of the UDCs results in constraining the RM coefficients in the range [1/2, 1].

The expected amplitude of the internal signals is provided by the L norms together with the amplitude of the filter input. So from the L norms, the relative ranges of input signal and the internal signals are defined. The aspect of the scaling will be examined first for the system state variables and then for the auxiliary signals used to build the state variables.

Scaling the state variables of a system is done by multiplying the state vector by a diagonal transformation matrix T. The state-space description becomes

$$\begin{aligned} s\cdot TW(s) &= TAT^{-1}\cdot TW(s) + TB\cdot X(s) \\ Y(s) &= CT^{-1}\cdot TW(s) + D\cdot X(s) \end{aligned} \qquad (4.17)$$

So if the state variable w_k is scaled up by a factor of two, the k^{th} element of W(s) is multiplied by two, the k^{th} row of A and B are multiplied by two and the k^{th} column of A and C are divided by two.

For ODLFs, this means that if one RM coefficient is smaller than one half, it can be doubled. In this case, T is an identity matrix with one diagonal element equal to 2. With the transformation, a row of A is multiplied by two and the corresponding column of A has is divided by two, which means that the RM pulse rate has to be divided by two before each UDCs it is fed into. Also the relations between the UDC-RM pair to the input and to the output have to be updated. Nevertheless, this scaling of the state variables does not bring any SNR enhancement: the RM outputs can use a

larger dynamic range but the UDC inputs remain the same as in the original ODLF.

On the other hand, the scaling of the UDC outputs can improve the ODLF performances. The scaling of an UDC output is achieved by multiplying its inputs by a given value and by dividing its output by the same factor. Scaling down an UDC output by a factor of two can be realized by appending one Least Significant Bit (LSB) to the UDC and by multiplying the associated RM coefficient by two. As the maximal value for an ODLF coefficient is 1, the number of bits of the UDCs can be enlarged until the corresponding RM coefficient is greater than 1/2.

This scaling by a power of two necessitates little hardware overhead and causes the L norms of the UDC outputs to be of same order of magnitude as the ones of the RMs. As the amplitude of all ODLF internal signals have been normalized to the interval [0, 1], the L norms are used to determine the allowed range for the input in order to preclude overflow.

4.4.3 Quantization step

The analysis of the quantization of the filter variables is done to estimate the deviation of the transfer function. The state space description allows to incorporate the quantization noise sources in the system description and to compare the power of the input to the total power of the noise sources at the filter output.

The quantization step determines the power of the quantization error. Quantization is modeled as a stationary random noise source [Cla84]. This noise is propagated through the filter as well as the signal. Under the assumption that all noise sources are uncorrelated with each other or with the internal filter variables, the noise can be considered as a probabilistic process. The noise power and the noise transfer function to the output are used to to estimate the global filter SNR.

In the state-space description, quantization is formulated by the adjunction of new entries in the input vector X at each place in the filter where quantization appears. The system is then considered as linear again. The transfer functions of all noise sources to the filter output can be determined according to equation (4.4) and the effects of all these sources have to be superposed in order to estimate the total noise power spectrum at the filter output and, from this, the filter SNR.

In the ODLFs, quantization is done in the RMs, this is at the location of the state variables. The quantization mode has been presented in chapter 2 where the amplitude of the noise signals is given by equation (2.11) or in dB by equation (2.12). The state-space equation set becomes:

$$\begin{aligned} s\cdot W(s) &= A\cdot W(s) + [B\ A]\begin{bmatrix} X(s) \\ E(s) \end{bmatrix} \\ Y(s) &= C\cdot W(s) + [D\ C]\begin{bmatrix} X(s) \\ E(s) \end{bmatrix} \end{aligned} \qquad (4.18)$$

So the transfer functions from the quantization noise sources to the output are given by:

$$\frac{Y(s)}{E(s)} = C\,(sI-A)^{-1}\,A + C = G(s)\cdot A + C \qquad (4.19)$$

The superposition of the effects of all noise sources is realized by multiplying the transfer functions by the amplitude of the noise sources (which are the inverses of the overesampling ratio) and by summing the absolute values of these functions. The minimal amplitude of the transfer function in the passband is estimated by subtracting the deviation to the ideal transfer function. Conversely, the maximal amplitude of the transfer function in the stopband is estimated by adding the deviation to the ideal transfer function.

The statistical analysis of quantization noise was based on the assumption of uncorrelated noise sources. For most data signals, with sufficient amplitude and spectral content, this assumption is valid and experimental noise spectra closely match the predictions [Jac87]. However, the error signals can happen to be correlated with each other and with the input signal. This affects the validity of the white noise model and results in error signals of larger energy than predicted in some frequency bands. It can even give rise to autonomous parasitic oscillations also called small scale limit cycles inside the filter [Cla84]. The analysis of the existence of limit cycles and eventually of the bounds of their values is not straightforward. There do not exist fully reliable algorithms for the enumeration of the limit cycles in a given filter and the worst case bounds are quite pessimistic about the filter performances.

In review, the state-space description shows to be a worthy approach for digital filter analysis. The analysis passes through the following steps:

- First, the filter coefficients are quantized in order to reduce the filter hardware. The quantized values are inserted in the state-space matrices and the transfer function is checked to satisfy the filter specifications. This is repeated until an optimal coefficient set is found.
- Then, the dynamic range of the internal filter variables is estimated. This is used to determine the maximal amplitude of the filter input or, reciprocally, the extra range of the internal variables of the filter compared to the input. The internal signals can eventually be scaled for optimal filter performance.
- Last, the combined effects of quantization noise are evaluated in order to determine the number of bits or the oversampling ratio required for the filter internal variables to achieve a specified SNR.

The proposed algorithms are general enough to enable the analysis of any other integrator based filter structures. Indeed, the analysis method is the same for ladder and for lattice active filters, only the state-space matrices differ. In this point of view, the cascade of second order sections or

the direct tranafer function realization could ba analyzad using the aame algorithma.

As already atated, tha analyaia ia based on reaaonabla assumptions, but filter aimulation atill ia a necsssary tool for tha varification of tha proper functioning of ths devics. Simulation, too, meraly rsquirea tha atate-apaca deacription of the filter.

Only when a deaign haa aucceaafully paaaed tha atepa of analysis and aimulation, it is matura for practical electronic realization.

5 Realization

The simplicity of the ODLF operators makes this kind of filter an interesting candidate for an Application Specific Integrated Circuit (ASIC) realization. The present work will focus on the full custom Complementary Metal-Oxyde-Semiconductor (CMOS) integrated circuit realization. The approach starts from the high-level system requirements. In addition, the CMOS realization of the frequency divider by 2 is presented, as this is the basic ODLF building block.

The UDC is realized as a set of identical bit slices and one control slice. The operator is synchronous to the system clock.

As the UDC, the RM is made out of bit slices and one control slice. However, the RM pulse frequency modulated signals are synchronized to the other phase of the system clock.

The bit rate adder is such a simple operator that it merely requires one bit slice.

The VLSI implementation of the slices has been realized using a 1.6 μm gate CMOS technology.

5.1 System considerations

5.1.1 System timing

Two major timing schemes are generally used for VLSI circuits, namely synchronous systems and self-timed systems [Mes80]. Synchronous systems are controlled by a master clock the period of which determines the delay granted to the logical blocks to generate the next output from the present input. In self-timed systems, each element waits for an initiation command (receives a request), executes its task and then signals the completion of it (sends an acknowledge). The synchronous system clocking is best suited for the ODLF realization. Indeed, the implementation of a self-timed system requires a hardware overhead in contradiction to the simplicity of the ODLF Moreover, the coefficient generation and thus the changes of the internal signals are synchronized to a sampling clock.

As a digital filter, the ODLF can be realized as a finite state (Moore) machine. A safe clocking scheme requires at least two phases [Mes80]. The corresponding block diagram together with the clock phases are shown in figure 5.1. The switches are closed when the control signal is high. The two clock phases are referred as clk1 and clk2. They are not overlapping, this means that the switches are never both closed in the same time in order to clearly dissociate the present and the next state.

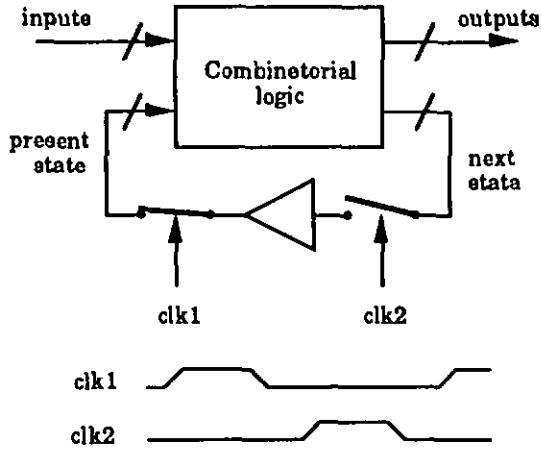


Figure 5.1 Two phase clocked state machine.

An alternate realization of a state machine is to replace the register between the phase switches with buffered combinatorial logic, as presented in figure 5.2. This scheme is of interest for the realization of ODLFs which are made out of two basic operator types: the UDCs and the RMs.

Accordingly, the UDCs are placed in one logic block and the RMs in the other.

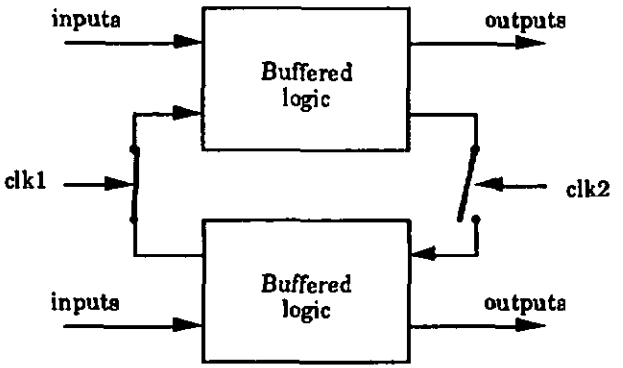


Figure 5.2 Alternate realization of a state machine.

If the UDCs are placed in the top logic block of figure 5.2, they are updated at the rising edge of phase 1. They have a delay of length of the phase pulse to settle to their new value. During this time, their inputs, which are RM outputs, must remain stable. Similarly, the RMs are refreshed during phase 2. In both logic blocks, the operator outputs are registered. These timing specifications are illustrated in figure 5.3.

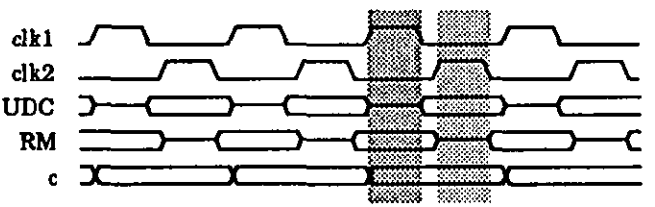


Figure 5.3 ODLF system timing.

As the RM outputs, the filter input must not vary during phase 1 for the proper working of the UDCs and the coefficient pulses must be stable during phase 2 for the RMs. If the filter output is read from an UDC, it is valid during phase 2 and if it is taken from a RM, this has to be done during phase 1.

Now that the timing has been defined for the operators, their practical realization can be examined.

5.1.2 Bit slices

Both the UDC and the RM are basically made out of a counter. The counter is a regular structure and is best implemented as a set of bit slices. One individual bit slice stores the value of one bit of the counter and calculates the next value of this bit from the previous value and from a carry information. The slice also determines a carry output to the next slice. As the ODLF is meant to require as little silicon surface as possible and as the counter outputs are needed as fast as possible, the counters will be implemented with a carry ripple-through logic.

The counter has been shown to necessitate less hardware when realized as an asynchronous rather than as a synchronous device [Bot89]. The asynchronous counter is realized by a chain of dividers by two. In consequence, the realization of a divider by two will be examined first.

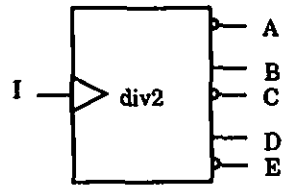
5.1.3 Divider by two

In the ODLFs, the MSBs of the counters can happen to remain without change for long periods, so the storage of their values has to be implemented with static logic. Eventually, the change from one state to the complementary can be of dynamic type. Thus the implementation of the divider by two will be examined using first static logic and then semi-dynamic logic.

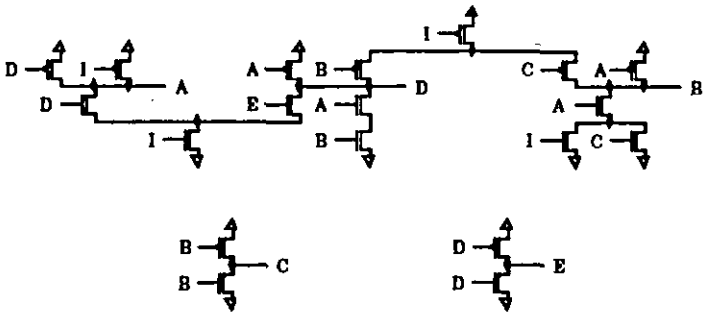
5.1.3.1 Static divider by two

Figure 5.4 presents a CMOS race-free static divider by two. The device is made out of 19 transistors.

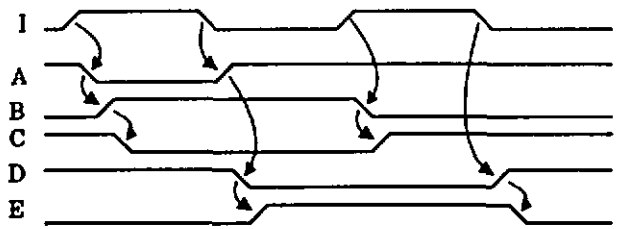
The divider by two provides five outputs. B and C are complementary outputs of a positive edge triggered divider by two. Similarly, D and E are complementary outputs of a negative edge triggered divider by two. A is a auxiliary variable used to generate B and D.



a) frequency divider by two



b) CMOS logic



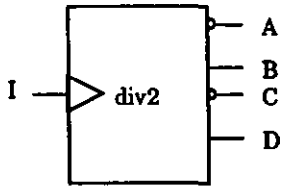
c) Timing waveforms

Figure 6.4 CMOS race-free static divider by two.

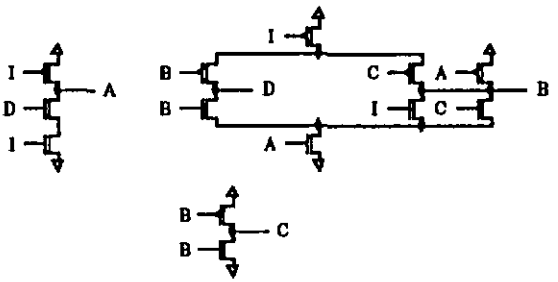
As can be seen from the timing waveforms, the transitions of B and D are derived directly from the input I and the auxiliary variable A whereas C and E change only after B and D. Thus, a carry ripple trough counter will be faster if the carry function is generated directly from A, B and D.

5.1.3.2 Semi-dynamic divider by two

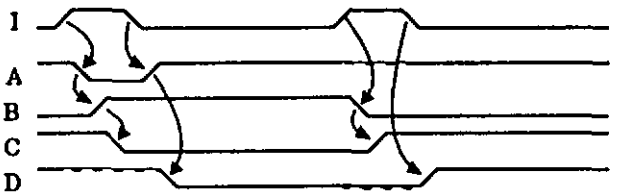
The semi-dynamic divider by two is derived from the static divider by two. Indeed, some transistors of the static device are used to complement the state of a variable whilst others only serve to hold the variable in a same state. If the clock pulses of I are high only during short intervals, the transistors used to hold variables in a same state during this time can be removed [Ogu74]. In the case of the divider by two of figure 5.4, this procedure enables to remove 6 transistors thus simplifying the design to 13 transistors. For means of comparison, a fully dynamic race-free divider by two requires 9 transistors [Ogu74].



a) frequency divider by two



b) CMOS logic



c) Timing waveforms

Figure 5.5 CMOS race-free semi-dynamic divider by two.

The dashed lines in the timing waveforms indicate that the variable is neither driven high nor low by any transistor. The variable is assumed to remain in the same logic state due to the capacitance between its node and the ground. The variable E of the static divider by two was used to keep D in the high state during the high state of the input I. It is no more necessary for the semi-dynamic device and has been removed.

The divider by two of figure 5.5 is dynamic for the high state of I and static for its low state. Eventually, the semi-dynamic divider by two could be redesigned in order to be dynamic for the low state if I and static for its high state.

5.2 UDC

The UDC is built up out of a set of identical bit slices, each containing a divider by two, and a control slice generating the command signals for the bit slices. The UDC inputs which are the clock phases clk1 and clk2, the incrementing command cntUp, and the decrementing command cntDn are fed to the control slice which processes them and creates the adequate signals for the control of the bit slices. Each bit of the UDC output is provided by one bit slice.

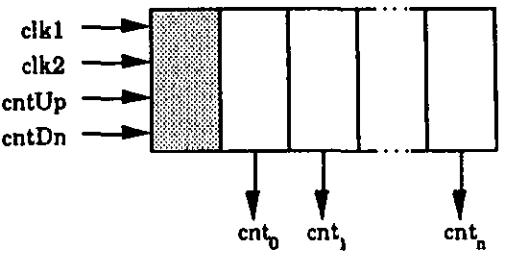


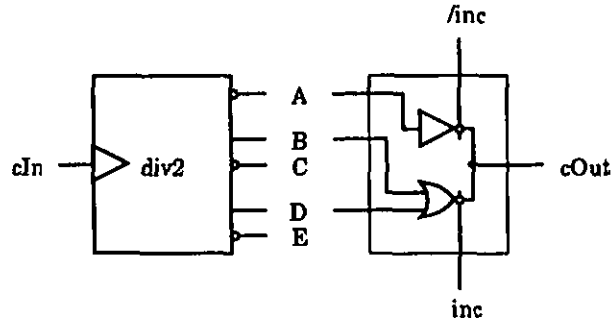
Figure 5.6 UDC slices.

First the bit slices are determined. As they are more numerous than the control slices, the surface minimization is achieved primarily for them, eventually at the expense of a greater surface in the control slice.

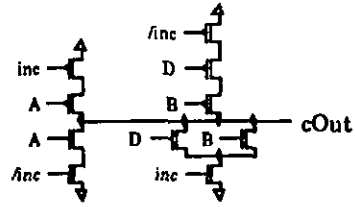
5.2.1 UDC bit slices

An asynchronous counter is realized by the cascade of dividers by two. If the output B of the dividers by two are considered to be the bits of the counter, the device is made to count up if the complementary output C is fed as a clock to the next divider by two of the chain. The device counts down when the direct output B is the ripple carry function which serves as clock for the next divider by two.

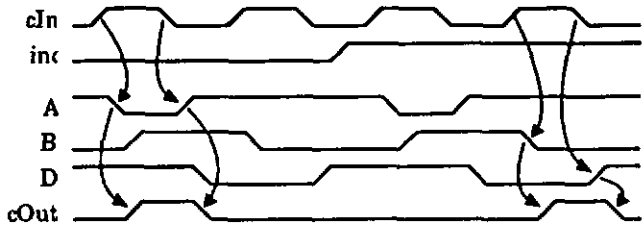
The UDC can thus be realized by inserting multiplexers between the dividers by two. The multiplexers are governed by a variable inc which determines the incrementing or decrementing mode of the counter. However, the inputs to the multiplexer should not be complementary functions because the switching from one to the other could result in a pulse of the multiplexer output [Bot89]. The multiplexed ripple carry logic is given in figure 5.7.



a) ripple carry multiplexer



b) CMOS logic



c) Timing waveforms

Figure 5.7 CMOS ripple carry logic.

As it is the case for the signals inside the divider by two, the output of the ripple carry logic is race-free as far as the control signals inc and /inc do not vary while cIn is high. It should also be noted that the carry out pulses are of same (and not of double) time extent as the carry in pulses. This is necessary for the use of semi-dynamic logic.

Additionally, the UDC must be checked for overflow and underflow to ensure proper saturation behavior. The carry output of the last slice provides this information, but unfortunately too late: the counter output has already wrapped around to the opposite end of the range when the carry output is signalled. Accordingly, overflow and underflow have to be checked

for by an appropriate logic. Underflow happens in decrement mode when all bits of the counter are zero. This can be checked by evaluating the logical OR of all bits of the counter. Similarly, overflow is checked by the means of the logical AND of all bits of the counter. The NOR and the NAND functions each require two transistors per bit slice [Bot89]. They realize the underflow control unf and the complemented overflow control /ovf.

As a result, the UDC bit slice is made out of three subcells: the divider by two, the ripple carry logic and the underflow-overflow logic

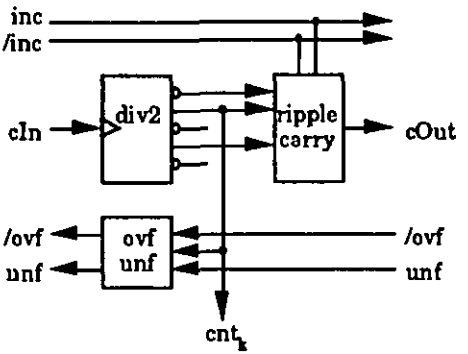


Figure 5.8 UDC bit slice.

The control slice has to prepare the inc and /inc controls for all the slices as well as the carry in pulse for the LSB slice.

5.2.1 UDC control slice

The control slice of a two input UDC receives the clock signals clk1 and clk2, the increment and decrement commands cntUp and cntDn as well as the underflow and overflow statuses unf and /ovf. From these inputs, the slice has to derive the global controls inc and /inc for all UDC bit slices as well as the carry in pulse /cIn to the first bit slice realizing the LSB.

The cntUp input can be redirected directly as the inc control and inverted for the complementary output. The carry in pulse /cIn is activated during phase 1, when cntUp is active while cntDn and ovf are inactive or when cntDn is active while cntUp and unf are inactive.

The ripple carry pulses are of same time extent as the clock phase 1 signal. Each pulse is a time skewed replicate of the one of the preceding slice. The sum of the delays between the ripple carry pulses is the main term setting the minimal period of phase 1. The rising edge of the last pulse must occur during this phase. The falling edge of the carry signals however can and most probably will occur during phase 2. To prevent the control signals inc and /inc to vary during a ripple carry pulse, both are latched during phase 2.

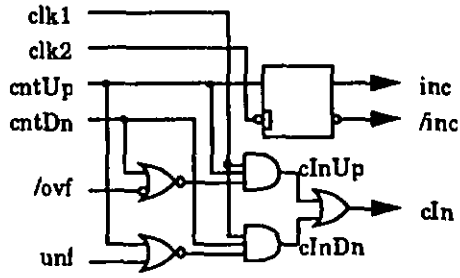


Figure 5.9 UDC control slice.

The CMOS realization of the control slice shown in figure 5.9 requires 36 transistors. The bit slices with semi-dynamic dividers by two count 27 transistors.

5.3 RM

The RM is realized in a similar manner as the UDC. It is made out of a set of identical bit slices commanded by a control slice. The bit slices realize a free running counter clocked by the pulse rate c during the clock phase 2. The counter output with the weight of the bits taken in the reverse order (the MSB considered as LSB and vice versa) implements a dither pattern used to quantify the RM output to a single bit. In this point of view, it is not important whether the counter counts up or down. The dither signal is added to the input x but only the MSB of the result is taken into account. The RM output is made out of the logical AND of this value and the input c. This output has to be registered for the next clock phase 1.

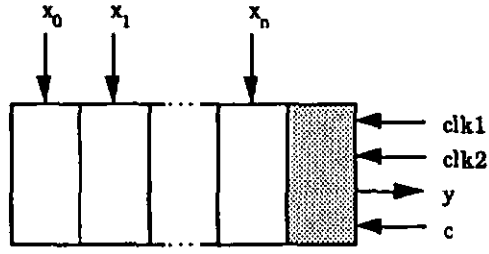


Figure 5.10 RM slices.

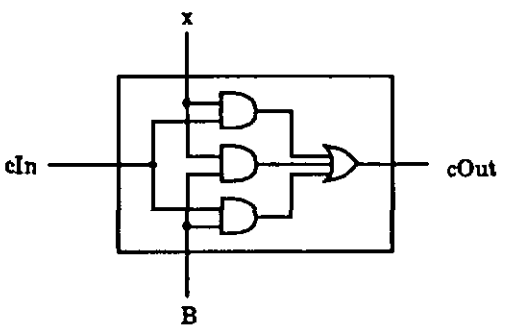
As for the UDC, first the RM bit slices are designed and then the control slice is determined to meet the requirements of the bit slices.

5.3.1 RM bit slices

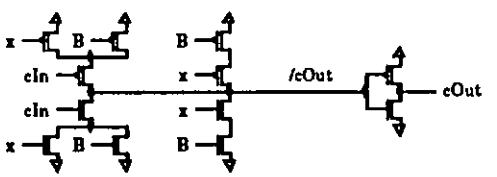
The RM bit slices implement a free running counter and the carry ripple logic of an adder. Indeed, only the MSB of the addition is taken into account. This MSB is given by the carry output of the last slice of the adder.

The counter can be realized as a chain of dividers by two. If a semi-dynamic divider by two is used, the carry propagation pulses must remain of short time extent. In the case of the divider by two of figure 5.5, the carry propagation pulse is best derived as the inverse of the variable A. Accordingly, the carry is propagated when the counter bit changes from low to high so the counter will be counting down.

The carry function of a two bit adder has three inputs: the two bits to add and the input carry. The output is true when at least two inputs are 1. A CMOS implementation of this function is proposed in figure 5.11. The carry input to the LSB of the adder must be set to zero.



a) carry function



b) CMOS logic

Figure 5.11 CMOS adder carry function.

Requiring 12 transistors, the adder carry function will have about the same size as the divider by two of the counter. Figure 5.12 presents the schematic of a RM bit slice.

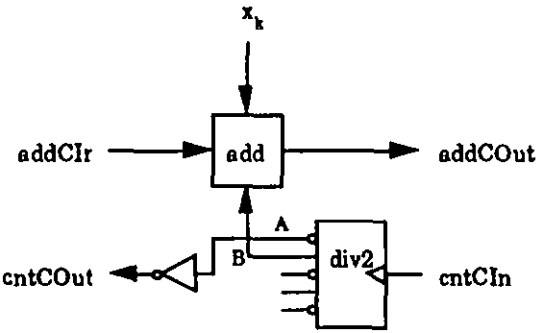


Figure 5.12 RM bit slice.

At every pulse of c, the RM has two operations to execute: decrement a counter and add two numbers. Nevertheless, decrementing the counter does not have to occur during phase 2. This task is best executed in the same time as the updating of the UDCs. Indeed, if the counter decrementation and the addition are carried out during the same phase, the delay from the rising edge of the input pulse to the settling of the output is the sum of both the carry ripple of the counter and the carry ripple of the adder. Conversely, if the decrementation is executed during the alternate phase, the result is available after only one carry ripple chain. Moreover, updating the RM counters at the same time as the UDCs results in having both outputs readily stable in the same time.

5.3.1 RM control slice

The RM control slice has to prepare the carry input to the counter LSB during phase 1. It also has to determine the logical AND between the pulse coded input c and the adder MSB. This value has to be stored as an input to UDCs during phase 1. Figure 5.13 is a graphical representation of these specifications.

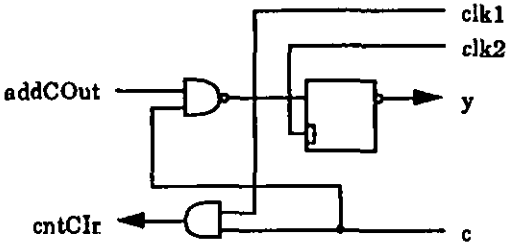


Figure 5.13 RM control slice.

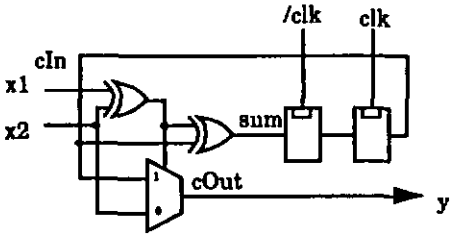
As the storage latch provides a complementary output, the AND gate comparing c and the carry output addCOut can be replaced by a more

economic NAND gate. This lessens the transistor count of the control slice to 20. The bit slices with semi-dynamic dividers by two count 27 transistors.

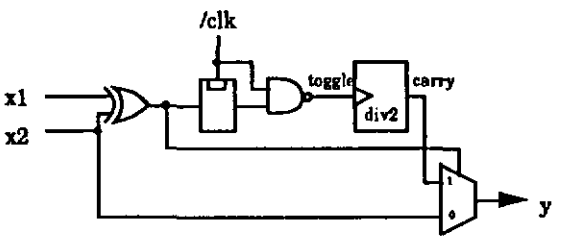
5.4 Adder

A bit-stream adder may be used for two purposes inside the ODLFs. For the UDCs having two decrementing inputs, either a special control slice has to be realized or the two decrementing inputs are added and the obtained sum is fed to an usual UDC. Also, the output of a lattice ODLF is the sum or the difference of the output of the two branches.

The bit-stream adder has been presented in chapter 2.3. The logic realizing this adder is presented in figure 5.14 a). An alternate realization is proposed in figure 5.14 b). This schematic enables to use an already designed divider by two instead of the register.



a) logic of the bit-stream adder



b) alternate realization using a divider by two

Figure 5.14 Bit-stream adder.

The adder counts 41 transistors. It can have the size of an UDC and a RM bit slice abutted together.

5.5 VLSI implementation

The VLSI implementation of the slices has been realized using the cmn16 technology of VTI (VLSI Technology Inc.) which is shrinked down from the cmn20s technology by a factor of 0.8.

The guiding rules for the layout of the slices was to draw the diffusion wells horizontally, the polysilicon lines vertically, the metal 1 horizontally and the metal 2 vertically. The metal 2 is mainly used for the power supply and, in the control slices, for the clock signals and the coefficients. The metal 1 interconnects the transistors and builds two large horizontal power supply lines, one on the top of the UDC slices and the other on the bottom of the RM slices. The polysilicon lines hold the logic variables and thus interconnect the cells of a slice. This approach leads to long polysilicon lines but proves to be efficient in terms of surface for highly interconnected cells as it is the case inside the divider by two or between the divider by two and the ripple carry logic of the UDC.

An *UDC* is realized by the horizontal abutment of one control slice and of the number of bit slices required for the operator. A RM is realized in a similar way, only the control slice lies on the opposite side of the bit slices. The two operators are then abutted vertically. Operator pairs are abutted horizontally such as the UDC control slice of one pair is placed above the RM control slice of the neighboring pair. By doing so, the output of the RM is automatically fed into the incrementing input of the UDC. The adder slice has been designed to be inserted between the rightmost bit slices and the corresponding control slices; they can also be abutted together. Once all operators have been abutted, the leapfrog feedback lines remain to be drawn.

Abutted together, one *UDC* and one *RM bit slice* expand over *40* λ x 287.5 λ, where λ stands for 1 μm in the cmn20s technology and for 0.8 μm in the cmn16 technology. If the top and bottom horizontal metal 1 power supply lines are not taken into account and the height of the slice is measured from the outermost sides of the bottom and top transistors, the active surface of the bit slice is of 40 λ x 261.6 λ. As it is composed of 64 transistors, the slice shows a density of 5'162 MOS/mm^2 in the cmn20s technology and of 8'066 MOS/mm^2 in the cmn16 technology. The control slices evidently have the same height but they are 70 λ wide; as they count 66 transistors, their density is only of 4'780 MOS/mm^2 in the cmn16 technology. The adder slice has the same dimensions as the abutted UDC and RM bit slice, but contains 41 transistors; the corresponding density is of 6'124 MOS/mm^2 in the cmn16 technology.

Figure 5.16 displays the basic ODLF slices. As they are placed, the slices can be abutted together horizontally and vertically.

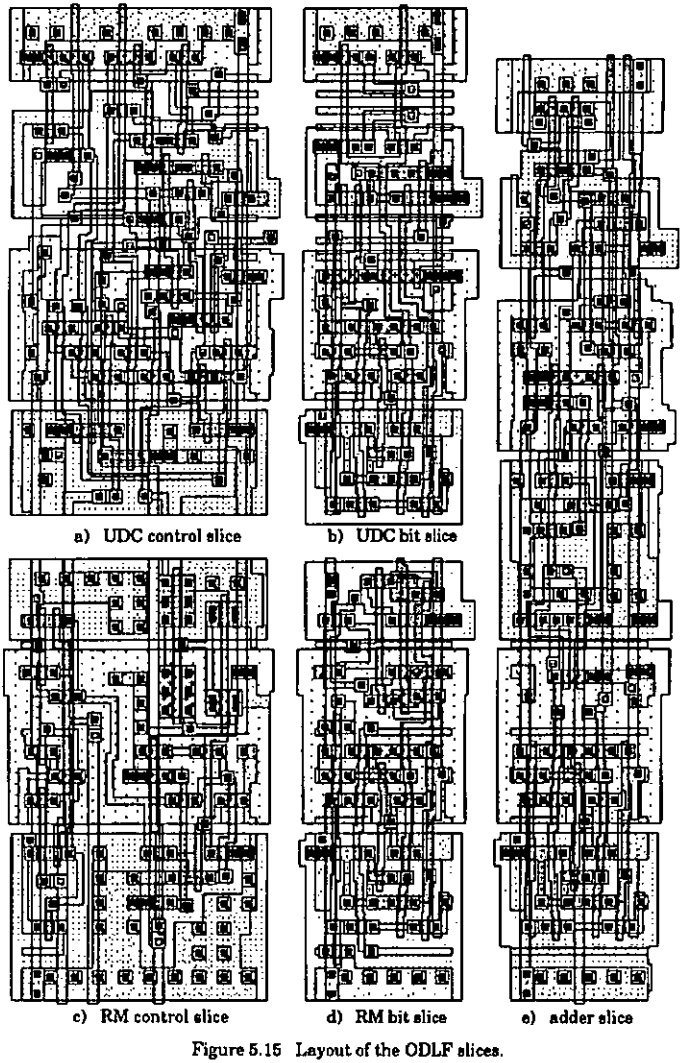


Figure 5.15 Layout of the ODLF slices.

Usual ODLF configurations use 10 bit operators. Accordingly, the surface used by an UDC-RM pair is a square of 122'905 λ^2 which corresponds to 0.078659 mm^2 in the cmn16 technology. In addition to this, the logic delivering the coefficient rate is required for the realization of one pole of the system. Hence, this filter realization necessitates about 0.1 ÷ 0.15 mm^2/pole. Figure 5.16 depicts the layout of a first order 8 bit ODLF.

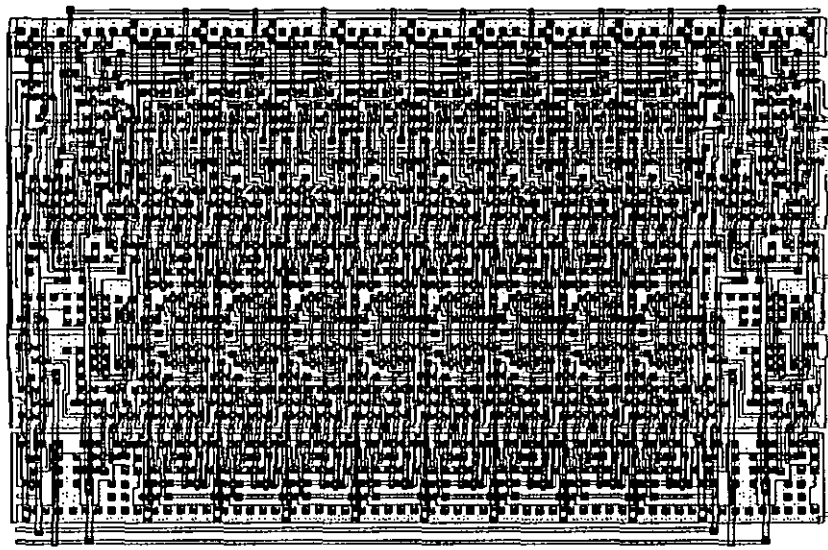


Figure 5.16 Layout of a first order ODLF.

The present work has focused on ASIC realization. Other realization types such as standard cells or Programmable Logic Devices (PLDs) are effective alternatives. The choice of one realization type rather than another one is dictated by the best available possibilities.

6 Design example

The best way to properly understand and to clarify a theory is to undergo an example. As an ODLF design example, a benchmark filter is proposed. The design leads all the way from the filter specifications to the VLSI realization.

The ODLF realization of the benchmark filter can be compared to various other implementations of the same filter.

6.1 Benchmark filter specifications

The CCITT specifications G712 PCM have been extensively used as a filter benchmark [Cla88], [Ans90]. These specifications correspond to a lowpass filter. The passband ripple is of ± 0.125 dB from 0 up to 3 kHz. The stopband attenuation is -14 dB from 4 kHz to 4.6 kHz and -32 dB higher. Table 6.1 summarizes these specifications.

Frequency band	frequency [kHz]	attenuation [dB]
passband	$< f_p = 3$	$\alpha_p = \pm 0.125$
stopband	$> f_{s1} = 4$ $> f_{s2} = 4.6$	$\alpha_{s1} = -14$ $\alpha_{s2} = -32$

Table 6.1 Benchmark CCITT G712 PCM filter specifications.

Figure 6.1 illustrates these specifications and displays a transfer function satisfying the requirements.

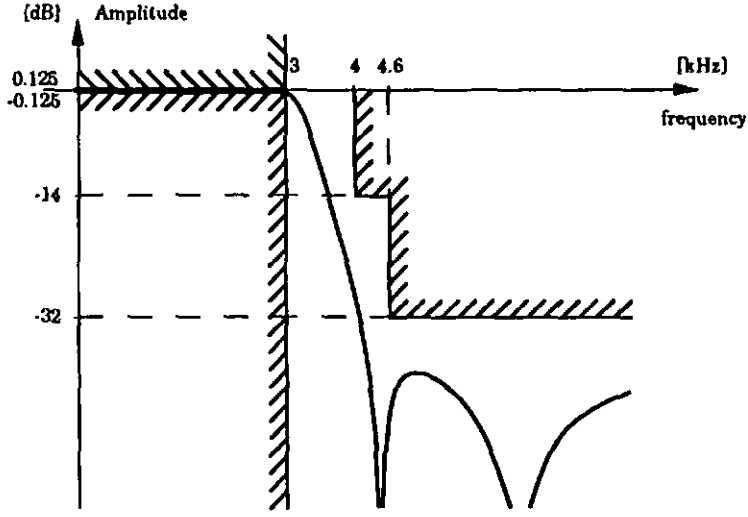


Figure 6.1 Benchmark CCITT G712 PCM filter specifications.

The filter characteristic has frequency dependant stopband attenuation specifications but can be approximated efficiently by classical lowpass functions. The approximation is achieved by setting the stopband attenuation to be -32 dB above 4.6 kHz. Then the transfer function is checked to verify that the attenuation of -14 dB at 4 kHz is fulfilled. If it is not the case, the approximation parameters are changed until the specifications are met.

6.2 Transfer function approximation

6.2.1 Filter type

The minimal filter order for the classical approximation functions satisfying the specifications is given in table 6.2.

Filter type	order
Butterworth	13
Inverse Chebyshev	8
Chebyshev	7
elliptic	5

Table 6.2 Minimal benchmark filter order for classical approximation functions.

From these, the elliptic filter is chosen because it requires the lowest filter order and so the minimal hardware. This filter type cannot be realized in the form of an all-pole ladder. The lattice realization requires an odd filter order. Thus the filter approximation will be a 5^{th} order elliptic filter.

6.2.2 Filter approximation parameters

The specifications presented in figure 6.1 can be met by different 5^{th} order elliptic filters. For example if a passband ripple of 0.125 dB is chosen, a stopband attenuation of 45 dB can be achieved. Inversely, if a stopband attenuation of 32 dB is chosen, a passband ripple of 0.006 dB can be obtained. The designer should take some margin when setting passband ripple and stopband attenuation in order to remain within the specifications even with the transfer function deviations due to quantization.

As the active filters derived from an analog LC prototype terminated in equal resistances show a small sensitivity to coefficients in the passband, the filter design margin will be mainly taken in the stopband. The parameters chosen are:

Filter parameter	symbol	specs [dB]	value [dB]
passband ripple	α_p	0.125	0.1
stopband attenuation	α_{s2}	32	40

Table 6.3 Elliptic filter design parameters.

From these specifications, the zeroes and the poles of the normalized prototype filter are determined.

6.2.3 Normalization

For reasons of efficiency, the computation of the filter values are preferably done for a normalized prototype.

As an exsmple, a filter of cut-off frequency of 1 kHz has poles of amplitude $\approx 10^3$, the time parametera such as group delay or rise time are of amplitude $\approx 10^{-3}$, the polynomials of the transfer function look like $(\approx 1)\cdot s^n + (\approx 10^3)\cdot s^{n-1} + (\approx 10^9)\cdot s^{n-2} + \ldots + (\approx 10^{3n})$. Theae differencea in the ordera of magnitude can lead to calculation errors or even computer underflow or overflow.

Frequency normalization is achieved by defining the normalized radian frequency:

$$\Omega = \frac{\omega}{\omega_0} = \frac{f}{f_0} \qquad (6.1)$$

where ω_0 or f_0 is a typical frequency of the filter, e.g. the cutoff frequency. In this way, the corresponding frequency of the normalized filter will be $\Omega = 1$. The dual time normslization is then given by:

$$\tau = \frac{t}{t_0} = \frac{2\pi}{\Omega} \qquad (6.2)$$

with:

$$t_0 = \frac{1}{\omega_0} = \frac{1}{2\pi \cdot f_0} \qquad (6.3)$$

In the case of elliptic filter design, normalization is preferably done in regard to the passband edge, this is:

$$f_0 = f_p \qquad (6.4)$$

The vslues of the zeroes and the poles of the normslized elliptic filter fulfilling the specifications given in table 6.3 are preaented in table 6.4.

zeroes	poles
$\pm 2.1725\,j$	$-0.1067 \pm 1.0637\,j$
$\pm 1.4690\,j$	$-0.4119 \pm 0.7935\,j$
	-0.6707

Table 6.4 Zeroes and poles of the normalized benchmark filter.

Ths location of the zeroes snd the poles of the normalized filter is plotted in figure 6.2. The corresponding (denormalized) transfer function is actusslly the one repreaented in figure 6.1.

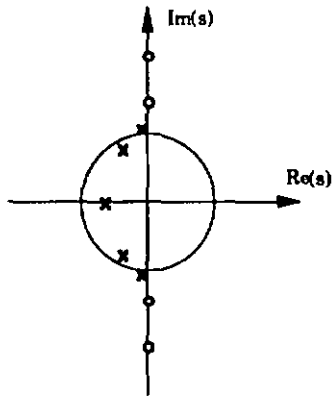


Figure 6.2 Zeroes and poles location of the benchmark filter.

Next, the element values of the reactances $Z_a(s)$ and $Z_b(s)$ of the analog prototype are obtained from the location of the poles.

6.3 Synthesis

6.3.1 Lattice reactance functions

The reactance functions $Z_a(s)$ and $Z_b(s)$ each serve to realize a subset of the transfer function poles. Writing reactance function $Z_a(s)$ as a ratio of polynomials

$$Z_a(s) = \frac{N_a(s)}{D_a(s)} \tag{6.5}$$

the partial transfer function H_a defined by equation (3.4) becomes

$$H_a(s) = \frac{N_a(s)}{N_a(s) + R \cdot D_a(s)} = \frac{N_a(s)}{G_a(s)} \tag{6.6}$$

where $G_a(s)$ is a Hurwitz polynomial also used for lattice wave digital filter synthesis. R is the generator and load resistance of the passive lattice prototype. It is normalized to 1 [Ω] for the calculations. As the transfer function of the lattice H(s) is realized by the parallel connection of the two branches $H_a(s)$ and $H_b(s)$, the poles of each partial transfer function is found in the overall transfer function H(s). The zeroes of the overall transfer function come from the parallel connection of the two branches.

For elliptic filters, the distribution of the poles of the transfer function H(s) among the lattice branches is described by [Gaz85]. The poles are distributed in an alternating order to the two branches $H_a(s)$ and $H_b(s)$, as represented in figure 6.3.

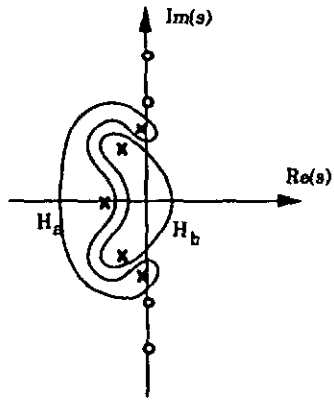


Figure 6.3 Alternating distribution of the poles among the lattice branches.

From the above considerations, the zeroes of the polynomials $G_a(s)$ and $G_b(s)$ have been determined. Retrieving $N_a(s)$ and $D_a(s)$ from $G_a(s)$ is facilitated by the property of the LC immitance $Z_a(s)$ that either its numerator is even and its denominator odd or vice-versa [Tem77]. Equation (6.6) indicates that $G_a(s)$ has to be split into its even part $G_{ae}(s)$ and its odd part $G_{ao}(s)$ to obtain the numerator $N_a(s)$ and the denominator $D_a(s)$ of the impedance $Z_a(s)$. Moreover, substituting the impedances Z_a and Z_b by their inverse $1/Z_a$ and $1/Z_b$ in the lattice filter results into the same transfer function [Tsm77]. Thus for the determination of $Z_a(s)$, one has the free choice to take either $G_{ae}(s)$ as the numerator $N_a(s)$ and $G_{ao}(s)$ as the denominator $D_a(s)$ or vice-versa.

Choosing $N_b(s)$ odd implies that $H_b(s)$ owns a zero at $s = 0$, as can be seen from equation (6.6). In order to obtain a lowpass function for $H(s) = H_b(s) - H_a(s)$, the numerator of $H_a(s)$ should be even. This gives:

$$\begin{cases} Z_a(s) = \dfrac{G_{ae}(s)}{G_{ao}(s)} \\ Z_b(s) = \dfrac{G_{bo}(s)}{G_{be}(s)} \end{cases} \qquad (6.7)$$

Obtained from the poles in table 6.4, the normalized lattice reactance functions are given in table 6.5.

branch	zeroes of g(s)	reactance function
a	$-0.1067 \pm 1.0637\,j$ -0.6707	$Z_a(s) = \dfrac{0.8840 \cdot s^2 + 0.7664}{s^3 + 1.2859 \cdot s}$
b	$-0.4119 \pm 0.7935\,j$	$Z_b(s) = \dfrac{0.8239 \cdot s}{s^2 + 0.7994}$

Table 6.6 Normalized lattice reactance functions.

The Cauer1 realization of the impedances Z_a and Z_b provides two branches in the form of a ladder. The values of the elements (impedances or capacitances) of the two branches will serve to determine the integrator gains of the active lattice branches and so the coefficients of the two ODLF branches.

6.3.2 Lattice element values

The values of the LC elements of the two branches is calculated by the continuous fraction expansion of the reactance functions [Tem77]. As the degree of the numerator of both reactance functions is smaller than the degree of the denominator, the Cauer1 realization of both impedances starts with a shunt capacitor. Figure 6.4 illustrates the two branches and table 6.6 provides the element values.

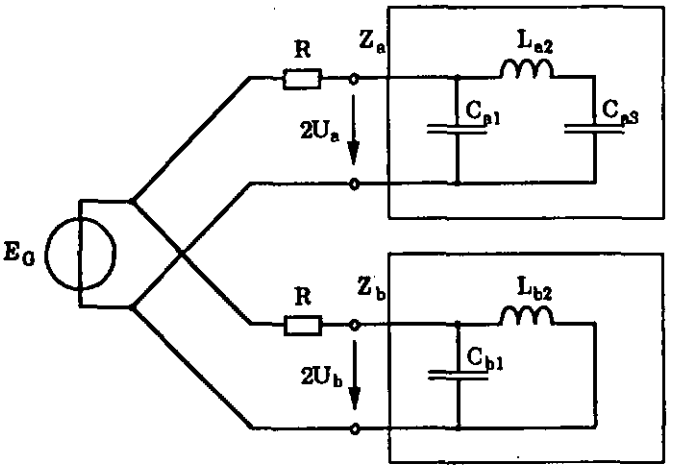


Figure 6.4 Cauer1 realization of the two lattice branches.

branch	C_1 [F]	L_2 [H]	C_3 [F]
a	1.1312	2.1105	0.5465
b	1.2138	1.0307	

Table 6.6 Normalized element values of the two branches.

From the element values of the passive lattice, the integrator gains of the active lattice are calculated.

6.3.3 Active lattice integrator gains

The integrator gains of the active lattice are radian frequencies proportional to the inverses of the element values. If the element is a capacitance C_k, the radian frequency is $\omega_k = 1/RC_k$ and if the element is an

inductance L_k, the radian frequency is $\omega_k = R/L_k$. The integrator gains are given in table 6.7.

branch	ω_1 [s^{-1}]	ω_2 [s^{-1}]	ω_3 [s^{-1}]
a	0.8840	0.4738	1.8297
b	0.8239	0.9703	

Table 6.7 Normalized active lattice integrator gains.

The ODLF coefficients correspond to the integrator gains of the active lattice.

6.3.4 ODLF coefficients

The ODLF coefficients are realized by pulse rates the values of which are normalized to the interval [0, 1]. Limiting the active lattice integrator gains to the interval [0, 1] is done by dividing them all by the highest gain value:

$$g_m = \max(\omega_k) \cdot 1[s] \qquad (6.8)$$

The resulting coefficients are provided in table 6.8.

branch	c_1	c_2	c_3
a	0.4832	0.2590	1
b	0.4503	0.5303	

Table 6.8 ODLF coefficients.

Dividing all the integrator gains of the active lattice by a constant means multiplying all the element values of the passive lattice by the same coefficient. This in turn corresponds to a frequency scaling of the transfer function.

The multiplication of all the element values of the passive lattice by the coefficient g_m displaces the passband edge $\Omega_p = 1$ to

$$\Omega_p = \frac{1}{g_m} \qquad (6.9)$$

The characteristic frequency is thus redefined as

$$f_o = f_p \cdot g_m \qquad (6.10)$$

The characteristic frequency of this particular design is given in table 6.9.

g_m []	f_p [kHz]	f_o [kHz]
1.8297	3	5.4891

Table 6.9 Characteristic frequency of the filter normalization.

The ODLF coefficients of table 6.8 are ideal values. Practical realization requires them to be quantized. Analysis allows to verify if the ODLF with quantized coefficients still meets the filter specifications and enables the designer to select the number of bits of the operators. A simulation is also welcome to verify if the analysis models are satisfactory.

6.4 Analysis

6.4.1 State-space description

The ODLF state-space matrices are built-up from the coefficient values. They are presented in table 6.10. The second row of matrices C and D correspond to the doubly complementary output.

$$A = \begin{bmatrix} -c_{a1} & -c_{a1} & 0 & 0 & 0 \\ c_{a2} & 0 & -c_{a2} & 0 & 0 \\ 0 & -c_{a3} & 0 & 0 & 0 \\ 0 & 0 & 0 & -c_{b1} & -c_{b1} \\ 0 & 0 & 0 & c_{b2} & 0 \end{bmatrix} \qquad B = \begin{bmatrix} c_{a1} \\ 0 \\ 0 \\ c_{b1} \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Table 6.10 State-space description of the ODLF.

The values of the coefficients are read from table 6.8. For a digital realization, they have to be quantized. In order to verify if the specifications are still met, the quantized coefficients are inserted into the state-space matrices and the corresponding transfer function is computed.

6.4.2 Coefficient quantization

A first approach for the quantization of the coefficients is to round them to the nearest binary value. In this manner, the coefficients given in table 6.8 can be rounded to a 6 bit representation with the corresponding filter still meeting the specifications. With the coefficients rounded to 5 bits, the filter meets the specifications if the characteristic frequency is increased to $f_0 = 6$ kHz.

Algorithms like simulated annealing or even, in the case of low order filters, trial and error can lead to far more economical solutions. In the case of the benchmark filter, an interesting set of coefficients is provided in table 6.11. This set of coefficients together with a new value of the characteristic frequency, given in table 6.12, supplies an efficient solution to the filter requirements.

branch	c_1	c_2	c_3
a	9/16	1/4	1
b	1/2	9/16	

Table 6.11 Quantized ODLP coefficients.

f_p [kHz]	f_o [kHz]
3	6

Table 6.12 Characteristic frequency of the filter with quantized coefficients.

Inserting the quantized coefficients of table 6.11 into the state-space matrices leads to the transfer function shown as the thick curve in figure 6.5. The figure also displays the ideal transfer function as a thin curve, the doubly complementary output and the individual transfer functions from the input to each state variable.

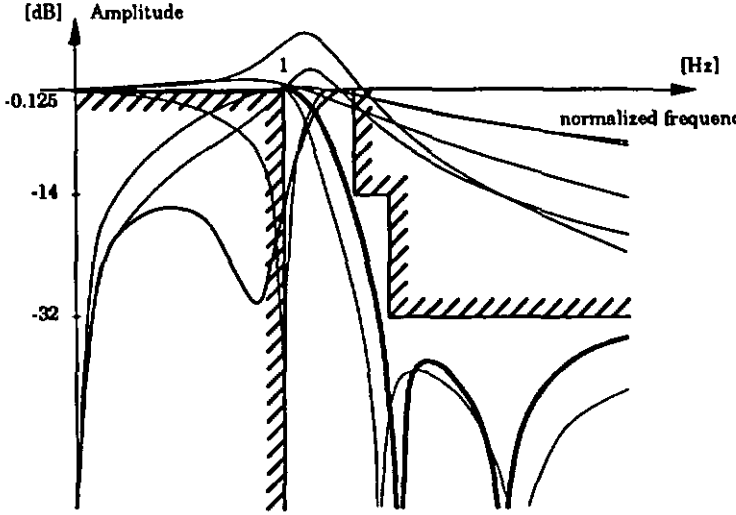


Figure 6.6 Transfer functions of the filter with quantized coefficients.

The quantization of the coefficients has lead to a distortion of the original frequency response. The quantization of the variables is regarded as noise sources and the number of bits of the ODLF operators has to be set in order to limit the effects of the noise sources to the filter output.

6.4.3 Signal quantization

The ranges of the L_2 norms and the L_∞ norms of the filter state variables, which correspond to the RM outputs, are presented in table 6.13.

The dynamic range of the UDC outputs is proportional to the range of the corresponding RM by the filter coefficient.

RM output	$\|w\|_2$	$\|w\|_\infty$
w_{a1}	1.329	1
w_{a2}	0.8862	1.421
w_{a3}	1.772	2.499
w_{b1}	1.253	1
w_{b2}	1.329	1.203

Table 6.13 Estimation of the RM output ranges.

The analysis of signal quantization inside the ODLF, as described in section 5.5.3, shows that the filter needs a 10 bit precision after the binary point to guarantee that the deviated transfer function does not overlap the stopband requirements. Even in this case, there is a slight overlap of the passband edge. This can be corrected by rescaling the characteristic frequency. The minimal and maximal value for f_o are provided in table 6.14.

n_{bits}	f_o [kHz]
10	6.04 ÷ 6.14

Table 6.14 Characteristic frequency range of the quantized filter.

One remark is appropriate here. The chosen elliptic transfer function approximation allowed to select a lower filter order but the stopband ripple brings the transfer function closer to the stopband requirements and so requires a better SNR in order to avoid an overlap. The choice of a Chebyshev approximation function could lead to a smaller number of bits required for the UDCs, allowing so a lower sampling rate, at the expense of a higher order device.

The above analysis has allowed to determine the number of bits of the RMs. The number of bits attributed to the individual UDCs are determined by the scaling of the signals.

6.4.4 Scaling of the signals

In order to allow a maximal dynamic range at the output of the UDC-RM pair, the ODLF coefficients are distributed among the two operators of the pair. First, a power of two scaling is realized by adding LSBs to the UDC, the rest of the coefficient is then provided as a pulse rate to the RM.

As an example, in the UDC-RM pair with coefficient 1/4, the UDC will have two additional LSBs which are not fed into the RM and the coefficient pulse rate will correspond to 1. The scaling factors and scaled coefficients

are presented in table 6.15 and the L norms of the scaled UDC outputs in table 6.16.

brench	c_1	c_2	c_3
a	$1 \cdot 9/16$	$1/4 \cdot 1$	$1 \cdot 1$
b	$1/2 \cdot 1$	$1 \cdot 9/16$	

Table 6.15 Scaled ODLF coefficients.

After this scaling, there will be three RMs with a coefficient of 1 and two RMs with a coefficient of 9/16. Thus, from an ASIC point of view, it might be interesting to generate the RM dither signals globally rather than locally in each RM.

UDC output	$\|w\|_2$	$\|w\|_\infty$
e_1	2.363	1.778
a_2	0.8862	1.421
a_3	1.772	2.499
b_1	1.253	1
b_2	2.363	2.139

Table 6.16 Estimation of the scaled UDC output ranges.

To prevent overflow, the input signal should be scaled or restricted to a prescribed range. The maximal amplitude is given by the inverse of the maximal L norm provided in table 6.16. The choice between the L_2 and the L_∞ norm, which depends on the estimated charecteristics of the input signals, is left to the designer.

6.4.5 Denormalization

Before any simulation or realization, the filter parameters have to be denormalized. It is not of interest here to denormalize the inductances, resistances and capacitances of the analog prototype. The ODLF coefficients however have been normalized to the range [0, 1]. It has been shown that their relative value to each other determines the shape of the transfer function, here an elliptic filter with 0.125 dB ripple in the passband and 32 dB attenuation in the stopband. The actual rate of their pulses will determine the frequency scale of the filter response. The maximal coefficient rate will determine the sampling frequency of the ODLF.

The normalized filter is realized with integrators of gain $c_k / j\Omega$ and has a passband edge of $\Omega = 1$. The denormalized filter has a passband edge of $f = f_0$ and the corresponding integrator transfer functions are

$$\frac{c_k}{j\Omega} = \frac{c_k}{j(\omega/\omega_0)} = \omega_0 \cdot \frac{c_k}{j\omega} \qquad (6.11)$$

The transfer function of a UDC-RM pair used as an as integrator is obtained from equations (2.5) and (2.9).

$$\frac{c_k}{j\Omega} = \frac{f_s \cdot c_k}{j\omega \cdot 2^{n_{bits}}} \qquad (6.12)$$

By comparing equations (6.11) and (6.12), we obtain the sampling frequency of the ODLF as a function of the number of bits of the operators:

$$f_s = \omega_0 \cdot 2^{n_{bits}} = 2\pi f_0 \cdot 2^{n_{bits}} \qquad (6.13)$$

To achieve the desired SNR, the system clock rate must be proportional to the power of two of the number of bits of the operators. The required values of the sampling frequency are shown in table 6.17 for the cases of 10 and 11 bit operators. This is of course the major limitation in the applicability of the structure.

n_{bits}	f_s [MHz]
10	38.9 ÷ 39.5
11	77.7 ÷ 79.0

Table 6.17 Sampling frequency range of the ODLF.

The values of table 6.17 provide the input rate of the RMs with coefficient 1. The sampling rate can of course be increased, but this would be unnecessary because the bit stream signals inside the device will never be of higher rate than the one of the highest coefficient. However, for a practical realization, it can be of interest to work with a system clock originating from a standard quartz oscillator. In the present case, a 40 MHz quartz can provide the sampling rate, and a pulse rate of 39 MHz must be derived from this clock to form a coefficient with amplitude 1, for example using a rate multiplier.

6.5 Simulation

The proposed simulation is carried out in two steps. First a zero signal is supplied to the filter input in order to check the DC working points of the operators. After this first test has been successful, sine wave inputs are applied to the device for the measure of its frequency response. The simulation is welcome to verify the correctness of the design, to confirm the exactness of the denormalization and to estimate the filter SNR.

6.5.1 DC working points of the operators

The study of ODLF synthesis has shown that the lattice branches can be realized with unsigned state variables. For this purpose, an offset U_o is added to the input signal and a DC value RI_o is fed into the negative input of the last integrator of the chains.

From these offsets, the DC working points of the state variables for the case of a zero input are determined. These values can be obtained from the analysis of the analog impedances where the inductances are considered as short circuits and the capacitances as open circuits. An alternate method to

find the DC working points of the state variables is to require all integrator inputs of the active device to be zero.

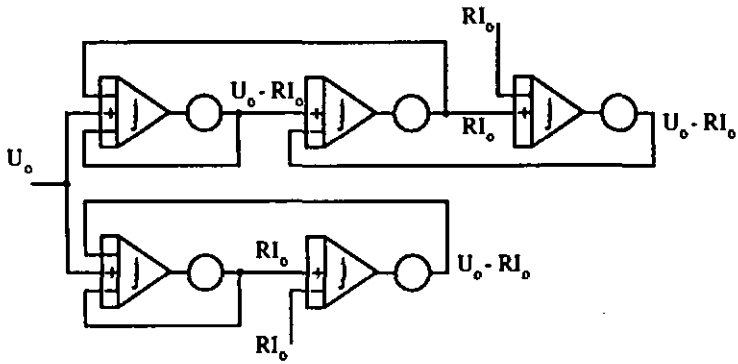


Figure 6.6 Working points inside the lattice filter.

The working points of the state variables inside the lattice filter are represented in figure 6.6. The offsets RI_0 and U_0 are chosen to allow a optimal range for the state variables. As all scaled coefficients are greater than one half, the possible values for RI_0 range from zero to one half. As some of the state variables have a working point of $U_0 - RI_0$, U_0 ranges from RI_0 to one. The actual choice of RI_0 and U_0 is determined by the following considerations.

Let X_0 be the DC value of the active filter input, which can be positive or negative. The input $X_0 + U_0$ is restricted to the interval [0, 1]. Consequently

$$X_0 \in [-U_0, 1-U_0] \tag{6.14}$$

Likewise, some internal state variables have the DC amplitude of $U_0 + X_0 - RI_0$ which must also be in the interval [0, 1]. Hence

$$X_0 \in [-U_0+RI_0, 1-U_0+RI_0] \tag{6.15}$$

As a result of the two preceding conditions, the DC input range is confined to the interval

$$X_0 \in [-U_0+RI_0, 1-U_0] \tag{6.16}$$

The condition for this range to be symmetric around zero gives

$$U_0 = \frac{1+RI_0}{2} \tag{6.17}$$

In this case, the condition (6.16) can be reformulated as

$$X_0 \in \left[-\frac{1-RI_0}{2}, \frac{1-RI_0}{2}\right] \tag{6.18}$$

With the above, the choice of RI_0 leads directly to the value of U_0 and the range of X_0. Two interesting working points are $RI_0 = 1/2$ and $RI_0 = 1/3$. The corresponding ranges are given in table 6.18.

RI_o	U_o	X_o	U_o-RI_o
1/2	3/4	[-1/4, 1/4]	1/4
1/3	2/3	[-1/3, 1/3]	1/3

Table 8.18 DC working points of the lattice ODLF.

For the simulation, the working point $RI_o = 1/2$ has been chosen. In this case, the DC working points of the operators for the case of a zero input are displayed in table 6.19

variable	UDC	RM
a1	0.4444	1/4
a2	2	1/2
a3	0.25	1/4
b1	1	1/2
b2	0.4444	1/4

Table 6.19 Estimation of the scaled UDC output ranges.

One further remark appropriate here is that the filter output, which is made out of the difference of the first state variable of each branch, also possesses an offset. Choosing $RI_o = 1/2$ leads to an output offset of $2RI_o$-U_o. For the complementary output, however, the offsets cancel each other and the total offset is zero.

Once the building blocks have been described correctly, simulation results show that the outputs of the UDCs stabilize to the values listed in table 6.19 after the vanishing of the transients. Now comes the time to verify the frequency response of the device.

6.5.3 Frequency response

The measure of the frequency response is done by applying a sine wave of a given frequency as an input to the filter and estimating the spectral content of the output at the same frequency. This task is repeated for all frequencies to be measured. The simulation has been done for a 10 bit device clocked at 39 MHz.

Choosing the $RI_o = 1/2$ means that the maximal dynamic range of the input signal is one half, as can be read from table 16.18. Thus, for the estimation of the transfer function deviation due to signal quantization, the input has a maximal range of 9 bits. The choice of a maximal range of 9 bits, which does not fulfill the specifications, originates from the difficulty to simulate at frequencies small compared to the sampling rate. Nevertheless, the main interest of the simulation is to compare the analytic and the measured transfer function deviation due to signal quantization.

Care has to be taken to ensure the validity of the measures. First, as the frequency of the sine wave is changed, the operator has to wait for the transients to clear before he can make any trustworthy measure. The time

to wait can be determined from the shape of the impulse or step response of the filter. Another method is to look at the Lissajou plot made by the combination of the filter input and output. In addition, if the input is not properly scaled, overflow can occur inside the filter without the output bearing a chaotic aspect.

The result of the measure of the frequency response is displayed in figure 6.6 together with the analytic transfer function and its expected deviation due to signal quantization. The curves plotted in the figure are tabulated in annex A.

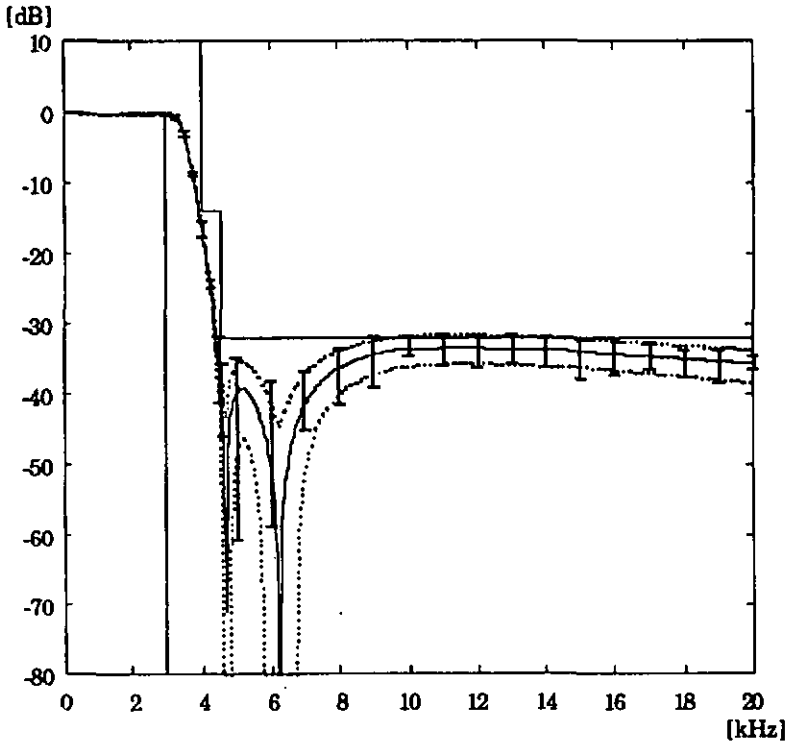


Figure 6.7 Measure of the frequency response of the DDLF.

The measure of the transfer function deviation does not show any significant peaking at the locations of the spectral lines of the dither patterns, as depicted in figure 2.6. Table 6.20 provides the base frequencies of the dither patterns spectra.

branch	[kHz]	[kHz]	[kHz]
a	21.4	9.52	38.1
b	19.0	21.4	

Table 6.20 Estimation of the scaled UDC output ranges.

For highly oversampled devices, the measure of the frequency response is extensive in terms of CPU time and memory usage. In this particular case, the simulation of one period of a sine wave in the transition band requires about one minute on a Macintosh IIci, so the measure of one frequency point takes something like one quarter of an hour. In addition, the simulation of one period of a sine wave of frequency equal or lower than 1 KHz resulted in a "not enough memory" message on the machine equipped with 5 Mb.

An interesting alternative to the software simulation of medium size digital systems is to download these into a PLD. However, the realization method discussed here is the implementation as a full custom integrated circuit.

6.6 Realization

The VLSI implementation of the 5^{th} order lattice ODLF has been realized on a multi-project wafer using a 1.6 μm technology (cmn16 of VLSI).

In order to test different filter types, the ODLF has been implemented without any scaling of the counters. The operators were designed using 11 bit.

The realization of the operators and the branches has been done by the abutment of the basic slices described in chapter V. The second order branch makes a rectangle of 1'130 λ x 261.6 λ, where λ stands for 0.8 μm. The branch occupies a surface of 295'495 λ^2 which is actually 189'117 μm^2. The third order branch expands over 1'640 λ x 261.5 λ = 428'860 λ^2 = 274'470 μm^2.

In addition to the two branches, an input block has been added for the synchronization of the input and the coefficients to the clock. This block is made out of six identical slices. As output logic, two adder slices have been placed in order to derive the complementary outputs of the filter. With these, the filter covers a rectangle of approximately 1.45 mm x 0.6 mm which is 0.87 mm^2.

The corresponding layout is represented in figure 6.7. The synchronization block is placed at the bottom left of the cell. The second order branch is located above it. The third order branch occupies most of the right hand side of the layout. In both branches, the first UDC of the chain is implemented by a 2 input device and an oversampled adder which combines the two decrementing inputs into one. The control slices come into view because of their smaller density of transistors; this allows to identify the different UDC-RM pairs. The adder slices are located at the bottom right of the cell. Compared to the lattice branches, the adder block is so small that the doubly complementary output is indeed obtained with a very little hardware overhead. Also, the LF feedback links hardly require any surface as the state variables are coded on a single bit.

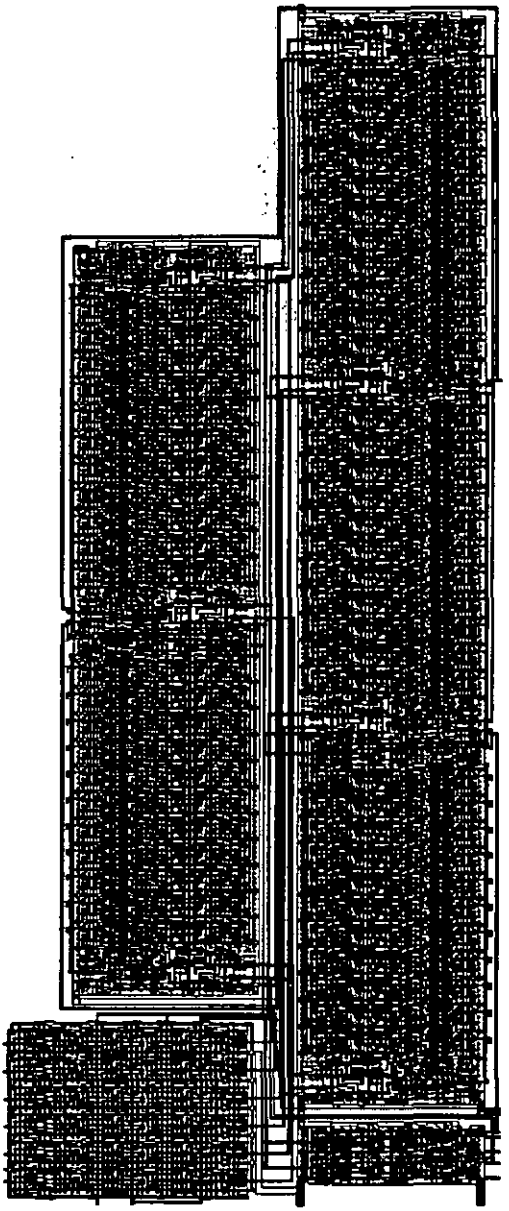


Figure 6.8 VLSI realization of the 5th order ODLF.

In order to work, the filter requires a pulse frequency modulated input and the five coefficients. It delivers a pulse frequency modulated output. For the benchmark specifications, the coefficients could have been generated on

chip, requiring about the same amount of logic than the coefficient synchronization block which would then no more be necessary.

6.7 Computer Aided Design

The design steps described in the paragraphs 6.2 to 6.5 have been traversed with the help of a signal processing set of routines written using MATLAB. MATLAB is an interactive software package for scientific an engineering applications that integrates numerical analysis, matrix computation and graphics. It is released by The MathWorks Inc and runs on a large number of platforms including IBM PC, Macintosh, Vax and HP-Apollo. The package allows the developer to define his own set of functions and routines based on the basic set of commands. Furthermore, some general purpose functions and routines are provided with the package.

The set of routines written for the ODLF design [Cor91a] comprises a filter approximation, a filter synthesis and a filter analysis tool. The filter approximation tool allows to determine the zero and pole locus of classical transfer functions such as Butterworth, Chebyshev, Elliptic, and Bessel from the specifications. The synthesis tools determines the element values or the coefficients of different filter types from the zero and pole locus. The analysis tool derives the state-space description of the synthesized filter and allows the quantization analysis of the filter.

The simulation has been carried out using the LabVIEW software from National Instruments [Cor91b]. The choice of this environment, besides the interest of investigating a new tool, arose from the comfortable user interface which allows the on-line inspection of the parameters passed to and retrieved from subroutines called by the main program during its execution. Furthermore, using LabVIEW, the description of an object such as a digital filter is done by drawing the data flow block diagram rather than writing the control flow in a programming language.

7 Conclusions

In order to estimate the interest of ODLFs, this filter type is compared with two other filter structures, namely switched capacitor and conventional digital filters.

General considerations about the operators and their limitations are made to point out some possible extensions to the present work.

The characteristics of ODLFs suggest the kind of domains where their application are of particular interest.

7.1 Comparison with other filter structures

The ODLF is a digital filter derived from an analog leapfrog structure. As such, its characteristics will be compared to the ones of its parents: the analog Switched Capacitor (SC) and the digital filter.

Comparison results are summarized in table 7.1. The characteristics of SC filters have been kindly provided by N. Häberle from Ascom Autophon AG. Data about WDFs originate mainly from [Cla88]. Obviously, the most particular characteristic of ODLFs is their high clock rate to cutoff frequency ratio. As a matter of fact, ODLFs are limited to very selective filter requirements due to their oversampling whilst the usual filter types start to have very sensitive coefficients (or capacitor ratios) as the cutoff frequency becomes small compared to the sampling rate.

structure		SC	ODLF	WDF
signals		analog	binary	digital
interfaces		S/H	$\Sigma\Delta$	A/D, D/A
Si surface	[mm^2/pole]	0.1	0.15	0.5 ÷ 5
SNR	[dB]	> 70	60	96
consumption	[µA/pole]	10	50 / MHz	
max. cutoff frequency	[kHz]	200	5	100
system clock to cutoff freq. ratio		10 ÷ 200	> 1'000	10 ÷ 500

Table 7.1 Comparison of filter structures.

The values for the silicon surface required by the devices only take in account the filters themselves without any input or output interface. Typically, if the input and output signals of the system are analog, all devices require an input anti-aliasing filter and an output smoothing filter. Additionally, the SC filter could need an input Sample and Hold (S/H) circuit, the ODLF can do with an input $\Sigma\Delta$ modulator and the digital filter requires an A/D and a D/A converter.

For an ODLF using a $\Sigma\Delta$ converter, the input anti-aliasing filter does not meet very stringent specifications as the transition band between the input spectrum and the sampling frequency is very large. A typical $\Sigma\Delta$ converter itself occupies about 1 mm^2 and consumes some 10 µA. However, the severeness of the output smoothing filter is a determining factor for the SNR of the system.

The non-linearities of the ODLFs is of same kind as the ones of digital filters. They are due to the quantization of the digital signals and their importance is estimated by the same analysis methods. Moreover, the SNR of the device can be directly influenced by the number of bits and the oversampling of the operators. On the other side, SC filters suffer from electronic noise and non-idealities of the amplifiers so the SNR is given by

the Integrated Circuit (IC) technology rather than by a filter design parameter.

The characteristics of ODLFs are quite similar to the ones of SC filters, only the oversampling is greater. As such, they show to become an interesting alternative to SC filters as the cutoff frequency decreases versus the sampling rate.

Compared to classical digital filters, ODLFs emerge through their ease of design and the simplicity of their operators. A peculiar characteristic of ODLFs is that the coefficients are given in the form of pulse rates whose generation requires silicon surface. Furthermore, multiplexing is of poor interest for ODLFs as the data busses would dramatically increase the surface of the device and the system clock frequency would further have to be raised. Practical examples have shown that ODLFs can cope with a smaller number of bits than classical digital filters.

7.2 Possible extensions to the work

Each ODLF operator contributes in its own way to the limitations of the structure. The incremental behavior of the UDC is responsible for the high oversampling whilst the dithering of the RM is the source of the quantization noise inside the device.

A reduction of the oversampling can only be achieved by the coding of the state variables on more than one bit. The use of three or four bits for the state variables can lead to ODLF performances comparable to the ones of classical digital filters as in the latter devices the system clock is quite higher than the sampling rate of the filter. This change naturally leads to the replacement of the UDCs by accumulators which will be realized in one part by an adder for the LSBs and in a second part by an incrementer for the MSBs.

The SNR provided by dithering inside the RMs showed to be sufficient because of the oversampling required by the ODLF structure. If the number of bits of the state variables is increased, the dithering quantizer has to be replaced by a $\Sigma\Delta$ converter and the multiplier which was realized by an AND gate will also show to become more considerable.

The optimal number of bits for the state variables certainly depends on the application. As the oversampling reduces, the DDI transform has to be replaced by an Lossless Discrete Integrator (LDI) transform [Bru75] and analysis has to be made with the time discrete (sampled) rather than with the time continuous model. Nevertheless, the proposed enhancements remain in contradiction with the first purpose which was the simplicity of the realization.

The operators have been designed using unsigned arithmetic. This choice has lead to the need for a DC offset to all state variables and necessitates a constant input RI_0 to the last UDC of each branch in the lattice synthesis. The definition of signed operators can easily be derived

from the workings of the actual ones. Yet, this modification is costly in terms of size and regularity of the saturation and multiplication logic.

The workings of the UDCs together with RMs are very similar to the description proposed for the neurons in the research about neural nets. For this reason, the use of these operators could be a valuable alternative to actual analog circuits for the hardware realization of neural networks.

7.3 Domain of interest for the ODLFs

The high oversampling is the major limitation to the applicability of the ODLFs. As a result, they are appropriate for the processing of low frequency signals. Possible purposes range from handling the measures of slowly varying physical quantities up to processing of telephone signals.

The shape of the ODLF inputs and outputs which are coded as PFM signals prone for their relevance in fields such as the measure of L, R or C impedances, the transmission of binary signals in noisy environments, the reading of quartz sensors, the vibration analysis of rotating machines or the regulation of stepping motors.

Important to note is that the oversampling of the ODLFs depends straightaway from the required SNR. For this reason, the use of higher order filters is mostly of interest for specifications bearing narrow transition bands rather than severe attenuations.

In addition, the ODLF delivers in its passband an output frequency corresponding to the input rate. As such, it can be regarded as a digital Phase Locked Loop (PLL) and applied for tasks such as digitally controlled frequency synthesis or the demodulation of frequency modulated signals.

Acknowledgements

This work has been realized at the Institute for Microtechnology of the University of Neuchâtel under the direction of Professor Dr F. Pellandini. Hereby I express him my gratitude for the advice he has kept providing me. I also want to thank Dr P.-A. Farine who worked on the same topic before me and spent a lot of time to introduce me in this subject and more generally in the domain of digital signal processing.

Likewise, I would like to thank the other members of the jury, Pr Dr M. Kunt and Pr Dr N. De Rooij, for their efforts and contributions.

Not the least I intend to thank my colleagues at the Institute and my family for the help and encouragement I received. Let me mention in particular

- Philippe Thévenaz for the time he spent to listen to the description of my problems and for the critical answers he provided, as well as for the maintenance of our common Macintosh,
- Ulf Sjöström for valuable information in the field of digital signal processing,
- Michel Ansorge and Ivan Defilippis for their support in VLSI design,
- Javier Braccamonte for the collaboration during the testing of the integrated circuits,
- Jean-Pierre Amann and Daniel Glauser for all they taught me about applied electronics,
- my parents who supported my education as an engineer,

and all the people with whom life has been great, so great that it also made sense for me to go to work, even when snow conditions have been outstanding.

My Ph.d work at the Institute has been supported by the Swiss National Foundation for Scientific Research under project numbers FN 20-5627.88 and FN 20-28536.90.

Abbreviations

A/D	Analog to Digital (converter)
ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal-Oxyde-Semiconductor (transistors)
D/A	Digital to Analog (converter)
DDI	Direct-transform Digital Integration
IC	Integrated Circuit
IIR	Infinite Impulse Response (filter)
LDI	Lossless Discrete Integrator (transform)
LF	LeapFrog (structure)
LSB	Least Significant Bit
LWDF	Lattice Wave Digital Filter
MIMO	Multiple Input Multiple Output (filter)
MSB	Most Significant Bit
ODLF	Oversampled Digital LeapFrog (filter)
PLD	Programmable Logic Device
PLL	Phase Locked Loop
QCDF	Quasi-Continuous Digital Filter
RM	Rate Multiplier
RMS	Root Mean Square
SC	Switched Capacitor (filter)
SISO	Single Input Single Output (filter)
SNR	Signal to Noise Ratio
UDC	Up-Down Counter
VLSI	Very Large Scale Integration

Glossary

Symbol	Significance
A	State matrix of a system
B	State matrix of a system
C	State matrix of a system
C_{ak}	Element value of the capacitance k of the branch a
C_{bk}	Element value of the capacitance k of the branch b
C_k	Element value of the capacitance k
c	ODLF coefficient, pulse rate input of a RM
c_{ak}	ODLF coefficient of branch a
c_{bk}	ODLF coefficient of branch a
c_k	ODLF coefficient
D	State matrix of a system
D(s)	Denominator of a transfer function
D(ω)	Transfer function deviation
$D_a(s)$	Denominator of the reactance function $Z_a(s)$
$D_e(s)$	Odd part of the denominator of a transfer function
$D_o(s)$	Odd part of the denominator of a transfer function
d	Dither signal inside a RM
E_G	Voltage on the generator used to drive the analog filter input
e	Quantization error
f	Frequency
F(s)	Signal gain function
f_b	Signal bandwidth
f_0	Typical frequency of a filter
f_p	Passband edge
f_s	Sampling rate
f_s	Stopband edge
G(s)	Noise gain function
$G_a(s)$	Hurwitz polynomial used for lattice synthesis
$G_{ae}(s)$	Even part of $G_a(s)$
$G_{ao}(s)$	Odd part of $G_a(s)$
$G_b(s)$	Hurwitz polynomial used for lattice synthesis
$G_{be}(s)$	Even part of $G_b(s)$
$G_{bo}(s)$	Odd part of $G_b(s)$
g_m	highest integrator gain value of the active LF
H	Transfer function of a filter
$\overline{H}$	Complementary transfer function of a filter
H_a	Transfer function of a lattice branch
H_b	Transfer function of a lattice branch
H_k	Transfer function between the input and the state variable k
$h_k(t)$	Impulse response corresponding to the transfer function H_k
I_1	Port current of the two-port
I_2	Port current of the two-port

I_{Ck}	Current in the capacitance k
I_{Lk}	Current in the inductance k
I_a	Current in a lattice filter reactance
I_b	Current in a lattice filter reectance
I_o	Additional offset for a lattice filter with positive state variables
L_{ak}	Element value of the inductance k of the brench a
L_{bk}	Element value of the inductance k of the branch b
L_k	Element value of the inductance k
N(s)	Numerator of a transfer function
$N_a(s)$	Numerator of the reactance function $Z_a(s)$
n_{bits}	Number of bits of the digital representation of a signal
osr	Oversampling ratio
R	Scaling resistance of the active filter state variables
R_G	Internal resistance of the analog filter input generator
R_L	Load resistance on the analog filter output
S_a	Reflectance function of a lattice wave digital filter
S_b	Reflectance function of a lattice wave digital filter
s	Complex frequency variable for time-continuous systems
T	State transformation matrix
T_s	Sampling period
t	Time
U_1	Port voltage of the two-port
U_2	Port voltage of the two-port
U_a	Voltage over a lattice filter reactance
U_b	Voltage over a lattice filter reactance
U_o	Input offset for a filter with positive state variables
U_{Ck}	Voltage over the capacitance k
U_{Lk}	Voltage over the inductance k
w	State variable of a system
w_{ak}	State variable in the branch a
w_{bk}	State variable in the branch a
w_k	State variable of a system
X_k	Reactance, stands for L_k or C_k
X_o	DC amplitude of the input signal x
x	Input of a filter or an operator
x_{max}	Maximal amplitude of the input x
x_q	Quantized value of x
y	Output of a filter or an operator
Z	Impedance
Z_a	Impedance of a lattice branch
Z_b	Impedance of a lattice branch
z	Complex frequency variable for sampled systems, $z = e^{sT_s}$
α_p	Attenuation in the passband
α_s	Attenuation in the stopband
Δ	Difference between to contiguous samples of a signal
λ	Technology size parameter
τ	Normalized time

Ω	Normalized radian frequency
Ω_p	Normalized radian frequency of the passband edge
ω	Radian frequency , $\omega = 2\pi \cdot f$
ω_b	Signal bandwith radian frequency , $\omega_b = 2\pi \cdot f_b$
ω_k	Active lattice integrator gain of the active LF
ω_o	Typical radian frequency of a filter
ξ_q	Signal to noise ratio associated to the quantization of a signal
ψ	Complex frequency variable used with wave digital filters

References

1 Introduction

Publications

[Gir70] F. E. J. Girling, F. E. Good, "Active filters 12 - The leapfrog or active-ladder synthesis", Wireless World, vol. 76, pp 341-345, July 1970.

F. E. J. Girling, F. E. Good, "Active filters 13 - Applications of the active-ladder synthesis", Wireless World, vol. 76, pp 445-450, Sept. 1970.

[Bru75] L. T. Bruton, "Low-sensitivity digital ladder filters", IEEE Trans Circuits and Syst., vol. CAS-22, No 3, pp 168-176, March 1975.

[Nei81] J. Neirynck, "Introduction aux principes des filtres électriques", Journées d'électronique: Techniques modernes de filtrage, Presses Polytechniques Romandes, Lausanne, pp 1-11, 1981.

[Far84] P.-A. Farine, F. Pellandini, "On the behavior of quasi-continuous digital filters", International conference on digital signal processing, pp 27-31, Florence, Italy, September 1984.

[Can85] J. C. Candy, "A use of double integration in sigma delta modulation", IEEE Trans. Comm., Vol. COM-33, No 3, pp 249-258, March 1985.

[Far85] P.-A. Farine, F. Pellandini, "Technique de filtrage numérique quasi-continu. Avantages et limitations", Dixième colloque sur le traitement du signal et ses applications, pp 449-454, GRETSI, Nice, France, 1985.

Books

[Tem77] G. C. Temes, J. W. LaPatra, "Introduction to circuit synthesis and design", International Student Edition, McGraw-Hill Kogakusha Ltd, Tokyo, 1977.

[Cou84] F. de Coulon, "Théorie et traitement des signaux", Traité d'Electricité, Editions Georgi, Lausanne, 1984.

[Far86] P.-A. Farine, "Filtrage numérique caractérisé par un traitement quasi-continu des signaux", Thèse de doctorat, Université de Neuchâtel, Suisse, 1986.

2 System and signals

Publications

[Cha90] K. C.-H. Chao, S. Nadeem, W. L. Lee, C. G. Sodini, "A higher order topology for interpolative modulators for oversampling A/D converters", IEEE Trans. on Circuits and Syst., Vol. CAS-37, No 3, pp 309-318, March 1990.

[OLe90] P. O'Leary, F. Maloberti, "Bit stream adder for oversampling coded data", Electronics letters, Vol. 26, No 20, pp 1708-1709, Sept. 1990.

3 Synthesis

Publications

[Gaz85] L. Gazsi, "Explicit formulas for lattice wave digital filters", IEEE trans. on Circuits and Syst., vol. CAS-32, no 1, pp. 68-88, Jan. 1985.

[Mit85] S. K. Mitra, Y. Neuvo, P. P. Vaidyanathan, "Complementary IIR digital filter banks", Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, Tampa, Florida, 1985.

[Fet86] A. Fettweis, "Wave digital filters: theory and practice", Proc. IEEE, Vol. 74, No 2, pp 270-327, February 1986.

[Joh89] D. A. Johns, W. M. Snelgrova, A. S. Sedra, "Orthonormal ladder filters", IEEE Trans. Circuits and Syst., Vol. CAS-36, No 3, pp 337-343, March 1989.

Book

[Saa79] R. Saal, "Handbook of filter design", AEG-TELEFUNKEN, Berlin, 1979.

4 Analysis

Publications

[Kam77] K. D. Kammeyar, "Quantization arror analysis of the distributed arithmetic", IEEE trans. on Circuits and Syst., vol. CAS-24, no 12, pp. 681-689, December 1977.

[Cla84] L. J. M. Claasen, H. J. De Man, J. Vandewalle, "DIGEST: a digital filter evaluation and simulation tool for MOSVLSI filter implementations", IEEE J. Solid-State Circuits, Vol. SC-19, No 3, pp 414-424, June 1984.

[Sne86] W. M. Snelgrove, A. S. Sedra, "Synthesis and analysis of state-space active filters using intermediate transfer functions", IEEE Trans. Circuits and Syst., Vol. CAS-33, No 3, pp 287-301, March 1986.

Books

[Sch69] H. Schwarz, "Einführung in die moderne Systemtheorie", Friedr. Vieweg & Sohn GmbH Verlag, Braunachwieg, Germany, 1969.

[Jac87] L. B. Jackson, "Digital filters and signal processing", Kluwer academic publishers, Boston, USA, 1987.

5 Realization

Publication

[Ogu74] H. Oguey, E. Vittoz, B. Gerber, "Diviseurs de fréquence ultra-rapides", IX congrès international de chronométrie, vol. C1, Stuttgart, pp 629-633, 1974.

Books

[Mea80] C. Mead, L. Conway, "Introduction to VLSI systems", Addison-Wesley Publishing Company, 1980.

[Bot89] J. Botelho, "Réalisation et implantation VLSI des filtres numériques quasi-continus", travail de diplôme, Institut de Microtechnique, Université de Neuchâtel, 1989.

6 Design example

Publications

[Cla88] L. Claesen, F. Catthoor, D. Lanneer, G. Gooseens, S. Note, J. Van Meerbergen, H. De Man, "Automatic synthesis of signal processing benchmark using the CATHEDRAL silicon compilers", IEEE 1988 custom integrated circuits conference, pp 14.7.1-14.7.4, 1988.

[Ans90] M. Anaorga, U. Sjöström, I. Defilippis, P. Balaigar, F. Pellandini : "A flexible low-power digital signal processor based on a contant-addressable memory", EUSIPCO'90, Barcelona, Spain, pp. 1399-1402, September 1990.

Books

[Cor91a] F. Corthay, "An open tool for ODLF design", Univarsity Neuchâtel, report 297 PE 09/91, September 1991.

[Cor91b] F. Corthay, "ODLF simulation using LabVIEW", University Neuchâtel, report 298 PE 09/91, September 1991.

Annex A
Transfer function deviation

The table provided in this annex is the list of the calculated and measured values used to build the plot of figure 6.7.

Table A.1 displays the values building the plots of figure 6.7. The first column, named "freq", lists the frequencies at which the transfer functions have been evaluated and measured. Column 2, "H(s)", is the evaluated transfer function of the analog continuous time prototype. Column 3, "H(z)", is the evaluated transfer function of the discrete time oversampled filter. Column 4, "+ dev", and column 5, "- dev", are the expected transfer function deviation for internal signals coded on 9 significant bits where all error sources contribute respectively to an increase or a decrease of the amplitude of the transfer function. Column 6, "+ measure", and column 7, "- measure", are respectively the maximal and the minimal measures of the transfer function amplitude.

freq	H(s)	H(z)	+ dev	- dev	+ measure	- measure
2	-0.0440	-0.0468	0.0040	-0.0921	0.0360	-0.0900
2.5	-0.0054	-0.0110	0.0657	-0.0771	0.1070	-0.0920
3	-0.0491	-0.0613	0.0630	-0.1644	0.0980	-0.0790
3.2	-0.5149	-0.5347	-0.3941	-0.6523	-0.2510	-0.6340
3.5	-2.9505	-2.9770	-2.8235	-3.1414	-2.4710	-3.2160
3.7	-8.5639	-8.5844	-8.3612	-8.8604	-8.1750	-8.8100
4	-15.9088	-15.9254	-15.5444	-16.4261	-15.1570	-17.3320
4.2	-24.4178	-24.4307	-23.5706	-25.5467	-23.4520	-24.7320
4.5	-34.1289	-36.1394	-33.2086	-41.0758	-31.6270	-40.8860
4	-44.4494	-44.4580	-38.2304	-120.0000	-35.3590	-45.8300
5	-40.0238	-40.0322	-35.7641	-48.5747	-34.7370	-40.4670
6	-50.2612	-50.2673	-40.9870	-120.0000	-37.8760	-58.5430
7	-41.4627	-41.4687	-37.0161	-50.8448	-36.7300	-45.0350
8	-36.2099	-36.2154	-33.6006	-39.9199	-33.2170	-41.4490
9	-34.3378	-34.3431	-32.2396	-37.0948	-31.7510	-38.7440
10	-33.5727	-33.5778	-31.6798	-35.9901	-31.4390	-34.5540
11	-33.3074	-33.3124	-31.5028	-35.5862	-31.5680	-35.6670
12	-33.3011	-33.3060	-31.5284	-35.5321	-31.3820	-35.8400
13	-33.4392	-33.4440	-31.4670	-35.6714	-31.5460	-35.4660
14	-33.6610	-33.6658	-31.8703	-35.9243	-31.6150	-35.8330
15	-33.9320	-33.9368	-32.1103	-36.2459	-32.3190	-37.6320
16	-34.2314	-34.2362	-32.3705	-36.6099	-31.9600	-37.2960
17	-34.5465	-34.5512	-32.6403	-36.9998	-32.5510	-36.2550
18	-34.8690	-34.8737	-32.9134	-37.4053	-33.4540	-37.4330
19	-35.1937	-35.1984	-33.1854	-37.8194	-33.4870	-38.2520
20	-35.5171	-35.5217	-33.4539	-38.2374	-34.3620	-34.4430
25	-37.0585	-37.0431	-34.7016	-40.3147	-36.3670	-37.7040
30	-38.4330	-38.4375	-35.7710	-42.3036	-37.8560	-39.9130
40	-40.7280	-40.7325	-37.4620	-46.0392	-39.8400	-41.6330
50	-42.5732	-42.5777	-38.7295	-49.6581	-41.0380	-44.6070
75	-44.0041	-46.0086	-40.8470	-60.5110	-43.4420	-49.3480
100	-48.4712	-48.4757	-42.1654	-120.0000	-44.6600	-58.8550
200	-54.4614	-54.4655	-44.6473	-120.0000	-50.6630	-86.5540
300	-57.9776	-57.9813	-45.6679	-120.0000	-50.1270	-80.8160
500	-62.4117	-62.4139	-46.5820	-120.0000	-53.7260	-70.2580
1000	-68.4311	-68.4262	-47.3373	-120.0000	-55.9440	-82.0790

table A.1 Frequency response of a fifth order ODLF.