

# OPTIMIZED REAL TIME IMPLEMENTATION OF SPECTRAL ANALYSIS AND QUANTIZATION FOR THE CELP FS1016 SPEECH CODER

*S. Grassi, M. Ansorge, and F. Pellandini*

Institute of Microtechnology

University of Neuchâtel, Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland  
Tel.: +41 32 718 3432, Fax: +41 32 718 3402, E-mail: sara.grassi@imt.unine.ch

## ABSTRACT

The optimized implementation of the CELP FS1016 spectral analysis and quantization block on a DSP56001 is presented in this paper. This implementation can be seen as a preparation for an optimal low power, small size, custom VLSI implementation. The optimization is based on careful algorithmic optimization and study of the fixed-point quantization effects. Algorithmic optimization deals with the choice and modification of the algorithms, their optimal interrelation in the whole system, and a good match between algorithms and architecture. Using algorithmic optimization, a saving of 66 % was obtained in the DSP56001 implementation of the LSP calculation functional block.

## 1. INTRODUCTION

The purpose of a speech coder is to compress speech signals, employing as few bits as possible in their digital representation. The term "speech coding" is commonly used to refer to the coding of telephone bandwidth speech (300-3400 Hz) sampled at 8 kHz.

Typical applications of speech coding are in telecommunications, voice storage systems, personal communications systems, and multimedia for personal computing, where voice storage is becoming a standard feature. All these applications require real time implementation. Additionally, application in portable devices such as digital cellular telephones, vocal pagers, and portable multimedia terminals and computers, require low power consumption and small size.

The implementation of a speech coding algorithm can be done either on a commercial DSP, or on custom hardware (ASIC). In both cases, the choice of a fixed-point arithmetic is a key point to decrease cost, size and power consumption. Therefore, the analysis of the fixed-

point quantization effects is of great importance in carefully optimized implementations.

Optimization at the algorithmic level (algorithm choice and simplification) is the key for a low power implementation as it allows savings of orders of magnitude in power consumption. Another aspect is the determination of the optimum scaling and the minimum wordlength needed at every node of the algorithm.

The U.S. Federal Standard 1016, is a Code-Excited Linear Predictive (CELP) speech coder operating at 4.8 kbps [1]. This coder was intended originally for secure voice transmission, but it is also suitable for storage applications. The computational complexity of this coder (10-20 MIPS) makes it a challenge for a low power VLSI implementation.

In this paper, the optimized implementation on a DSP56001 of the CELP FS1016 spectral analysis and quantization block is presented. This implementation can be seen as a preparation for an optimal low power, small size custom VLSI implementation.

This paper is organized as follows. In Section 2, it is shown how spectral analysis and quantization is done in the CELP FS1016 speech coder. This coder uses an LSP representation of LPC parameters, which is defined in Section 3. Then, the calculation of LSP parameters from LPC coefficients is explained in Section 4, while the inverse transformation is explained in Section 5. The interpolation and quantization of LSP parameters is discussed in Section 6.

In Section 7, it is explained how the fixed-point quantization effects were evaluated, to determine the optimum scaling and minimum wordlength needed at every node of the algorithm. The result of this study was used in the DSP56001 implementation presented in Section 8. Finally, conclusions and a discussion on further work are given in Section 9.

## 2. THE CELP FS1016 SPEECH CODER

The CELP FS1016 uses a 30 ms frame size, with four subframes of 7.5 ms each. Spectral analysis is performed by linear predictive coding (LPC) analysis, once per frame and the obtained LPC coefficients are quantized and transmitted. These coefficients are interpolated and used for the optimum codevector search, which is done once per subframe [1].

The detailed diagram of the spectral analysis and quantization block is given in Figure 1 and it is explained next.

### 2.1. Spectral Analysis and Quantization

LPC analysis is performed by 10-th order autocorrelation analysis (Levinson-Durbin recursion) using a 30 ms Hamming window (centered at the end of the previous frame), no pre-emphasis, and 15 Hz bandwidth expansion [1]. The bandwidth expanded LPC coefficients are converted to a set of LSP parameters and quantized using a 34-bit, independent, nonuniform scalar quantization. Then, two adjacent quantized sets of LSP parameters are linearly interpolated, obtaining four sets of interpolated LSP parameters, one set per subframe. Each of these LSP sets is converted back to LPC, to be used in the LPC synthesis filter for the optimum codevector search.

## 3. LPC ANALYSIS AND LSP REPRESENTATION

Linear predictive coding (LPC) is an accurate and economic representation of the speech spectral envelope which is widely used in speech coding as well as in other speech processing areas such as speech synthesis and voice recognition.

Line spectrum pair (LSP) parameters have a one to one correspondence with the LPC coefficients and allow more efficient encoding of the spectral information.

LSP representation of 10-th order LPC coefficients is used in nearly all speech coder standards, with bit rates of less than 16 kbps [2]. Additionally, nearly all the CELP coders found in recent publications use LSP representation of 10-th order LPC. Hereafter, an LPC order of 10 is assumed.

### 3.1. Definition of LSP Parameters

The starting point for deriving the LSP parameters is the 10-th order LPC analysis filter:

$$A_{10}(z) = 1 + \sum_{k=1}^{10} a_k \cdot z^{-k} .$$

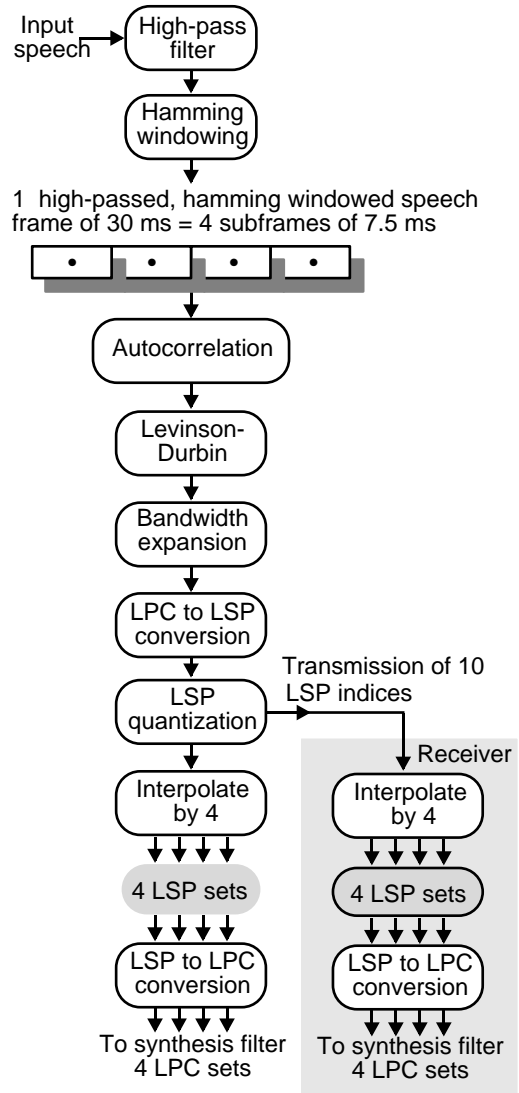


Figure 1: Short-term spectral analysis in the CELP FS1016 speech coder.

Two polynomials,  $P_{10}(z)$  and  $Q_{10}(z)$ , are formed by [3]:

$$\begin{aligned} P_{10}(z) &= A_{10}(z) + z^{-11} \cdot A_{10}(z^{-1}) \\ Q_{10}(z) &= A_{10}(z) - z^{-11} \cdot A_{10}(z^{-1}) . \end{aligned}$$

These polynomials have each a trivial zero at  $\pm 1$ , which is removed, obtaining:

$$\begin{aligned} P'_{10}(z) &= P_{10}(z) / (1 + z^{-1}) \\ Q'_{10}(z) &= Q_{10}(z) / (1 - z^{-1}) . \end{aligned}$$

$P'_{10}(z)$  and  $Q'_{10}(z)$  are symmetrical, have real coefficients, and their zeros are on the unit circle and interlaced [3]. The angles of their zeros (upper semicircle of the  $z$ -plane only) are the 10 LSP parameters, denoted as  $\omega_i$ . These LSP parameters completely specify  $P'_{10}(z)$  and  $Q'_{10}(z)$ .

#### 4. CALCULATION OF THE LSP PARAMETERS

The calculation of LSP parameters from LPC coefficients is a computationally intensive task, as it involves the resolution of polynomials by numerical root search. A survey of existing algorithms for LSP calculation was done [2] and it was found that Kabal's [3], Saoudi's [4] and Chan's [5] algorithms were the most promising for efficient real time implementation. We also proposed two novel efficient algorithms referred to as "Mixed-LSP" [6] and "Quantized-search Kabal" [7].

These two proposed LSP calculation algorithms were compared with the algorithms of Kabal, Chan and Saoudi from the point of view of accuracy (see § 4.4), reliability and computational complexity [2]. It was found that, among the existing methods, Kabal's algorithm is the most efficient and suitable for application in the CELP FS1016. This algorithm, as well as "Mixed-LSP" and "Quantized-search Kabal", were used in the DSP56001 implementation (see § 8), and are briefly explained in the next sub-sections.

##### 4.1. Kabal's Algorithm

The 5-th order polynomials  $P'_{10}(x)$  and  $Q'_{10}(x)$  are obtained by evaluating  $P'_{10}(z)$  and  $Q'_{10}(z)$  on the unit circle ( $z = e^{j\omega}$ , where  $\omega$  is the angular frequency, with  $0 \leq \omega \leq 2\pi$ ), and using the mapping  $x = \cos(\omega)$ . The roots of  $P'_{10}(x)$  and  $Q'_{10}(x)$  are the LSPs in the "x-domain", denoted as  $x_i$ , with  $x_i = \cos(\omega_i)$ . In the numerical solution proposed by Kabal in [3], the zero crossings are searched starting at  $x = +1$ , with decrements of  $\Delta = 0.02$ . Once a zero crossing is found, its position is refined by four successive bisections and a final linear interpolation. A maximum of 150 polynomial evaluations is needed. An efficient polynomial evaluation requiring only 4 multiplications and 9 additions is also proposed in [3].

##### 4.2. Mixed-LSP

The precision of the LSPs obtained with Kabal's method is higher than required by speech coding applications, but the number of bisections cannot be decreased, or the size of the grid increased, without compromising the zero crossing search.

In [6], it is shown that five intervals, containing each only one zero crossing of  $P'_{10}(x)$  and one zero crossing of  $Q'_{10}(x)$ , can be calculated, as the roots of a 4-th order polynomial,  $D_{10}(x)$ . This fact is used to avoid the zero crossing

search, allowing a trade-off between LSP precision and computational complexity. The resulting algorithm is called Mixed-LSP algorithm, and needs a total of 60 polynomial evaluations. These evaluations are done using Kabal's efficient polynomial evaluation.

The calculation and sorting of the roots of  $D_{10}(x)$  was carefully optimized [6], and finally needs the following operations: 20 multiplications, 34 add/sub, 2 divisions and 5 square roots, as well as 3 comparison/swapping operations.

It was found that Mixed-LSP algorithm needs 33 % less MIPS than Kabal's algorithm on a DSP56001 implementation (see § 8).

The proposed Mixed-LSP algorithm is computationally less expensive but also less accurate than Kabal's method (see § 4.4). On the other hand, the accuracy of the Mixed-LSP algorithm is sufficient for speech coding applications using the 34-bit quantizer of the CELP FS1016 [2].

The Mixed-LSP algorithm can be used not only with the scalar quantization of the CELP FS1016, but also with vector quantization or predictive quantization.

##### 4.3. Quantized-search Kabal

The LSPs obtained with Mixed-LSP or Kabal's algorithm are not quantized. If these methods are used in the CELP FS1016, the LSPs are first calculated, and then quantized using the 34-bit non-uniform scalar quantization [1]. The algorithm denoted as "Quantized-search Kabal" can be seen as a modified version of Kabal's algorithm, in which the zero crossings search is done on a grid formed with the values of the quantization tables [7].

As the actual LSPs are not calculated, two new criteria to select the quantized LSPs which are closer to the actual LSPs are proposed [7]. These criteria take into account the interaction between successive LSPs.

The efficiency and reliability of this algorithm are improved by using the interlacing property of the LSPs and knowledge of the direction of the sign-change at every zero-crossing.

The quantization performance of "Quantized-search Kabal" is very close to the performance of Kabal's algorithm followed by quantization (see § 4.4). The maximum number of polynomial evaluations is reduced to 71, resulting in a saving of 66 % of the MIPS on a DSP56001 implementation (see § 8).

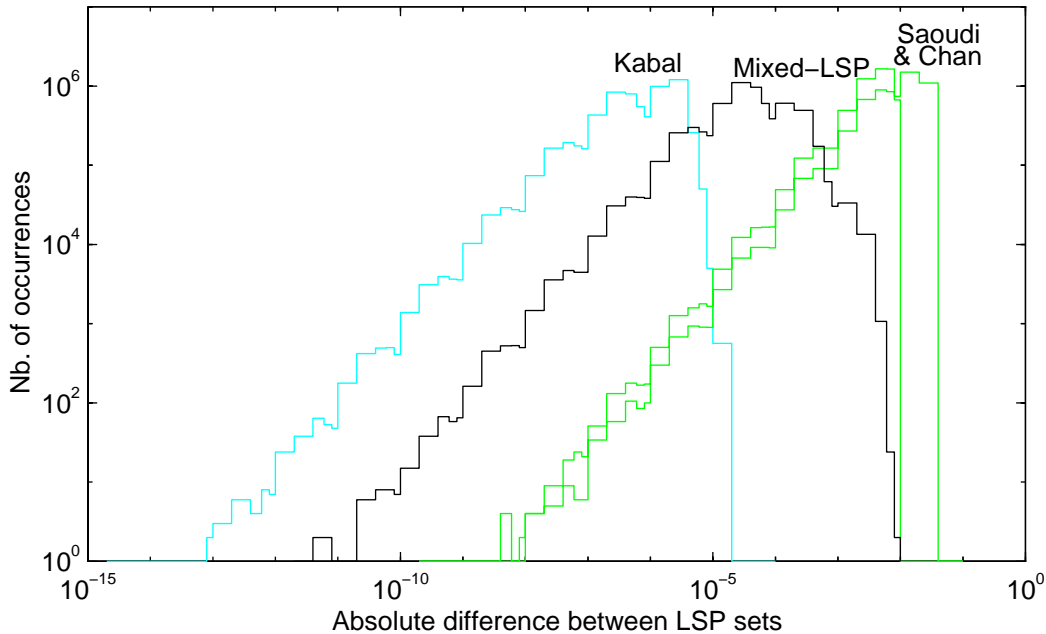


Figure 2: Histograms of the absolute difference between LSP sets calculated with high accuracy on one side, and Kabal's, Chan's, Saoudi's and Mixed-LSP algorithms on the other side

#### 4.4. Experimental Evaluation

The accuracy of Kabal's, Saoudi's, Chan's and Mixed-LSP algorithms was evaluated using the whole TIMIT database [2]. For every speech file, two sets of LSP vectors were compared, one set calculated with the algorithm under evaluation, and the other set calculated with a high accuracy method ( $\epsilon < 10^{-16}$ ).

The histograms of the absolute differences found for every algorithm under test are given in Figure 2. Note that the LSPs calculated with Chan's algorithm were converted from the " $\omega$ -domain" to the " $x$ -domain" in order to make a meaningful comparison. It is observed that Kabal's is the most accurate among the algorithms under evaluation, followed by Mixed-LSP and then Saoudi's and Chan's.

#### 4.5. Spectral Distortion

Kabal's, Saoudi's, Chan's and Mixed-LSP algorithms, as well as the high accuracy method were used to calculate the LSPs, which were then quantized with the 34-bit scalar quantizer. Spectral distortion was measured in all cases. The resulting average spectral distortion and percentage of outliers (with spectral distortion between 2-4 dB, and greater than 4 dB) are given in Table 1, together with the spectral distortion measured for the "Quantized-search Kabal" algorithm.

The results obtained using Kabal's, Mixed-LSP and "Quantized-search Kabal" algorithms are very close to those obtained with the high accuracy method. Thus, although the Mixed-LSP method is less accurate than Kabal's method, it is sufficient for speech coding applications using the 34-bit scalar quantizer of the CELP FS1016. It is also observed that the "Quantized-search Kabal" algorithm can be used to speed up the calculation and quantization processes, without great degradation of the quantization performance. On the other hand, the quantization performance is degraded when the algorithms of Saoudi and Chan are used for LSP calculation. This is due to the inaccuracy of these algorithms, observed in Figure 2. The accuracy of these algorithms can be improved, but at the cost of increased complexity [2].

Algorithm	Spectral Distortion (dB)		
	average	% 2-4	% >4
High accuracy	1.5329	12.3450	0.1888
Kabal	1.5329	12.3453	0.1888
Mixed-LSP	1.5331	12.3631	0.1885
Q.-s. Kabal	1.5330	12.3501	0.1895
Saoudi	1.6536	19.1166	0.2025
Chan	1.7273	24.4656	0.2265

Table 1: Comparison among different methods to calculate quantized LSPs, in terms of spectral distortion.

## 5. LSP TO LPC TRANSFORMATION

The LPC analysis filter can be expressed as:

$$A_{10}(z) = \frac{P_{10}(z) + Q_{10}(z)}{2}$$

where  $P_{10}(z)$  and  $Q_{10}(z)$  can be obtained from the LSP parameters  $\{\omega_i\}$  [2].

Three existing algorithms for LSP to LPC transformation were considered [2]. These are the Direct Expansion algorithm, the LPC Analysis Filter algorithm, and Kabal's algorithm. The number of operations required by these three algorithms is given in Table 2, showing that Kabal's is the least expensive. Besides, this algorithm is highly regular and numerically stable [3], which is advantageous for efficient implementation.

It is also important to notice that these transformation algorithms require the LSPs to be in the "x-domain", with  $x_i = \cos(\omega_i)$ .

<i>LSP to LPC Algorithm</i>	<i>MULT</i>	<i>ADD</i>
Direct Expansion	62	92
LPC Analysis Filter	30	70
Kabal's algorithm	20	59

Table 2: Number of operations required by three different LSP to LPC transformation algorithms.

## 6. LSP QUANTIZATION AND INTERPOLATION IN THE "X-DOMAIN"

In the CELP FS1016, the LSP coefficients are quantized and interpolated in the angular frequency domain " $\omega$ -domain" [1]. As the LSPs obtained with the methods of Kabal, Saoudi and Mixed-LSP are in the "x-domain", it is desirable to perform the quantization and interpolation in this domain. This avoids the computationally expensive conversion from the "x-domain" to the " $\omega$ -domain" for quantization and interpolation, and then from the " $\omega$ -domain" to "x-domain" for LSP to LPC transformation. Quantization and interpolation performed in the " $\omega$ -domain" and in the "x-domain" were evaluated using spectral distortion measure on the whole TIMIT database [2] and it was observed that their performance is equivalent.

## 7. QUANTIZATION EFFECTS

The study of the fixed-point quantization effects was done, using the method described in [8], to determine the optimum scaling and minimum

wordlength needed at every node of the algorithm. The algorithms were then modified to include the scaling, and to fit their dynamic range needs into the dynamic range available in the DSP56001 registers. This is done by using normalization and denormalization steps at some localized nodes of the algorithms [2] which have higher dynamic range needs.

All the functional blocks of the CELP FS1016 spectral analysis and quantization were coded in C language (using double-precision floating-point arithmetic) and interfaced as Matlab functions. These programs are used to characterize the "infinite precision" behavior of each block and are the "reference system" to evaluate the degradation in performance in the case of a fixed-point implementation.

Each functional block was then implemented on a DSP56001 in assembly language and on a workstation in C language, including the DSP56001 arithmetic effects [2]. This C program is thus a "model" of the corresponding DSP56001 implementation. Each of these C "models" is also interfaced as a Matlab function.

It was verified that each DSP56001 implementation and its corresponding C model have exactly the same output under the same input. This verification was carried out using the whole TIMIT database. After that, the C model is used to measure the performance of the DSP56001 implementation.

The advantage of this approach is that the C models are easily interfaced (within Matlab) with the rest of the system. The C models can also be used to try out different implementation options before actually implementing them on the DSP56001.

## 8. DSP56001 IMPLEMENTATION

The different functional blocks that constitute the CELP FS1016 spectral analysis and quantization were implemented on a DSP56001. LSP calculation was implemented using Kabal's, Mixed-LSP, and "Quantized-search Kabal" algorithms.

The parallelism of the DSP56001 is exploited, trying to perform as much data transfers as possible in parallel to the arithmetic instructions. The resulting computational complexity for each functional block is given in Table 3.

The total computational complexity for a DSP56001 implementation of the CELP FS1016 spectral analysis and quantization is given in Table 4. Three variants are shown, depending on the LSP calculation method used.

<i>Functional Block</i>	<i># Cycles</i>	<i>MIPS</i>	
High-pass filter	3390	0.0565	
Windowing	984	0.0164	
Autocorrelation	6158	0.1026	
Levinson-Durbin	1592	0.0265	
BW Expansion	52	0.0009	
LSP	Kabal	10540	0.1757
Calcu- lation	Mixed-LSP	6986	0.1164
	Q.-S. Kabal	4262	0.0710
LSP Quantization	2168	0.0361	
Interpolation	236	0.0039	
(4x) LSP to LPC	906	0.0151	

Table 3: Complexity, for a DSP56001 implementation, of each functional block of the CELP FS1016 spectral analysis and quantization.

<i>LSP calculation method</i>	<i># Cycles</i>	<i>MIPS</i>
Kabal	26026	0.4338
Mixed-LSP	22472	0.3745
"Quantized-search Kabal"	17580	0.2930

Table 4: Total complexity for a DSP56001 implementation of the CELP FS1016 spectral analysis and quantization block.

## 9. CONCLUSIONS AND FURTHER WORK

The optimized implementation of the CELP FS1016 spectral analysis and quantization on a DSP56001 was presented. The optimization was based on careful algorithmic optimization and the study of the fixed-point quantization effects. Algorithmic optimization deals with the choice and modification of the algorithms, as well as their optimal interrelation in the whole system.

The optimal implementation on a fixed-point commercial DSP such as the DSP56001 can be seen as a preparation for an optimal low power, small size, custom VLSI implementation. In Table 4, it is observed that the most efficient implementation is the one using "Quantized-search Kabal" for LSP calculation. This implementation should be chosen for the VLSI implementation.

The next step is to use the quantized C models to minimize the wordlength needed, while keeping an acceptable performance. Preliminary work suggests that a word-length of 16 to 20 is needed as input of the multiplier and storage.

For possible future VLSI implementation, we propose the use of an architecture similar to the one of the DSP56001, with a bit-parallel MAC and separate X and Y data memories, buses and address generation units. The accumulator of the

MAC should have 32 to 40 bits, plus 8 bits extension. Only the subset of instructions used in the algorithms need to be implemented. As the sequencing of DSP56001 instructions was carefully optimized, it can be directly used for the VLSI implementation.

## 10. ACKNOWLEDGEMENTS

The financial support provided by the Swiss Federal Office for Education and Science under Grant OFES C97.0050 (COST 254 project) is acknowledged.

## 11. REFERENCES

- [1] "Federal Standard 1016, Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bit/second Code Excited Linear Prediction (CELP)", National Communications Systems, Office of Technology and Standards, Washington, DC20305-2010, 1991.
- [2] S. Grassi, "Optimized Implementation of Speech Processing Algorithms", Ph.D. Thesis, IMT, University of Neuchâtel, Switzerland, Feb. 1998.
- [3] P. Kabal and P. Ramachadran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials", IEEE Trans. ASSP, Vol. 34, No. 6, pp. 1419-1426, Dec. 1986.
- [4] S. Saoudi and J. Boucher, "A New Efficient Algorithm to Compute the LSP Parameters for Speech Coding", Signal Processing, Vol. 28, No. 2, pp. 201-212, Elsevier, 1992.
- [5] C. Chan and K. Law, "An Algorithm for Computing the LSP Frequencies Directly from the Reflection Coefficients", Proc. EUROSPEECH'91, pp. 913-916, 1991.
- [6] S. Grassi, A. Dufaux, M. Ansorge, and F. Pellandini, "Efficient Algorithm to Compute LSP Parameters from 10-th order LPC Coefficients", Proc. ICASSP'97, Vol. 3, pp. 1707-1710, 1997.
- [7] S. Grassi, M. Ansorge, and F. Pellandini, "Fast LSP Calculation and Quantization with Application to the CELP FS1016 Speech Coder", to be presented at EUSIPCO'98, Rhodes, Greece, Sept. 8-11, 1998.
- [8] S. Grassi, A. Heubi, M. Ansorge, and F. Pellandini, "A Study of a VLSI Implementation of a Noise Reduction Algorithm for Digital Hearing Aids", Proc. EUSIPCO'94, Vol. 3, pp. 1661-1664, 1994.