A LOW-POWER VLSI FEATURE EXTRACTOR FOR SPEECH RECOGNITION

M. Felici, M. Borgatti, A. Ferrari and R. Guerrieri

DEIS, University of Bologna
Viale Risorgimento, 2
40136 - Bologna, Italy
Central R&D, SGS-Thomson Microelectronics
Via Olivetti, 2
20041 - Agrate Brianza, Italy

ABSTRACT

A low-power feature extraction chip computing cepstral coefficients from linear predictive analysis on one-bit quantized speech signal is presented and its VLSI implementation is evaluated. An isolated-word small-vocabulary speech recognizer based on these features has been developed. Its recognition accuracy is within 2% below a system based on standard linear predictive cepstral features. The power consumption of the feature extractor chip is 30μW at 0.9V.

1. INTRODUCTION

Low-cost VLSI speech recognition systems have several promising applications in large volume products such as personal digital assistants and communicators (PDAs, PDCs) and toys. These systems must feature very low-power electronics, single battery-sized supply and simple analog-to-digital conversion.

Previous works [1, 2, 3] show that a drastic one-bit quantization of the speech signal does not severely affect its intelligibility and discrimination power in isolated-word recognition tasks. In particular [2, 3] present recognition systems proving that the recognition accuracy has a small degradation when the set of features is computed from the one-bit quantized (1BQ) speech through linear-predictive (LP) analysis. Hence, reduced computational complexity and simplified analog circuitry can be used, and voltage scaling techniques can be effectively applied to the digital recognition system to drop power consumption.

In this paper we present a low-power, low-voltage VLSI feature extractor based on one-bit quantized speech that computes a set of variance-weighted cepstral coefficients for recognition. This system avoids the full analog-to-digital conversion of the speech signal allowing the reduction of the analog circuitry to simple filtering.

The proposed speech preprocessor is part of a complete speech recognizer to be implemented in a single chip. This block has been realized as a stand-alone block to evaluate the impact of aggressive voltage scaling techniques on this kind of applications.

The goal of low-power consumption is pursued starting from the algorithmic down to the electrical level. Suitable algorithms should feature simple arithmetics and should achieve efficient parallelization of the computation so that an architectural voltage scaling approach can be exploited [4].

In the following sections the main algorithmic and architectural choices are described.

2. ALGORITHM

Optimal LP filter computation requires the evaluation of the autocorrelation function (ACF). Short-term p-th order autocorrelation of the speech signal $s(i)$ can be computed as follows:

$$r_k = \frac{1}{N}\sum_{i=1}^{N} s(i)w(i)s(i+k)w(i+k) \qquad k = 0,\ldots,p$$

where $w(i)$ is a function that is zero out of the N-samples window under examination. In case of 1BQ speech (i.e. $s(i) \in \{-1,1\}$) and rectangular windowing function, the k-th autocorrelation coefficient can be simply computed by counting how many sign changes occur between samples belonging to the speech window at distance k, in fact:

$$\begin{aligned} r_k &= \frac{1}{N}\sum_{i=1}^{N-k} s(i)s(i+k) = \\ &= \frac{1}{N}\sum_{i=1}^{N-k}[1-2s_b(i)\oplus s_b(i+k)] = \\ &= \frac{N-k-2\sum_{i=1}^{N-k} s_b(i)\oplus s_b(i+k)}{N} \end{aligned}$$

where the multiplication is replaced by a logic function XOR between the samples coded at 1 bit (values $-1, 1$ of $s(i)$ are mapped to the binary values $0, 1$ respectively for $s_b(i)$). In [3] it is shown that r_k may be estimated as:

$$r_k \cong \frac{N-2\sum_{i=1}^{N} s_b(i)\oplus s_b(i+k)}{N} \qquad (1)$$

This formulation is useful for hardware implementation when the analysis is performed on overlapping windows and the frame duration is an integer divider of the window duration. In our system window duration is 32ms, frame duration is 8ms and speech signal is sampled at 8kHz. In this case a relevant reduction of computation can be achieved splitting the counting $\sum_{i=1}^{N} s_b(i)\oplus s_b(i+k)$ on single frames rather than counting over the whole window. In this way r_k is obtained as sum of frame-based counts that can be reused in the analysis of the next overlapping windows.

The Levinson-Durbin (LD) recursion is used to compute the LP coefficients from the ACF. This algorithm relies on the positive

definiteness of the autocorrelation matrix which is a sufficient condition for the consistence of results [5].
The use of approximation (1) requires a correction of the autocorrelation matrix to guarantee its positive definiteness. A way [6] to obtain this result is to increase the 0-*th* order autocorrelation coefficient r_0 from 1 to $(1+\lambda)$, $\lambda > 0$ (typical value for λ are in $[0.1, 0.5]$).
Spectra in Fig.1 give an idea of the information retained by 1BQ speech. The LP-smoothed spectra of a 12BQ and 1BQ speech segment are compared. 12-th order LP analysis with Hamming windowing and 16-th order LP analysis using approximation (1) have been used respectively. The plots show that the peaks in LP

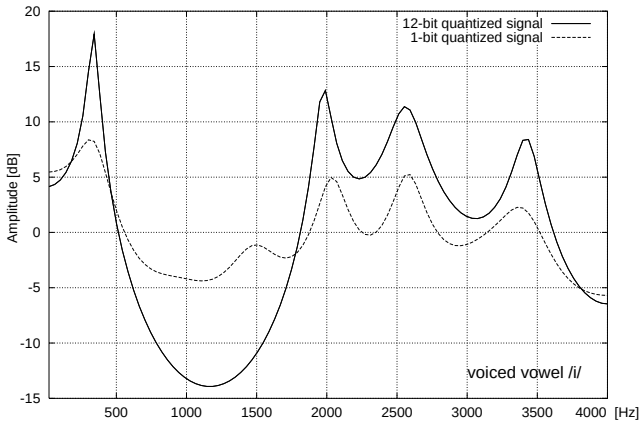

Figure 1: Comparison of LP inverse filter spectra

filter spectrum (formant of speech) are preserved after 1BQ of the speech signal.

The Levinson-Durbin iteration may be summarized as follow:

$$\begin{aligned} k_m &= -\beta_m/\alpha_m \qquad (2)\\ a_{m+1,l} &= \begin{cases} a_{m,l} + k_m a_{m,m+1-l} & l = 1,\cdots,m \\ k_m & l = m+1 \end{cases} \\ \alpha_{m+1} &= \alpha_m + k_m\beta_m \\ \beta_{m+1} &= r_{m+2} + \sum_{i=1}^{m+1} a_{m+1,i}\, r_{m+2-i} \end{aligned}$$

with initial conditions $\beta_0 = r_1,\ \alpha_0 = r_0 = 1+\lambda$. It computes all optimal LP *m-th* order filter coefficients ($a_{m,1},\cdots,a_{m,m}$), with $m = 0,\cdots,p$.

The fixed point implementation of the algorithm requires scaling to the allowed integer range of the involved variables. Let us assume that this range is mapped to the interval $[-1, 1[$.

It can be shown[6] that $k_m \in]-1, 1[$ and from eq. (2) we have $|\beta_m| < \alpha_m$ so that correct scaling of α_m implies correct scaling of β_m. The algorithm bounds α in $]\lambda, 1+\lambda]$ and it is found that $a_{i,j} \in]-4, 4[$.

By reformulating the iteration using $\overline{r}_k = r_k/2$ and $\overline{a}_{i,j} = a_{i,j}/4$ correct scaling of all variables may be obtained since this scaling bounds α_m in $]\lambda/2, (1+\lambda)/2]$. This allow the use of $\overline{\alpha}_m = 2\alpha_m - \lambda \in [0, 1]$ instead of α_m in the iteration, achieving the exploitation of all bits used for its representation.

In the following the notation $\ll n$ and $\gg n$ is used to indicate respectively n left-shifts and n right-shifts. The multiplication of two n bit integers numbers generates a 2n bit number. To obtain the n bit two's complement integer representation of the result it is necessary to shift right n-1 times the result, that is right bits have to be selected. The operator $\times$ is used for this kind of multiplication with the meaning $a \times b = (ab) \gg (n-1)$. The fixed point formulation of the Levinson-Durbin iteration is:

$$\begin{aligned} k_m &= -\beta_m \times \frac{2}{\overline{\alpha}_m + \lambda} \\ \overline{a}_{m+1,l} &= \begin{cases} \overline{a}_{m,l} + k_m \times \overline{a}_{m,m+1-l} & l = 1,\cdots,m \\ k_m \gg 2 & l = m+1 \end{cases} \\ \overline{\alpha}_{m+1} &= \overline{\alpha}_m + (k_m \times \beta_m) \ll 1 \\ \beta_{m+1} &= \overline{r}_{m+2} + [\sum_{i=1}^{m+1} \overline{a}_{m+1,i} \times \overline{r}_{m+2-i}] \ll 2 \end{aligned}$$

with initial conditions $\beta_0 = \overline{r}_1,\ \overline{\alpha}_0 = 1$. The evaluation of the function $f(\overline{\alpha}) = \frac{2}{\overline{\alpha}_m + \lambda}$ should be implemented in a specialized hardware unit so that no divisions are required.

Cepstral coefficients can be computed directly from the LP model. If we call c_i the *i-th* cepstral coefficient and a_i the *i-th* LP coefficient, the iteration is:

$$\begin{aligned} c_1 &= -a_1 \\ c_i &= -a_i - \sum_{j=1}^{i-1}(1 - \frac{j}{i}) a_j c_{i-j} \quad i = 2,\cdots,N_{cep} \end{aligned}$$

Multiplying the second relation by i and exploiting the position $\xi_i = -i\, c_i$, the iteration may be rewritten:

$$\begin{aligned} \xi_1 &= a_1 \\ \xi_i &= i\, a_i + \sum_{j=1}^{i-1} a_j \xi_{i-j} \quad i = 2,\cdots,N_{cep} \end{aligned}$$

achieving a substantial saving in the number of multiplications. For 15 cepstral coefficients this formulation reduces the number of multiplications to 57% of those in the trivial implementation. Since the chosen set of features is cepstral coefficients normalized according to the inverse of their standard deviation, the division by $-i$ (required to obtain c_i from ξ_i) can be absorbed in the normalization operation. It is found that $c_i \in]-4, 4[$ while $\xi_i \in]-16, 16[$, so scaling starting from scaled a_i version is straightforward too, Calling $\overline{\xi}_i = \xi_i/16$ the scaled version of ξ_i, the fixed point implementation of the algorithm is:

$$\begin{aligned} \overline{\xi}_1 &= \overline{a}_1 \gg 2 \\ \overline{\xi}_i &= i \times \overline{a}_i \gg 2 + [\sum_{j=1}^{i-1} \overline{a}_j \times \overline{\xi}_{i-j}] \ll 2 \quad i = 2,\cdots,N_{cep} \end{aligned}$$

Finally scaled cepstral features $\overline{c}_i = c_i/4$ may be obtained as follow:

$$\overline{c}_i = [(\frac{1}{i\sigma_i}) \times \overline{\xi}_i] \ll 2$$

where $\frac{1}{i\sigma_i}$ are fixed weights (σ_i are evaluated analyzing speech coming from different speakers in various environmental conditions) that can be precomputed.

With the above positions the LP-Cepstrum computation can be efficiently performed on a fixed-point system featuring an arithmetic unit with integer multiplication, sum and shift, and a specialized unit for the evaluation of $f(\overline{\alpha}) = \frac{2}{\overline{\alpha}_m + \lambda}$.

The word length has a great impact on power consumption especially when multiplications are involved. In this case the switching activity depends quadratically upon word length. To determine minimum word length, the deviation of fixed point cepstral coefficients from floating point ones versus word length has been evaluated. If cepstrum are normalized upon their standard deviation and scaled in order to fit the allowed range, test shows that only the five most significant bits of cepstral coefficients are needed for recognition purposes. The value of LSB is about 6% of the range in this case and 16 bit-wide integer arithmetic is enough to compute cepstra with an error lower than the LSB quantization step.

3. ARCHITECTURE

The ACF computation has been implemented as shown in Fig. 2: the sixteen scaled autocorrelation coefficients are computed in parallel on an 8ms frame basis. A counter for each block computes

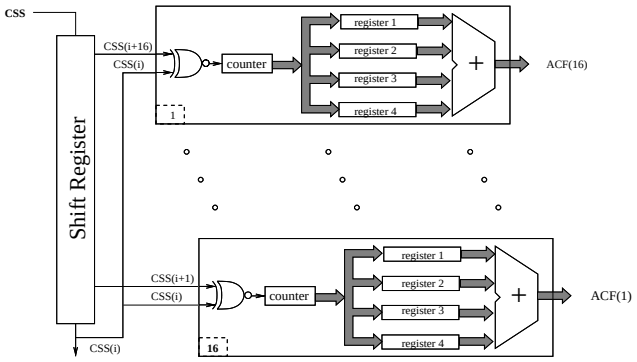

Figure 2: ACF computation

the partial autocorrelation coefficient over the current frame. This value is circularly stored in 4 shadow registers, then the sum of their values returns the ACF coefficient of the current window.

The LD and cepstrum recursions are implemented using 16-bit two's complement integer arithmetic. The two recursions require 256 16x16-bit and 135 16x8-bit multiplications, 16 reciprocal computations and 346 16-bit accumulations. To pack all these operations in 256 clock cycles, an arithmetic unit with two multipliers, two accumulators and one shifter has been implemented and the two basic computations have been joined.
The block that computes the function $f(\overline{\alpha})$ has been implemented by a piecewise-linear approximation made up of four segments with slope -8, -4, -2, -1 respectively. The maximum relative error of the piecewise linear approximation over the entire α range is 1.9%. Compared with a simple lookup table implementation this solution guarantees a strictly decreasing function that avoids recursion convergence problems.

To store the temporary values required by the operations, a set of 30 registers of 16 bits each is provided, in addition to the output data register. A microcoded sequencer controls all the operations and performs the correct input and output selection of each block and the current computation performed by the arithmetic unit. The block diagram of the feature extractor is shown in Fig. 3.

A partial power-down modality has been implemented in order to reduce the power consumption during the silence in the speech signal. For this purpose, an external signal (COMPUTE), controlled by a voice detector is provided. The power reduction in

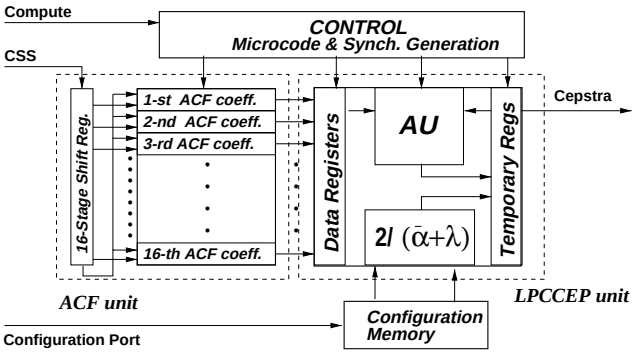

Figure 3: Chip block diagram

sleeping mode is about 50%. The need for a partial power-down is due to the fact that the autocorrelation coefficients must be still computed in order to promptly react to a new voice command when issued.

The clock frequency in the ACF unit is the speech sampling frequency (8khz), while in the LPCCEP unit the clock frequency is 32khz that gives 256 clock cycles per frame. These clocks are derived from a 64khz external clock signal.

4. IMPLEMENTATION AND RESULTS

The chip has been implemented using a three-metal, 0.5μm CMOS Sea of Gates technology. The chosen power supply is 0.9V which is less than the sum $V_{TN} + |V_{TP}|$. Hence, no short-circuit power is dissipated even when very slow commutations occur. The measured power consumption of the chip at the nominal operation frequency of 64kHz is 30μW at 0.9V. Chip measurements are reported in Fig. 4.

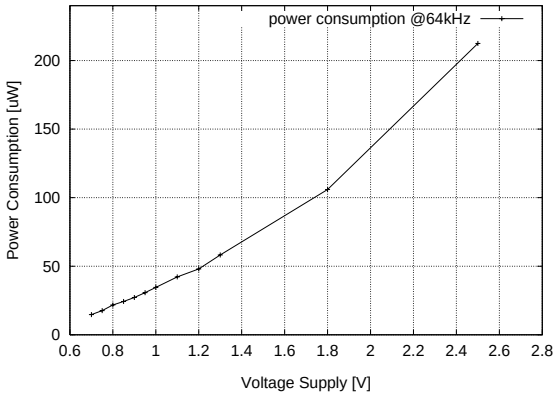

Figure 4: Measured power consumption vs. voltage supply.

The presented unit has been used to evaluate how the proposed set of features compares with conventional approaches in recognition accuracy. The recognition system is based on a pattern matching approach. Classification relies on a Dynamic Time Warping

time alignment procedure based on L2 distance [3]. Tests of the recognition system have been based on "TI 46 Isolated Word Corpus", in particular a set of 20 words (digits and commands) has been used.
The best recognition rates for our recognition system have been reached using an LP filter of order 12 to compute 11 cepstral coefficients on 12BQ speech signal and using a filter of order 16 to compute 15 cepstral coefficients in the case of 1BQ speech. In both cases signal is preemphasized before sampling. Words have been automatically endpointed. In a multi-speaker (8 speakers) task, recognition rate is 99.5% when using features derived from 12BQ speech signal and 98.9% for 1BQ speech signal derived cepstrum . Recognition rate becomes 99% and 97% respectively when white noise at 10dB SNR is added to all the utterances in the database before training and classification.

Process Technology:	3-metal, 0.5μm CMOS SoG
Number of MOST:	100,000
Threshold Voltages:	V_{TN}=0.62V, V_{TP}=-0.64V
Operating Conditions:	V_{DD}=0.9V, f_{clk}=64kHz
Power Consumption:	30μW (150μW/MOPS)

Table 1: Chip features

5. CONCLUSION

The design of a VLSI architecture has been described starting from the algorithmic point of view down to the electrical level. A novel approach based on 1 bit quantization of speech signal that avoids full A/D conversion has been adopted. This approach makes the design of the feature extractor fully digital leaving out a simple shaping (preemphasis), band limiting analogue filter.

An isolated-word small-vocabulary speech recognizer based cepstral coefficients from linear predictive analysis of 1 bit quantized speech has been tested. Its recognition accuracy is within 2% below a system based on linear predictive cepstral features derived from 12BQ speech signal.

In the authors opinion these results in terms of power consumption show the feasibility of a single-chip speech recognizer in submicron CMOS technology featuring a power consumption in the range of few hundreds microwatts.

6. REFERENCES

[1] J.C.R.Licklider. Effects of amplitude distorsion upon the intelligibility of speech. *J. Acoust. Soc. Am.*, 18:429–434, 1946.

[2] V. Lipovac. Zero-crossing-based linear prediction for speech recognition. *Electronic Letters*, 25(2):90–92, 1989.

[3] M. Felici, A. Ferrari, M. Borgatti, and R. Guerrieri. Extraction of LP-based features fron one-Bit quantized speech signals for recognition purposes. In *Proceedings of the 8th EUropean SIgnal Processing COnference*. EURASIP, September 1996.

[4] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen. Low-power digital CMOS design. *IEEE Journal of Solid–State Circuits*, 27:473–484, April 1992.

[5] John Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

[6] J.W. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9):1215–1247, 1993.