

# A 0.9V, 30 $\mu$ W Feature Extractor for Remote Speech Recognition

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

DEIS – Università di Bologna, ITALY

Central R&D – SGS–THOMSON Microelectronics, ITALY

E-mail: {aferrari, mborgatti, mfelici, rguerrieri}@deis.unibo.it

## Abstract

*A feature extraction chip for speech recognition computes fifteen cepstra each 8ms at 64kHz clock rate and dissipates 30 $\mu$ W at 0.9V. It has been implemented as a gate array in a 0.5 $\mu$ m, three-metal CMOS technology. The average energy required to process a single word of the TI46 speech corpora is 10 $\mu$ J. It achieves recognition rates over 98% in isolated-word, speech recognition tasks.*

## 1. Introduction

Personal systems providing ubiquitous access to distributed data and computing servers are object of increasing interest. Portable communication terminals must feature low-power electronics to improve autonomy and easy-to-use user interface. Vocal I/O, based on speech recognition and synthesis, delivers the easiest user access and reduces physical dimensions when compared with conventional keyboard-based or graphic interfaces. On the other hand, speech recognition is a very computational- and memory-intensive task and, generally, cannot be performed locally on the user terminal except for very simple command-oriented recognition tasks.

Previous works [1] showed applications in which the recognition task has been moved on a remote host and the vocal I/O is reduced to speech compression and coding for the radio link.

In our system the goal of low-power speech recognition has been reached partitioning the recognizer between the portable terminal and the remote hosts such that the power consumption involved with the recognition task and the data transmission leads to a minimum. This is achieved performing the speech feature extraction (which usually requires less than the 10% of the total computational complexity of the recognition) on the terminal and further compressing the feature stream before the radio channel coding. The extracted features are sent to a base station that completes the speech recognition task.

An overview of the speech processing section of the portable terminal is shown in Fig. 1. The system receives the analog speech signal and passes it to a low-pass filter performing the signal anti-aliasing and preemphasis. A two-level quantizer produces the digital signal *css*, then the feature extractor computes the cepstral coefficients on a frame by frame basis. The coefficients are compressed and sent to the channel coding and RF sections.

The goal of very low-power consumption has been pursued starting from the algorithmic down to the circuit level. Suitable algorithms should feature simple arithmetics and exploit data correlation to reduce switching activity. Moreover, they should achieve efficient parallelization of the computation so that an architectural voltage scaling approach can be exploited [2].

The main contribution of this paper is the description of the implementation of a 0.9V, 30 $\mu$ W feature extractor that implements a novel, highly simplified algorithm [3] achieving efficient speech feature computation and compression and very low-power and low-voltage operation.

The implemented algorithm avoids the full analog-to-digital conversion of the speech signal. This simplification allows us to minimize the value of the voltage supply at which the whole system can

run since the analog section is significantly reduced. On the other hand, the performance of the speech recognition system is only marginally affected by the two-level quantization of the speech signal [3].

## 2. Chip Architecture

From an algorithmic point of view the computation of the cepstral coefficients can be split into three main steps: the evaluation of the autocorrelation function of the speech segment under processing, the LP analysis and the cepstral transformation [3, 4]. Finally, the feature data stream is compressed reducing its temporal correlation and using a Huffman coding for each coefficient.

The speech signal is sampled at 8kHz and quantized at two levels. The obtained digital signal  $css$  is segmented in windows of 32ms overlapped by 24ms. Hence, each 8ms (frame rate) a new window is processed. Table 1 summarizes the amount of operations per frame of the main steps of the computation of a 15-th order cepstral analysis. This corresponds to a 0.23MOPS of computational power. The cost of the compression step is a negligible part (less than the 5%) of the whole amount of computations.

The architecture of the feature extractor is shown in Fig. 2. The signal  $css$  is delayed by a 16-stage shift register and, for each window (256 samples), sixteen scaled autocorrelation coefficients are computed as follows:

$$ACF(k) = \sum_{i=0}^{255} css(i) \oplus css(i+k) \quad k = 1, \dots, 16$$

where, since  $css$  is a one-bit signal, the multiplications between samples have been replaced by XNORs thus simplifying the computation and drastically reducing power consumption. As shown in Fig. 3, the sixteen scaled autocorrelation coefficients are computed by sixteen blocks, in parallel and on an 8ms frame basis. A counter for each block computes the partial autocorrelation coefficient over the current frame. This value is circularly stored in 4 shadow registers and, finally, an 8-bit adder adds these counter values for the whole window (32ms) every 8ms. The computed coefficients are then stored in a data register for the cepstral analysis.

An arithmetic unit, featuring a 16x16- and a 16x8-bit multipliers, has been designed to perform a Levinson-Durbin recursion and a cepstral transformation [4] at the maximum achievable degree of parallelism. Using the 16-th order autocorrelation function, sixteen LP-coefficients and then fifteen cepstrum are computed each frame. While most of the intermediate numeric computations are carried out using a 16-bit numeric representation, each cepstral coefficient requires only 5 bits for a total of 75 bits to represent one frame vector.

The fixed-point implementation of the chosen algorithm requires 256 16x16-bit and 135 16x8-bit multiplications, 16 inverse computations and 362 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 utilization of the logic in this unit is 83%. 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 that computes the inverse ( $1/\alpha$ ) has been simplified since the input range is algorithmically bounded to  $]\lambda/2, (1 + \lambda)/2]$ , where  $\lambda$  is a stabilization parameter in  $[0.3, 0.45]$  range applied to the 0-th order autocorrelation coefficient [4]. The reciprocal function has been approximated by a piecewise-linear approximation made up of four segments with slope -16, -8, -4, -2 respectively. The connecting points of the segments are computed to minimize the mean square error of the approximation on the  $\lambda$ 's range. The maximum relative error of the piecewise linear approximation over the entire  $\lambda$  range is 1.9%. The value of  $\lambda$  and the values of the connecting points (that depends on it) are programmable by the user in an internal configuration memory as well as the values of 15 cepstral normalization coefficients. This configuration memory is written immediately after the system power up.

In order to minimize the transmitted energy per word, the bit rate has been reduced introducing a frame compressor followed by a Huffman coder. The frame compressor reduces the number of frames per word since a new frame vector is emitted only if the Manhattan distance between the new frame and

the last emitted one is greater than a pre-defined threshold. The average reduction of transmitted frames given by this step is 66%. The Huffman coder encodes each cepstral coefficient with a precomputed code stored in a ROM. Since our algorithmic study shows that only 5 bits are required by a cepstral coefficient, this ROM has 32 entries of 14 bits each (11 representing the code and 3 for its length), making the power consumption of this block almost negligible. This last step reduces the required number of bits by an additional 15%.

A partial power-down modality has been implemented in order to reduce the power consumption during the silence in the speech signal, since no frames should be transmitted. For this purpose, an external signal (COMPUTE), controlled by a voice detector (still in development) or by the user, is provided. The power reduction in *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-OUT unit the clock frequency is 32kHz that gives 256 clock cycles per frame. These clocks are derived from a 64kHz external clock signal.

### 3. Chip Implementation and Measurement Results

The chip has been implemented using a three metal, 0.5 $\mu$ m CMOS Sea of Gates technology. A photograph of the chip is shown in Fig. 5. Table 2 summarizes the relevant technology data of our implementation. The chosen power supply level 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 design flow has been based on automatic synthesis from a VHDL RTL description. Automatic place & route has been carried out on a large base array already available to reduce the manufacturing turn-around time. The chip requires 100k transistors and meets its functional specifications down to 0.7V.

The measured power consumption of the chip at the nominal operation frequency of 64kHz and the maximum operation cycle length for various supply voltages are shown in Fig. 4. The measured average power consumption of the chip is 27 $\mu$ W at 0.9V. Measurements carried out on different spoken utterances, background noise and silence show that the variation of the power consumption of the computing core in different operating conditions is less than 10%. This suggests that most of the power consumption is due to sequential logic triggering and clock distribution which do not depend on the correlation of the signals. Hence the amount of energy needed to process a word is roughly proportional to the duration of the word itself.

The chip performance has been measured using an isolated-word vocabulary composed of ten digits and ten computer-oriented commands (TI46 [5]). Recognition rates over 98% have been reported in speaker-dependent and multispeaker tasks [3]. The average bit rate is 2.7kbps for male speakers and 2.8kbps for female ones. The average number of bits to be transmitted per word is 1130 and the average energy per word is 10 $\mu$ J at 0.9V.

The measured energy  $\times$  delay product versus power supply is also plotted in Fig. 4. The penalty due to the bias voltage significantly below the theoretical minimum located at  $2.5 \div 3$  threshold voltages is less than predicted using well-known models [6, 7]. At 0.9V the energy delay product is only 1.7 times the minimum value obtained at 1.8V. This is due to a 15% reduction of the total switched capacitance since most of each switching event of CMOS gates occurs in the weak-inversion region of the transistors. SPICE simulations show that the input capacitance of CMOS gates has a 40% reduction when the voltage supply drops from 1.8V to 0.9V. The effective reduction of the switched capacitance in the core logic is lowered by the interconnection capacitance that does not reduce with voltage supply and can become a dominant factor. Therefore, routing optimization in order to reduce interconnection capacitance is the key factor to obtain effective designs with voltage supply less than two threshold voltages.

### 4. Acknowledgments

The authors wish to thank Prof. G. Baccarani for his help and encouragement.

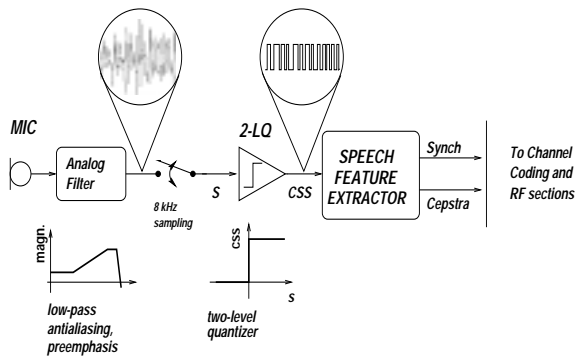


Figure 1. System-level overview

Oper.	ACF	LP	Cepstral	Tot
ADD	1088	256	106	1450
MUL	0	256	135	391

Table 1. Algorithm computational complexity

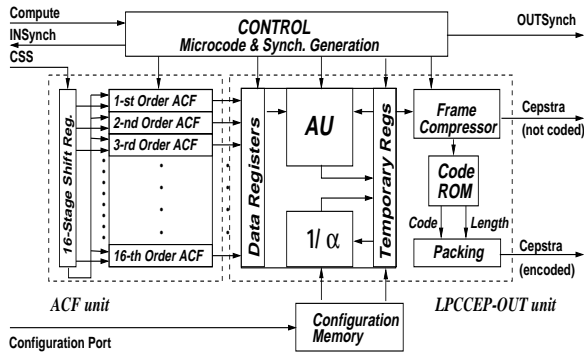


Figure 2. Chip block diagram

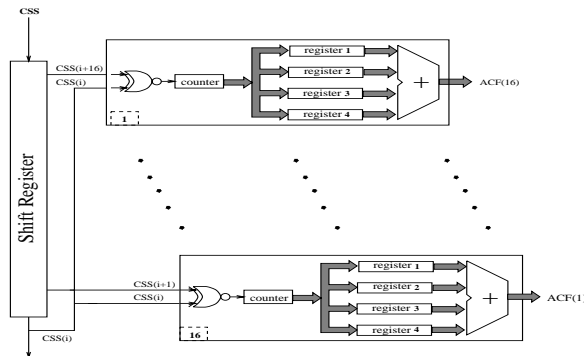


Figure 3. ACF computation

Process Technology:	3-metal, 0.5 $\mu$ m CMOS Sea of Gates
W/L ratio:	6.4 $\mu$ m/0.5 $\mu$ m
Threshold Voltages:	$V_{TN}=0.62V, V_{TP}=-0.64V$
Number of MOST:	100k
Voltage Supply:	0.9V
Clock Rate:	64kHz
Power Consumption:	30 $\mu$ W (150 $\mu$ W/MOPS)
Package:	144-lead PGA

Table 2. Chip features

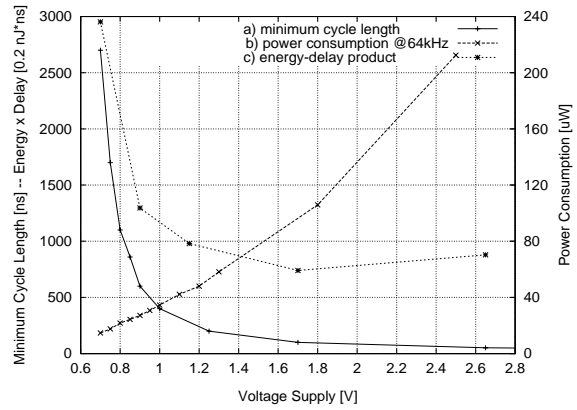


Figure 4. Measured speed and power

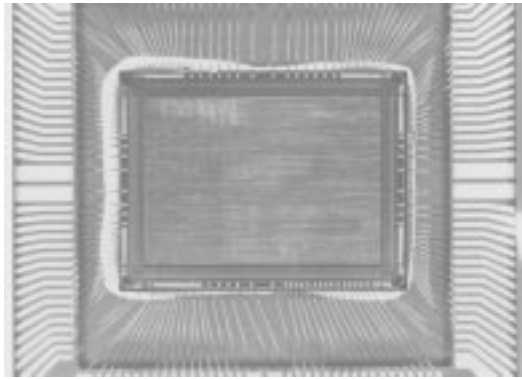


Figure 5. Chip microphotograph

## References

- [1] A.P. Chandrakasan, A. Burstein, and R.W. Brodersen. A Low-Power Chipset for a Portable Multimedia I/O Terminal. *IEEE Journal of Solid-State Circuits*, 29(12):1415–1428, December 1994.
- [2] 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.
- [3] M. Felici, A. Ferrari, M. Borgatti, and R. Guerrieri. Extraction of LP-Based Features from One-Bit Quantized Speech Signals for Recognition Purposes. In *8th European Signal Processing Conference*. EURASIP, September 1996.
- [4] J.W. Picone. Signal Modeling Techniques in Speech Recognition. *Proceedings of the IEEE*, 81(9):1215–1247, 1993.
- [5] G.R. Doddington and T.B. Schalk. Speech Recognition: Turning Theory to Practice. *IEEE Spectrum*, 18, September 1981.
- [6] M. Cao and H. Stork. Optimization of Low-Power Quarter Micron MOSFETS. In *IEEE Symp. on Low-Power Electronics, Digest of Technical Papers*, pages 84–85, 1995.
- [7] J.B. Burr and A.M. Peterson. Ultra Low-Power CMOS Technology. In *NASA VLSI Design Symposium*, pages 4.2.1–4.2.13, 1991.