Hardware Implementation of a Logarithm Processor for Mel-cepstra Parameters Extraction

Roger Pizzatto Nunes, José Gómez, Dante Barone, Sergio Bampi
rogerpn@inf.ufrgs.br

Universidade Federal do Rio Grande do Sul - Instituto de Informática
Cx. Postal 15064 - Av. Bento Gonçalves, 9500 - Campus do Vale - Bloco IV
Bairro Agronomia - Porto Alegre - RS - Brasil - 91501-970
Fone: 55-51-33167036 Fax: 55-51-33167308

Abstract

This work, shows the implementation of a CORDIC architecture based in FPGA, applied to the logarithm calculus for mel-cepstral parameters extraction in an automatic speech recognition system.

1. Introduction

Comparative studies have shown that in the most of operation modes of speech recognition systems, the mel-cepstra parameters produce the best recognition rates. In figure 1, the mel-cepstra parameters extraction is shown.

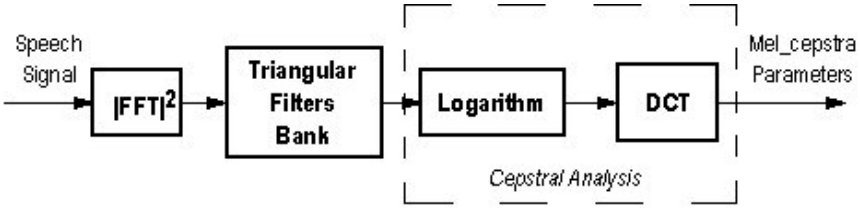

Figure 1 – Mel-cepstra parameters extraction.

The Fast Fourier Transform (FFT) is used to obtain the signal spectral energy [RAB 78]. In the sequence, spectral energy is filtered by a triangular filters bank and the logarithm function is applied to the result. Finally, the Discrete Cosine Transform (DCT) is used to generate a mel-cepstra parameters vector of the speech signal [GOM 01].

In this work, the CORDIC algorithm (Coordinate Rotation Digital Computer) [VOL 59] is used to implement the logarithm function. In the next sections, such implementation is shown.

2. CORDIC Algorithm

By means of the CORDIC algorithm, a vector (x_0, y_0) is rotated through an angle (rotation mode) or it is aproximated to the coordinate axis (vectoring mode).

These operations could be performed in linear, circular, and hyperbolic coordinate systems, but only the hyperbolic coordinate system is considered here. The algorithm is based on rotations over known elementary angles ($arctgh(2^{-i})$ in hyperbolic coordinates), stored in a table.

The basic iteration or *microrotation* in hyperbolic coordinates is described by the following mathematical equations:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{bmatrix} = \begin{bmatrix} x_i + y_i \cdot \delta_i \cdot 2^{-i} \\ y_i + x_i \cdot \delta_i \cdot 2^{-i} \\ z_i - \delta_i \cdot \gamma_i \end{bmatrix} \quad (1)$$

where (x_0, y_0) are the initial coordinates of the vector and the z coordinate accumulates the angle. The coefficient δ_i specifies the direction of each microrotation.

CORDIC has a convergence domain limited by $[-\pi/2, \pi/2]$, due to use of 2^0 for the hyperbolic tangent in the first iteration [WAL 71][AND 96][EDW 96]. For numbers out of this interval, CORDIC is applied to the scaled value. The relation between arctangent and natural logarithm is:

$$\ln(\beta) = 2 \cdot arctgh \cdot \left(\frac{\beta - 1}{\beta + 1} \right) \quad (2)$$

Owing to the divergence of hyperbolic arctangent in the singular point 1, the interval for the value should be $0.5 \le \beta < 1$. Then, values in $1 < \beta < 0.5$ are multiplied or divided by 2 until the rule above is satisfied. In analytic form:

$$\ln(\beta) = \ln(\beta' \cdot 2^n) = \ln(\beta') + n \cdot \ln(2) \quad (3)$$

where *n* may be a positive, zero or negative integer number, when $\beta < 0.5$, $0.5 \le \beta < 1$ or $\beta > 1$, respectively.

3. FPGA Implementation

The system proposed is composed by two processing stages. Initially, the input value is analyzed and, if necessary, the scaling process is performed. Next, the CORDIC algorithm of the residual value is computed, and its result is added with the radix from the scaling stage (see figure 2).

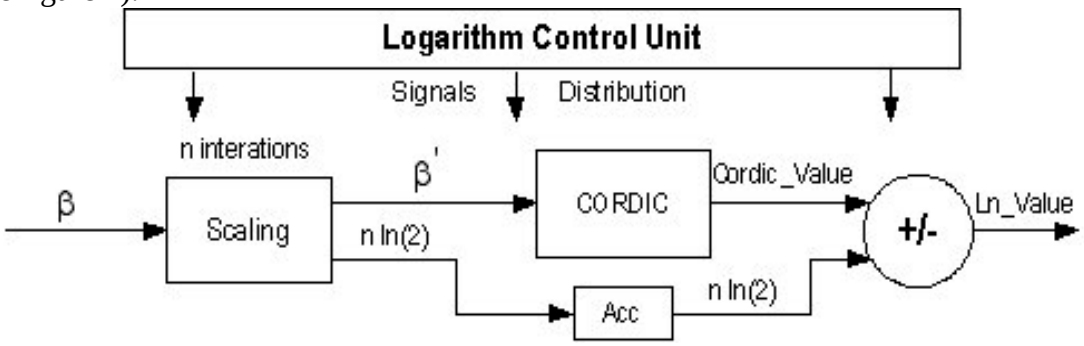

Figure 2 – The Logarithm processor.

In figure 3, the datapath of the scaling block is shown.

In the beginning, the multiplexor *Mux_Load* receive the initial value sending it to the combinational component named *Analyze*, which decide the necessity of shift left (multiplication by 2), or right (division by 2). If necessary, the value is passed to the *D_Mux*,

configuring the scaling operation and storing the result in the register named *Acc_PValue*. In the same time, the partials adds of $\ln(2)$ are computed. The final result β' is sent to CORDIC block, to compute the first term of (3), while the value $n \cdot Ln(2)$ is stored in the *Acc* register (see figure 2), for future processing.

In figure 4, the CORDIC architecture developed is shown.

In the first moment, the β' value is loaded with '1' (selected by *P* multiplexers) by the *L* multiplexers, executing an addition or subtraction and storing the values in its registers. Next, the algorithm for logarithm computation is executed and the final value $Ln(\beta')$ is added to the previous number obtained by scaling. It was used fixed point representation, with 16 bits precision and two's complement, where 10 bits are dedicated for the integer part and 6 bits for the decimal part. All the system was tested by comparison of results from Maxplus II and Matlab tools. Table 1 shows the results of hardware implementation simulation. The processor was synthesized in a EPF10K30ETC144-1 device of FLEX10K family

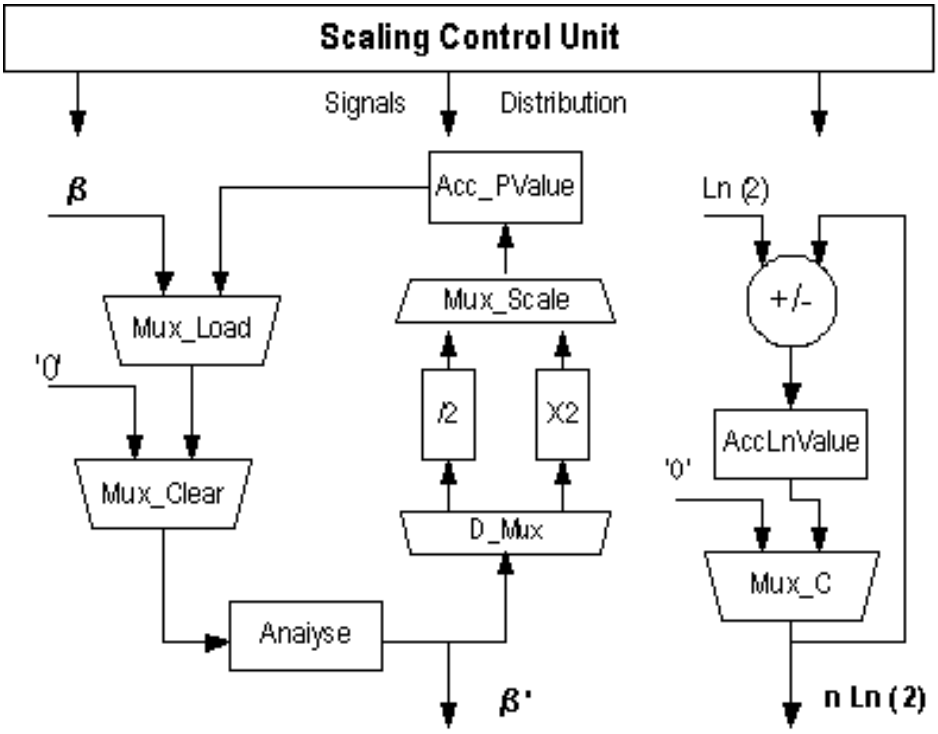

Figure 3 – Scaling Hardware.

4. Conclusion

The CORDIC is a versatile algorithm that, with the flexibility of VHDL description and the potential of FPGA based systems, allows the development of efficient circuits for many proposes.

The goal is to use this processor for mel-cepstra parameters extraction of voice signals, in an automatic speech recognition system.

Table 1 – Hardware implementation results

Device	*Input Pins*	*Output Pins*	*Bidir Pins*	*Memory Bits %*	*Memory Utilized*	*LC's*	*LC´s (% Utilized)*	*Clock Frequency (MHz)*
EPF10K30ETC144-1	20	17	0	128	0%	650	37%	28.73

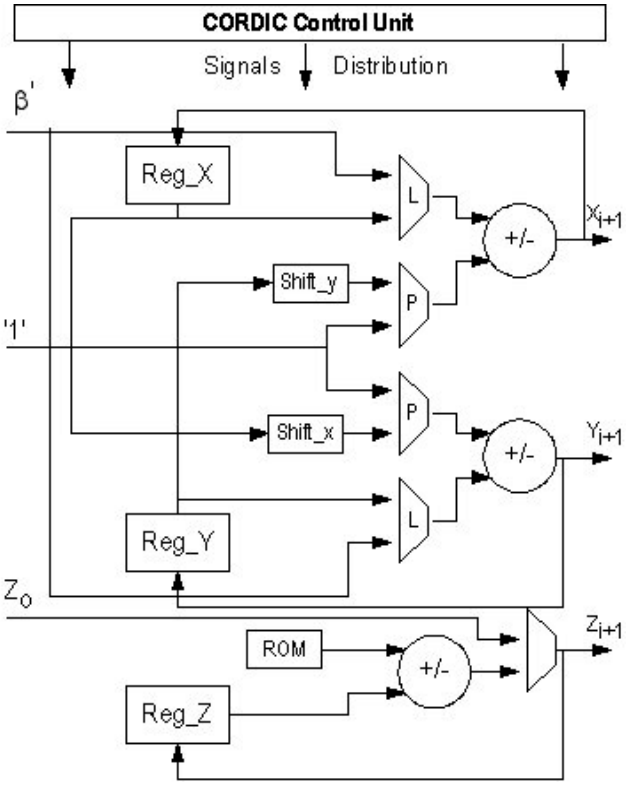

Figure 4 – CORDIC Implementation.

References

[VOL 59] VOLDER, Jack E. "The CORDIC Trigonometric Computing Technique", IRE Transactions on Electronic Computers, volume EC-8, No. 3, September 1959, pp330-334.

[WAL 71] WALTHER, J.S. "A unified Algorithm For Elementary Functions", Spring Joint Computer Conf., 1971, proc.,pp. 379-385.

[AND 96] ANDRAKA , Ray. "A survey of CORDIC algorithms for FPGA based computers", FPGA Monterey CA USA, 1998.

[EDW 96] EDWARDS, Bruce H.; JACKSON, Robert T.; UNDERWOOD, Jeremy M. "How Do Calculators Calculate?, Proceedings of the Ninth Annual International Conference on Technology in Collegiate Mathematics, Reno, Nevada, November, 1996.

[RAB 78] RABINER, L. R.; SCHAFER, R. W. "Digital Processing of Speech Signals", New Jersey: Prentice-Hall, 1978. 512p.

[GOM 01] GOMEZ, José Cipriano; NUNES, Roger Pizzatto; BARONE, Dante; BAMPI, Sergio. "Arquitetura Específica de Pré-processamento com Extração de Parâmetros Mel-cepstrais para um Sistema de Reconhecimento Automático de Voz", VII Workshop Iberchip, Março 2001, Montevidéu, pág.62.