SYMBOLIC SYNTHESIS OF NON-LINEAR DATA CONVERTERS

J. Guilherme [1] N. C. Horta [2] J. E. Franca [2]

[1] Escola Superior de Tecnologia de Tomar
Quinta do Contador, Estrada Serra
2300 Tomar Portugal
Phone/Fax: (351)-49-328196/7
Email: jorge.g@ipt.pt

[2] Instituto Superior T cnico
IST Center for Microsystems
Av. Rovisco Pais, 1, 1096 Lisboa Codex, Portugal
Phone/Fax: (351)-1-8417675
Email: n.horta@ieee.org

ABSTRACT

This paper discusses the use of symbolic methods applied to the design automation of non-linear data converters. The proposed approach is an extension of an already proved methodology, for the symbolic synthesis of linear data converters, by both introducing non-linear characteristics in a modified signal flow graph approach and reusing library functional blocks by appropriate retargeting for the new design goals. The design process is here exemplified with recently proposed architectures for logarithmic data converters.

1. INTRODUCTION

On the recent past, large attention have been devoted to the development of reliable design automation (DA) tools for data conversion systems. However, this have been done exclusively for linear data converters besides the use of non-linear converters, e.g. logarithmic converters, in communications, instrumentation and hearing aids, among other application areas, where it is needed to adapt the dynamics of signals to the dynamics of the channels used for transmission [1]. Although, originally the logarithmic converters have been implemented in bipolar technology to explore the inherent exponential I-V characteristic of the transistors [2]. Alternative architectures for implementation in the more cost-effective CMOS technology have also been proposed based on the piecewise approximation of a logarithmic function [3]. Examples of these include, among others, the successive approximation [4] and the pulse width modulation type of logarithmic A/D converters [5]. Recently new types of logarithmic A/D converters suitable for high-frequency applications and implementation in CMOS technology had been proposed, based in the pipeline [6], and in the two-step flash architectures [7], whereby the signal operations are carried out in the logarithmic domain by simple scaling operations. Therefore, this paper proposes covering part of the existing lack of DA tools for data conversion systems by expanding a proved methodology for linear converters [**N**], to logarithmic converters specified at algorithm level and aiming at determining both the appropriate topologies and sub-block specifications.

2. ANALYSIS OF NON –LINEAR ADCS

Although linear and non-linear converters have different intrinsic operations the structure of conversion algorithms are identical. Next, those differences and requirements for their implementation are outlined.

2.1. I/O Signal Characterization

The input and output signal relation on linear and non-linear, e.g. logarithmic, A/D data converters can be compared by the different analog threshold signals that lead to similar quantization levels. Assuming an A/D conversion N bits and considering an analog input signal ranging from Vin_min to Vin_max, the threshold signals for the nth quantization level would be for the linear and logarithmic cases, respectively,

$$V_{tn} = V_{in\_Min} + \left(V_{in\_\max} - V_{in\_Min}\right)\frac{n}{2^N} \qquad (1)$$

and $$V_{tn} = V_{in\_Min}\, e^{\ln\left(V_{in\_\max}/V_{in\_Min}\right)\frac{n}{2^N}} \qquad (2)$$

The comutation of arithmetic operations between the two signal domains is now outlined. More, the logarithmic approach allows the achievement of different dynamic ranges, defined as the ratio between maximum input signal range and the LSB, for identical input signal ranges, as depicted in fig. 1, by selecting the appropriate k value from [7]

$$DR_{db} = 20\log\frac{1}{k\left(e^{\ln(1+1/k)2^N} - 1\right)}. \qquad (3)$$

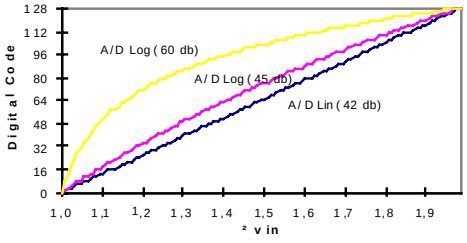


Fig. 1: Linear Vs Logarithmic Approach.

2.2. Signal Processing Operation

As in the linear conversion case, algorithms can be defined from the one step conversion, or parallel, till the 2^n steps conversion, or serial. However, due to specific functions of non-linear data converters special attention is required to identify the new functional and electrical blocks to be included in the design library to fulfill a large scope of algorithm specifications. In order to better illustrate the required signal processing functions a two-step conversion was chosen once it belongs to a class of algorithms, multi-step, covering a large scope of applications. In the linear case a usual two-step topology, illustrated in fig. 2, is composed by a linear 1-step A/D converter to extract the MSBs followed by a residual structure generation, including both a linear D/A converter and a subtractor, and again a linear 1-step A/D converter to extract the LSBs and which may be preceded by an amplification unit.

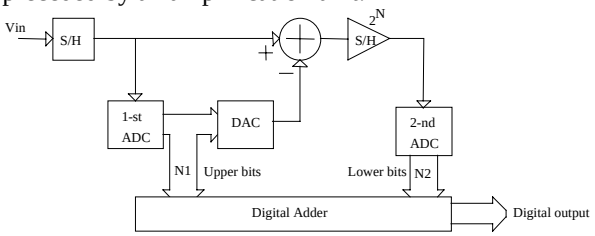


Fig. 2: Linear Two-Step ADC Topology.

According to the I/O signal characterization of the previous functions the logarithmic two-step A/D converter should be carrie out similar operations but in the logarithmic domain. This implies that the structures corresponding to the 1-step conversion, respectively with N1 bits and N2 bits, must be performed by a logarithmic conversion structure with the threshold voltages given by (2) and resulting in the digital code

$$n = trunc\left(\frac{\ln(V_{in}/V_{\min})}{\ln(V_{\max}/V_{\min})}2^{N1}\right) \qquad (4)$$

in analogy to linear case, given by

$$n = trunc\left(\frac{V_{in}-V_{\min}}{V_{\max}-V_{\min}}2^{N1}\right). \qquad (5)$$

The intermediate D/A converter should perform an exponential conversion to achieve the quantified voltage in the same scale as the input, corresponding to the threshold voltage associated to the previous digital code and given by (2). Then, the residue signal is here achieved by a division operation due to the logarithmic domain. Next, to use the same scale as in the previous A/D conversion the residue signal should be raised by 2^{N1}, again due to logarithm domain. Finally, another possibility to avoid using the lower part of the signal range can be achieved by moving the signal to the highest signal range. While this operation, the introduction of an offset, in the linear domain would result in the same difference between successive threshold voltages in the logarithmic domain due to the amplification operation this difference is also amplified, by a factor of

$$k = e^{\frac{\log(1+a)}{2^N}\left(2^N - 2^{N2}\right)} \qquad (6)$$

allowing the specs relaxation of threshold level detectors, i.e. comparators.

2.3 Typical Functional and Electrical Blocks

The previous analysis lead to the need of defining specific blocks for implementing logarithmic converters. Namely an 1-step logarithmic A/D converter unit, an exponential D/A converter, a signal divider and conditionally an exponential amplifier. While the first blocks are easily derived from existing ones for the linear case as illustrated in figures 3 to 5 the exponential amplifier have not a reliable implementation within the precision required for data conversion systems. In the following discussion the exponential amplifier will be avoided by considering distinct quantization scales on the two logarithmic A/D converters.

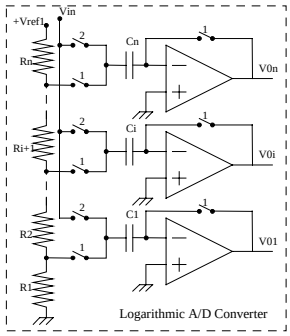


Fig. 3: Logarithmic One-Step ADC.

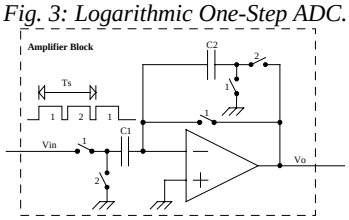


Fig. 4: Amplifier Block.

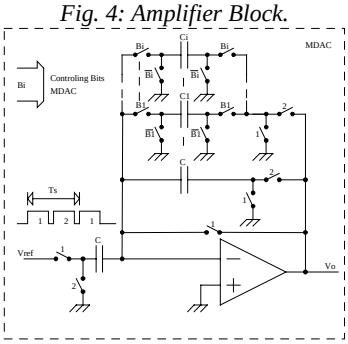


Fig. 5: Basic Module of the Exponential MDAC.

3. DA OF NON-LINEAR DATA CONVERTERS

The DA of non-linear data converters from algorithm descriptions will be now described based on the required non-linear functionalities outlined by the previous discussion and the developed methodology for the algorithm-driven synthesis, illustrated in fig. 6.

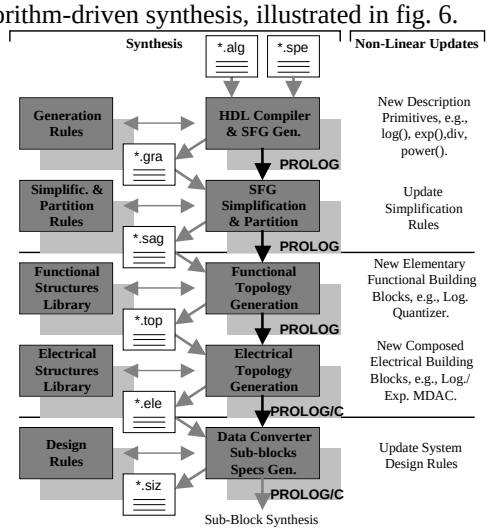


Fig. 6: Design Methodology

3.1. Algorithm and Modified SFG Descriptions

The algorithm description, sampled in fig. 7, should now cover non-linear operations. Though, a superset of the previously used HDL must be defined to include operations such as logarithm, exponential and power of a signal and also division and multiplication to perform on the logarithmic domain the equivalents of subtraction and addition on the linear domain.

```
…
If k4 vx1 > k1 ( k7 Vrefp + k8 Vrefn) then
  make l2 =1 else l2 =0 endif
…
If lm2 then
  make vx3 = vx3 + k14 Vrefp endif
…
  make vxresampl = k16 (vx1/vx3)
…
```

Fig. 7: Fidel HDL Algorithm Description.

Next, the algorithm synthesis implies the identification of functional structures and there instantiation with electrical circuitry which is achieved through the symbolic analysis of a modified SFG representation of the algorithm description. The modified SFG approach introduced in [**N**] already allowed the distinct representation of analog and digital signals. Now, this representation is expanded to support non-linear descriptions made either implicitly or explicitly in the HDL algorithm description. An implicit non-linear description occurs whenever a non-linear operation is described as a series of coefficients values pre-determined, e.g. define the logarithmic A/D processing as a comparison to exponential threshold values in step of using explicitly the I/O expression containing either the logarithm or the exponential function. Therefore, an explicit description corresponds to a direct use of the new primitives in the algorithm description. So, an expansion of the modified SFG to represent non-linear descriptions is achieved by defining functional branches, illustrated in fig. 8, which transform the input according to specified function, and redefining the nodes to be seen not simple as sum points but either sum or multiplication points.

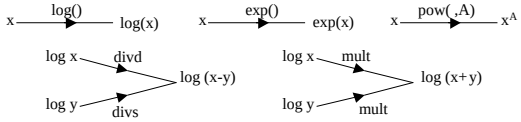
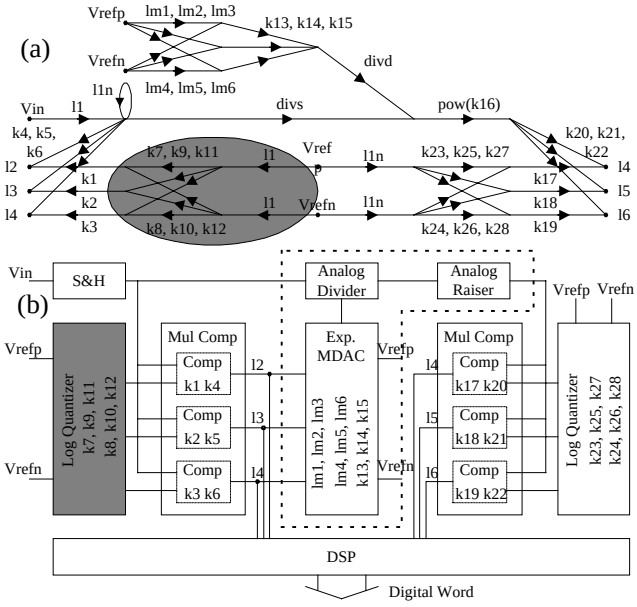


Fig. 8: Introduced non-linear and 2-signal operations.

3.2. Topology Mapping

From the algorithm description and using the translation/simplification method described in [**N**] the following SFG are obtained for an example of a multi-step non-linear conversion, e.g. 2-step conversion. The SFG, illustrated in fig. 9 (a), outlines the use of non-linear operations as well as allows the identification of patterns corresponding to specific functionalities. Therefore, a pattern recognition task leads to a topology, illustrated in fig. 9 (b), generation independently from



the technology required for the implementation.

Fig. 9: Topology Generation Process: (a) Analog partition SFG. (b) Functional topology.

3.3. Architecture Definition and Characterization

The identified functional blocks together with their links define the conversion topology. However, depending on

the set of desired specifications and technology implementation may vary significantly. Next, an example of topology instantiation based on switch capacitor modules is shown to illustrate the automatic architecture generation. First, the logarithm quantizer, the multiple comparator and the sample and hold associated the way illustrated in fig. 9. correspond to the well known 1-step or flash converter, however, having in this case a logarithmic characteristic. Another association of function blocks can be clearly identified to perform a similar operation. Finally, the three residual blocks connected as shown on the topology picture can be implemented by only one electrical structure, illustrated in figure 10, which performs the three operations when activated with the appropriate control signals and having the capacitors weighted based on the k coefficients extracted from the graph. Note that although the graph was presented for two-step conversion with 4 bits, just for simplicity, the result is obviously generalized.

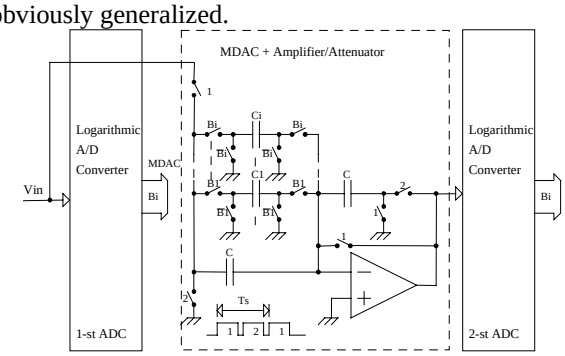


Fig. 10: Data converter architecture.

Finally, a sizing procedure is made by associating to each electrical block a relation between component values which are absolutely determined on run-time depending on the performance specs and technology desired. In the present case, the following values and relations are achieved for a 7 bit (4+3) two-step converter:

Block\Par.	Rs and Cs	OpAmp/Comp.
A/D Log1	$Ri = Rk * \left(\frac{V_{tn\_i} - V_{tn\_i-1}}{V_{in\_max} - V_{in\_min}} \right)$	$A = 20 * \log \left(\frac{V_{in\_max} - V_{in\_min}}{4 * (V_{tn\_i} - V_{tn\_i-1})} \right)$
A/D Log2	$Ri = Rk * \left(\frac{V_{tn\_i} - V_{tn\_i-1}}{V_{in\_max} - V_{in\_min}} \right)$	$A = 20 * \log \left(\frac{V_{in\_max} - V_{in\_min}}{4 * (V_{tn\_i} - V_{tn\_i-1})} \right)$
MDAC Exp	$Ci = Ck * e^{\ln(1+1/k)/2^N * (2^N - 2^{N2i})}$	$A = 20 * \log \left(\frac{V_{in\_max} - V_{in\_min}}{4 * (V_{tn\_i} - V_{tn\_i-1})} \right)$

Resulting in a logarithmic characteristic presented in fig. 11.

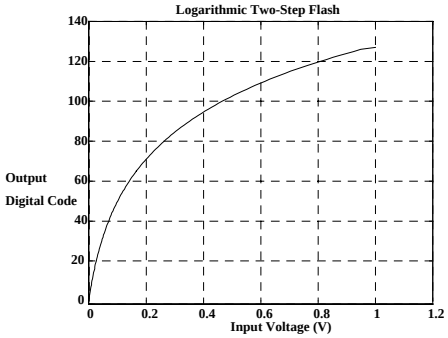


Fig. 11: Conversion Characteristic.

4. CONCLUSIONS

An algorithm-driven methodology initially developed for the synthesis of linear data converters is expanded to cover non-linear data converters defined on the logarithmic domain. This was achieved, mainly, by introducing new non-linear primitives at different abstraction levels, such as, HDL, SFG, topology and architecture descriptions and without having to change the original methodology structure, though, outlining the generality of the adopted approach. Finally, the results presented show the possibility of automatically exploring a wide range of conversion depending on the specification of parameters, such as, dynamic range, resolution, conversion steps.

REFERENCES

[1] Y. P. Tsividis and all, "Companding in Signal Processing", *Electronic Letters* 16th August 1990 Vol. 26, Nº17.

[2] Henry O. Kunz, "Exponential D/A Converter with a Dynamic Range of Eight Decades", *IEEE Trans. on Circuits and Systems*, Vol. CAS-25, no. 7, July 1978.

[3] Jorge Guilherme and José E. Franca, "Digitally-Controlled Analogue Signal Processing and Conversion Techniques Employing a Logarithmic Building Block", *Proc. IEEE Int. Symp. on Circuits and Systems* Vol. 5 pp. 377-380 , London May 1994

[4] Gottschalk, "Logarithmic analog-digital converter using switched attenuators", *Rev. Sci. Instrum.* Nº49, pp. 200-204, Feb. 1978.

[5] J. N. Lygouras, "Nonlinear ADC with Digitally Selectable Quantizing Characteristic", *IEEE Trans. on Nuclear Science*, Vol. 35, No. 5, pp. 1088-1091 October 1988.

[6] Jorge Guilherme and José E. Franca, "New CMOS Logarithmic A/D Converters Employing Pipeline and Algorithmic Architectures", IEEE Proc.Int. Symp. on Circuits and Systems, Seattle May 1995.

[7] Jorge Guilherme and José E. Franca, "New CMOS Logarithmic Two-Step Flash A/D Converters with Digital Error Correction", IEEE 38TH Midwest Symp. on Circuits and Systems, Rio de Janeiro, August 1995.

[N] N. Horta, J. Franca, "Exploring Data Conversion Architecture by Symbolic Computation", *Proc IEEE International Symposium on Circuits and Systems* , pp., 1998

[N] N. Horta, J. Franca, "Algorithm-Driven Synthesis of Data Conversion Architectures", to appear in *IEEE Transactions on Computer-Aided Design*, Vol. 14, No. 10, 1997.

[1] R. Van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, 1994.

[2] G. G. E. Gielen, J. E. Franca, "CAD Tools for Data Converter Design: An Overview", *IEEE Transactions on Circuits and Systems - Part II*, Vol. 43, No. 2, pp. 77-89, 1996.

[3] ANACAD Computer Systems, ELDO V4.4.1 - Users Manual, 1994.

[4] N. Horta, J. Franca, "Algorithm-Driven Synthesis of Data Conversion Architectures", to appear in *IEEE Transactions on Computer-Aided Design*, Vol. 14, No. 10, 1997.

[5] N. E. Franca, M. A. Lança, J. E. Franca, "OpCadsys: An Open Tool for Automatic Synthesis of Circuits Components for Data Converters", *Proc. IEEE Midwest Symposium on Circuits and Systems*, pp. 343-346, 1994.

[6] B. S. Song, S. H. Lee, M. F. Tompsett, "A 10-b 15-MHz CMOS Recycling Two-Step A/D Converter", *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 6, pp. 1328-1337, 1990.

[7] J. C. Vital, "Analog-Digital Data Conversion Integrated System with Functional Reconfiguration and Digital Testability", Ph.D. Thesis, Universidade Técnica de Lisboa - Instituto Superior Técnico, 1994.

[8] J. Goes, J. Vital, J. Franca, "A CMOS 4-bit MDAC with Self-Calibrated 14-bit Linearity for High-Resolution Pipelined A/D Converters", *Proc. IEEE Custom Integrated Circuits Conference*, pp. 6.6.1-6.6.4, 1996.

[9] G. Van der Plas, J. Vandenbussche, G. Gielen, W. Sansen, "EsteMate: a Tool for Automated Power and Area Estimation in Analog Top-Down Design and Synthesis", *Proc. IEEE Custom Integrated Circuits Conference*, pp. 7.7.1-7.7.4, 1997

[10] N. Horta, J. Franca, "Exploring Data Conversion Architecture by Symbolic Computation", *Proc IEEE International Symposium on Circuits and Systems* , pp., 1998.