TREIZIÈME COLLOQUE GRETSI - JUAN-LES-PINS DU 16 AU 20 SEPTEMBRE 1991

SYMBOLIC DESIGN OF LADDER WAVE DIGITAL FILTERS

M. Ansorge, U. Sjöström, I. Defilippis, and F. Pellandini

Institut de Microtechnique, Université de Neuchâtel, Rue A.-L. Breguet 2
CH-2000 NEUCHATEL, Switzerland

RÉSUMÉ

Cet article présente un logiciel expérimental servant à la conception symbolique de filtres numériques d'onde décrits dans l'espace d'état. Ce logiciel repose sur des techniques de transformations structurelles préservant exactement les caractéristiques comportementales des filtres initiaux. Ce logiciel permet en particulier la conception de filtres numériques d'onde dérivés de filtres de référence analogiques en échelle et qui sont traités en détail dans l'article.

ABSTRACT

This paper presents an experimental software tool for the symbolic design of state-space Wave Digital Filters using exact behavior-preserving structural transformations. This software tool supports in particular the design of state-space Wave Digital Filters based on analog LC ladder reference filters which will be discussed in detail.

1. INTRODUCTION

Symbolic methods show to be very attractive for the design of Digital Signal Processing (DSP) algorithms, since they provide the necessary facilities for high-level algorithm developments. In particular, they are well-suited for the design, transformation, analysis, and optimization of (DSP) algorithms. By supporting efficient *hierarchical*, *structural*, and *functional* algorithm descriptions, symbolic methods provide an essential insight in the fundamental properties of the algorithms.

In addition to these features, symbolic methods can advantageously be used in conjunction with *numerical methods* to enhance the overall performance. Especially during repeated algorithm analysis and optimization procedures, they can improve the computation accuracy (reduction of numerical error propagations) and shorten the processing time.

Symbolic methods have been very soon recognized as a valuable approach to complement numerical methods, altough their *generalized* use is (still) limited because of the involved computation complexity. A survey of early works on symbolic methods for the design and analysis of analog networks is given in [1]. The results have then been extended to digital networks, as described for example in [2]. More recently, many papers on the application of symbolic methods to DSP algorithms have been published (see for instance [3]).

This paper describes an experimental software tool for the symbolic design of state-space Wave Digital Filters. This software, called *Symbolic Filter Design tool* (SFD), uses a rule-based approach to perform exact transformations preserving the properties of the filters. SFD supports the design of state-space Wave Digital Filters derived from analog reference filters, i.e. lumped or commensurate distributed lossless analog filters [4, 5]. However, SFD can also be used to obtain the state-space form of Wave Digital Filters (WDFs) which have been directly designed in the digital domain (e.g. Lattice WDFs [6]).

SFD is currently under development and will be used as part of a more general CAD framework described in [7]. In this framework, Wave Digital Filters (WDFs) have been selected for the realization of Infinite Impulse Response (IIR) filters because of their excellent numerical properties, in particular with respect to limit cycles and forced response stability, which are preserved even under looped conditions. In addition, the extensive research on WDFs which has been undertaken over the years has resulted in an extended theory and straightforward design methods [5].

The state-space form has been chosen within the same CAD framework as a uniform and compact representation of the filters, leading to efficient hardware and software implementations (high regularity, maximum parallelism, high throughput, flexibility, etc.) with reduced round-off noise. It turns out that by properly deriving the state-space form, so-called *Essentially Equivalent State-Space* (EESS) WDFs can be obtained which preserve all the properties of the initial reference filter [8].

The present paper will concentrate on the design of WDFs based on *doubly terminated analog LC ladder reference filters* by extending the basic concepts which have been discussed in a former paper [9].

2. FILTER DESIGN FRAMEWORK

The overall filter design framework within which SFD will be used is shown in Figure 1 [7, 8]. Once the specifications are defined, the realization of a given filter begins first by choosing the desired filter class and filter structure. In the context of this paper, the filter class corresponds to WDFs. In a second step, the basic reference filter is obtained using either general (e.g. FILSYN [10]) or specific filter synthesis programs (e.g. FALCON [6]).

SFD appears in the next step to map the reference filter into the digital domain and derive the corresponding EESS WDF form. The obtained filter is then optimized by properly quantizing the coefficients and by scaling the whole filter. All these operations are performed in function of the required hardware/software implementation.

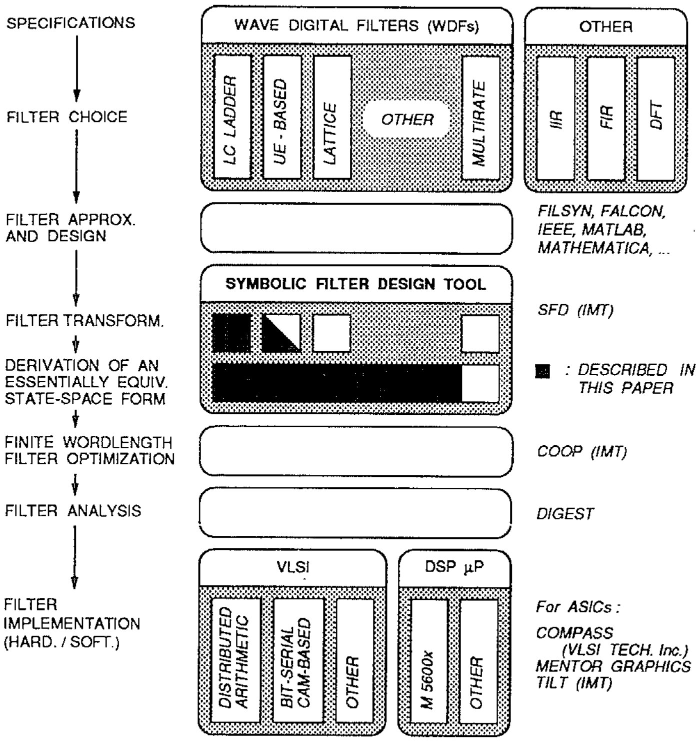

Figure 1 : Overall filter design framework

Next, the resulting filter is verified in detail with the DIGEST program by taking into account all the finite wordlength effects (e.g. round-off noise, limit cycles, etc.) [11]. Finally, the filter can be implemented on ASICs using dedicated processor architectures (e.g. Distributed Arithmetic, for which an extended VLSI cell-library has been developed [8, 14]; Bit-Serial Content-Addressable Memory based architectures [12]; etc.). As an alternative solution, the obtained filter can also be implemented on DSP processors [13].

3. SYMBOLIC FILTER DESIGN TOOL

As shown in Figure 2, SFD can be used in two operational modes for the design of ladder Wave Digital Filters, namely in an automated and in a manual mode.

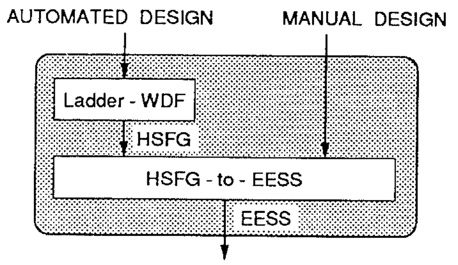

Figure 2 : Operational modes and structure of the SFD tool

3.1 Automated and manual operational modes

The *automated operational mode* provides a fully automated design of state-space WDFs, including some optional optimizations. Starting with an analog reference filter, SFD can generate canonic EESS WDFs leading to compact solutions for low to medium-order filters. For applications requiring higher-order filters, SFD can design extended pipelined filters which are obtained by inserting supplementary state variables to cut down and balance the computational paths of the filter structure. The additional state variables are introduced using Unit Elements (UEs) which are systematically propagated to the optimal locations by applying Kuroda-Levy's transfomation [4].

For the moment, the automated operational mode of SFD does not support the design of generalized ladder structures containing second-order Brune or Darlington C-sections.

In order to alleviate the last limitation and also to provide some more flexibility, SFD can also be used in the *manual operational mode*. In this case, the EESS WDFs are derived from WDF reference filters. Generalized ladder WDFs can be obtained by specifying explicitly the location and the desired implementation variant of the Brune or Darlington C-sections appearing in the filter structure [5].

More generally, the manual mode is helpful for the state-space realization of any kind of WDF obtained for example from lattice WDFs [6] or from cascaded UE reference filters [4].

3.2 Structure of SFD

SFD is subdivided into two distinct modules (Fig. 2). The first module, *Ladder-WDF*, is exclusively used in the automated mode, while the second module, *HSFG-to-EESS*, is employed in both operational modes.

Starting with the input-netlist of a specific analog LC ladder reference filter, *Ladder-WDF* generates an optimized Hierarchical Signal Flow Graph[1] (HSFG) of the corresponding ladder WDF by applying rule-based transformations. In addition to the structural description of the obtained WDF, *Ladder-WDF* prepares also a set of constraints for later use in the coefficient optimization process performed by another program.

The second module, *HSFG-to-EESS*, maps the HSFG of a WDF into an Essentially Equivalent State-Space WDF. This mapping is achieved first by flattening and then by compacting the whole filter structure specified by the HSFG. At the end, the expected state-space WDF is generated [9].

4. AUTOMATED DESIGN OF LADDER WDFS

The remaining part of this paper will essentially concentrate on the automated operational mode of SFD.

The automated design of a ladder WDF begins with the synthesis of a doubly terminated analog LC reference filter fulfilling the given specifications. This reference filter is obtained by conventional means, using for instance the FILSYN program. A typical example of an elliptic 5th-order low-pass reference filter is given in Figure 3.

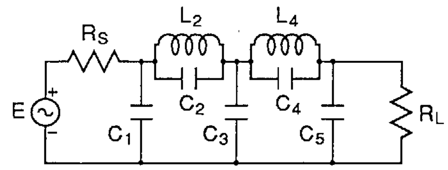

Figure 3 : Example of a 5th-ordre reference filter

4.1 Extraction of the Explicit Filter Structure

The structure of the reference filter has to be translated into an SFD input-netlist[2] which will be processed by the *Ladder-WDF* module. The input-netlist is first parsed in order to verify the syntax and to check the structural correctness of the ladder network. In a further step, the *explicit* description of the filter structure is extracted from the netlist. This can be achieved by

1 In the context of designing Wave Digital Filters, an HSFG corresponds to what is sometimes defined as a Wave Flow Graph
2 The detailed syntax description of the input-netlist and of the HSFG netlist can be found in [9].

noticing that ladder networks are essentially characterized by alternating parallel and serial interconnections [4]. Hence, the explicit description of the filter structure can be derived by recursively applying parallel and serial reduction steps until the full network has been merged into a single one-port element (Fig. 4).

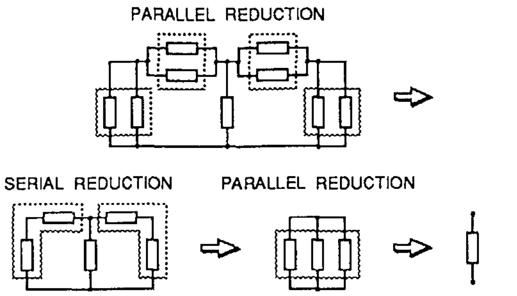

Figure 4 : Reduction of the 5th-order reference filter

At the end, a tree-form can be obtained from the reduction sequence. The leaves of the tree correspond to the basic components of the filter (i.e. access ports, capacitances and inductances), while the internal nodes and the root of the tree indicate how these components are interconnected (Fig. 5).

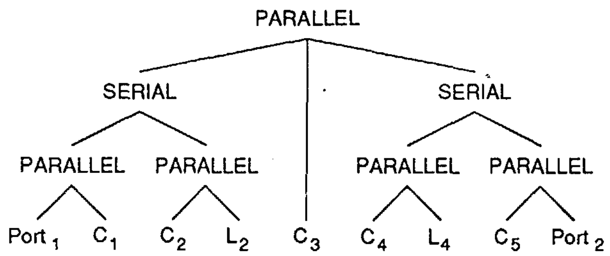

Figure 5 : Explicit description of the 5th-order filter

4.2 Derivation of the Ladder WDF

For simple situations where no additional optimization is needed, the desired WDF is directly obtained from the tree-form:

- The basic filter components (i.e. tree leaves) are mapped to their WDF counter parts (e.g. a capacitance is substituted by a simple delay with an associated port resistance, etc.).
- The parallel and serial interconnections related to the internal nodes of the tree are respectively replaced by parallel and serial *constrained n-port adaptors*. The adaptor port which is oriented towards the root of the tree is reflection-free.
- The parallel or serial interconnection located at the root of the tree is substituted by a parallel or serial *unconstrained n-port adaptor*.

Using the tree-form, all WDF parameters can directly be determined. At the bottom of the tree, the port resistances are specified by the application. By systematically propagating the port resistances towards the tree-root, the adaptor port resistances are gradually fixed. At the end, the achieved WDF is completely defined, and a symbolic Hierarchical Signal Flow Graph can be generated and linked to a cell library describing the WDF elements and adaptors [9].

The proposed design procedure is very simple. In addition, it automatically insures the fulfillment of the basic realizability condition stating that digital filter structures should not contain any delay-free loop.

4.3 Structural Optimizations

SFD provides some optional optimization facilities for the generation of canonic WDFs, for the general equalization of the computational paths, and for the realization of extended pipelined higher-order filters.

Ladder reference filters that realize finite transmission zeros are not canonic in the number of reactive elements and, consequently, the WDFs derived from them contain redundant state variables. The redundancy is due to purely capacitive or purely inductive loops/cutsets existing in the reference filter. Using a simple procedure described in [5], SFD can detect and process these redundancies in order to generate canonic WDFs. Considering again the example of the 5th-order low-pass filter, two state variables should be cancelled to get a canonic WDF. In the proposed solution, the state variables related to capacitances C_2 and C_4 have been suppressed (Fig. 6).

As will be explained in section 4.4, the length of the computational paths of the filter structure has a direct effect on the complexity of the final state-space WDF coefficients. Expressed either in the symbolic form, or in the ultimate optimized finite wordlength representation, the coefficients become more complicated for longer computational paths.

In order to harmonize the global complexity of the state-space coefficients, it can be useful to equalize the computational paths, taking especially care of the longest of them, i.e. the critical paths. Considering again the tree description, which can become irregular for some filter structures, SFD can perform this equalization by properly balancing the tree.

The critical paths of a filter are determined by the distance separating the tree leaves from the tree root. Clearly, this distance will increase with the filter order and will affect the implementation cost of higher-order filters due to the growing coefficient complexity. As a consequence, the implementation cost of higher-order state-space WDFs rises exponentially with the filter-order, not only because of the dimension of the state-space matrix, but also due to the augmented coefficient complexity.

SFD proposes one possible solution to this problem by generating extended pipelined filters which are obtained by inserting supplementary state variables to cut down the critical

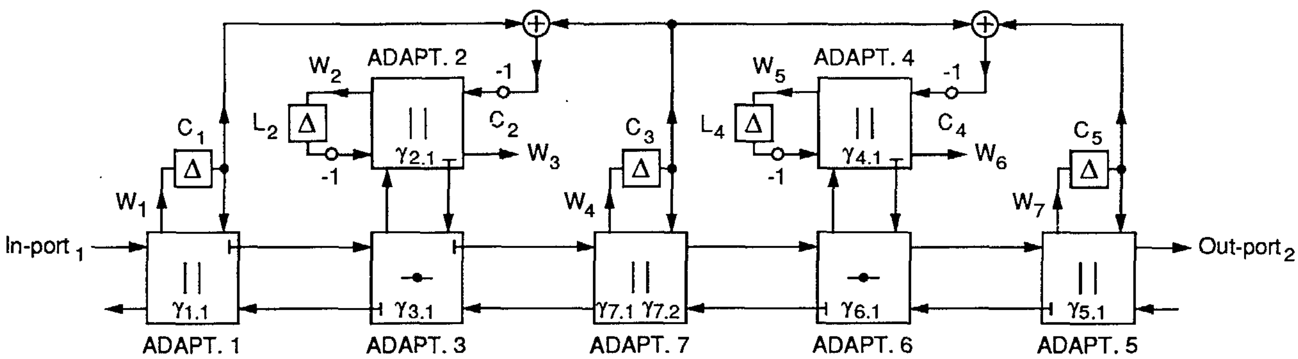

Figure 6 : Canonic WDF derived from the 5th-order ladder reference filter

paths. This can be achieved by introducing Unit Elements at the access ports of the filters and by propagating them to the best locations using Kuroda-Levy's identity. A final retiming step completes the operation [4]. Altough the obtained filters are not canonic, they are characterized by a sparse state matrix with less complex coefficients, leading to competitive implementations for a range of applications.

However, it should be mentioned that for particularly selective high-order filters, other solutions based for example on multirate filtering methods should be considered.

4.4 Derivation of the State-Space WDF

Once the convenient structure of the WDF filter has been defined, the corresponding state-space WDF can be derived with the second module of SFD.

This is achieved in a first step by parsing the HSFG netlist to check its syntactical and semantic correctness (signal fanin/fanout, hierarchical cell interdependences, etc.). The whole filter structure is then flattened to a one-level Signal Flow Graph (SFG) which essentially contains elementary algebraic operations. In a next step, the SFG is analysed to determine the dependency of all the appearing variables and to define a computational precedence graph [15]. The SFG is then compacted by reducing the precedence graph such that all intermediate variables are cancelled. In SFD, the compaction is symbolically solved using algebraic computation methods [16]. At the end, the state-space WDF description is obtained[3]. The resulting state-space coefficients are given as symbolic polynomials depending on the original WDF parameters (i.e. the n-port adaptor coefficients γ_{ij}) :

$$W_1(k+1) = [(\gamma_{11} - \gamma_{21})\cdot(2 - \gamma_{71})\cdot\gamma_{31} - 2\cdot\gamma_{11} + 1] \; W_1(k) + \ldots$$

It should be stressed that the basic properties of the WDFs should be preserved in the state-space form. This is important in order to have a direct equivalence between the original WDFs and the derived state-space WDFs, especially with respect to the finite wordlength effects. Considering in particular the optimization of the filter coefficients, it is mandatory first to quantize the original WDF adaptor coefficients γ_{ij} , and then to evaluate from there the state-space coefficients in full precision [14, 8]. Obviously, since the state-space coefficients are available in a symbolic form, this condition can easily be fulfilled during the later coefficient optimization. In addition, the optimization processing time will also be shortened.

State-space WDFs obtained using this design procedure are considered to have an Essentially Equivalent State-Space (EESS) form[4].

5. CONCLUSION

An experimental software tool for the symbolic design of state-space Wave Digital Filters has been presented. This tool supports in particular the automated design of state-space WDFs derived from analog LC ladder reference filters.

The symbolic processing approach shows to be advantageous, first to perform behaviour-preserving filter transformations, and second to shorten the processing time of computational intensive optimization programs.

The presented software is currently under development and is implemented in Common Lisp [17]. The fundamental functionalities of SFD are so far available. However, the existing optimization procedures have to be completed, in particular for the design of extended pipelined higher-order filters.

6. ACKNOWLEDGEMENTS

This project was supported by the *Swiss Foundation for Research in Microtechnology* (Grant FSRM CS 88/14), as well as by the *Commission for the Promotion of Applied Scientific Research* (Grant CERS C-2014.1).

7. REFERENCES

[1] P. M. Lin, "A Survey of Applications of Symbolic Network Functions", Trans. on Circuit Theory, Vol. CT-20, No. 6, Nov. 1973, pp. 732-737.

[2] K. Singhal and J. Vlach, "Symbolic Analysis of Analog and Digital Circuits", IEEE Trans. on Circuits and Systems, Vol. CAS-24, No. 11, Nov. 1977, pp. 598-609.

[3] IEEE ASSP Magazine, Special Issue on *Computer Aided Signal Processing Tools for Research*, Vol . 7, No. 2, April 1990.

[4] H. Baher, *Synthesis of Electrical Networks*, John Wiley & Sons, Chichester, UK, 1984.

[5] A. Fettweis, "Wave Digital Filters : Theory and Practice", Proc. IEEE, Vol. 74, No. 2, Feb. 1986, pp. 270-327.

[6] L. Gazsi, "Explicit Formulas for Lattice Wave Digital Filters", IEEE Trans. on Circuits and Systems, Vol. CAS-32, No. 1, Jan. 1985, pp. 68-88.

[7] U. Sjöström, I. Defilippis, M. Ansorge, F. Pellandini : "CAD Environment for Digital Filter Design and Implementation", Proc. URSI ISSSE'89, Erlangen, FRG, Sept. 1989, pp. 601-604.

[8] U. Sjöström, *On the Design and VLSI Implementation of Digital Signal Processing Algorithms: an Approach Using Wave Digital State Space Filters and Distributed Arithmetic*, Ph. D. Thesis, University of Neuchâtel, CH, in preparation.

[9] M. Ansorge, U. Sjöström, I. Defilippis, P. Balsiger, and F. Pellandini : "On the Automated Symbolic Design of Wave Digital Filters", Proc. ICASSP '91, Toronto, Canada, May 1991, pp. 1561-1564.

[10] G. Szentirmaï, "FILSYN - A General Purpose Filter Synthesis Program", Proc. IEEE, Vol. 65, No. 10, Oct. 1977, pp. 1443-1458.

[11] L. Claesen, H. J. De Man, and J. Vandewalle, "DIGEST : A Digital Filter Evaluation and Simulation Tool for MOS VLSI Filter Implementations", IEEE J. of Solid-State Circuits, Vol. SC-19, No. 3, June 1984, pp. 414-424.

[12] M. Ansorge, U. Sjöström, I. Defilippis, P. Balsiger and F. Pellandini : "A Flexible Low-Power Digital Signal Processor Based on a Content-Addressable Memory", Proc. EUSIPCO'90, Barcelona, Spain, Sept. 1990, pp. 1399-1402.

[13] P. Balsiger, U. Sjöström, and F. Pellandini, "Digital Signal Processor Implementation of Lattice Wave Digital Filters", Proc. EUSIPCO '90, Barcelona, Spain, Sept. 1990, pp. 1531-1534.

[14] U. Sjöström, I. Defilippis, M. Ansorge, F. Pellandini : "A Methodology for ASIC Implementation of Digital Filters", Proc. GRETSI-89, Juan-Les-Pins, France, June 1989, pp. 797-800.

[15] R. E. Crochiere and A. V. Oppenheim, "Analysis of Linear Digital Networks", Proc. IEEE, Vol. 63, No. 4, April 1975, pp. 581-595.

[16] R. Esser, E. Feldmar, *LISP : Fallstudien mit Anwendungen in der Künstlichen Intelligenz*, Vieweg, Braunschweig, FRG, 1989.

[17] G. L. Steele Jr., *Common Lisp : The Language*, 2nd Ed., Digital Press, Burlington, MA, USA, 1990.

3 As a result of this design procedure, the depth of the precedence graph, given by the computational paths appearing in the filter structure, has a direct effect on the complexity of the state-space coefficients (see section 4.3).

4 EESS WDFs are also characterized by a reduced round-off noise, since data quantization is only performed in front of the delay-element associated to each state variable [14].