System for Rapid Prototyping of Application Specific Signal Processors for ASIC Implementation

Antonio E. de la Serna, Michael A. Soderstrand, and Vojin G. Oklobdzija
Department of Electrical and Computer Engineering
University of California, Davis
Davis, California 95616
dsplab@ece.ucdavis.edu
(916) 752-6800

ABSTRACT

A system for Rapid prototyping of Application Specific Signal Processors (RASSP) is described. This system is used for the design and automatic generation of fixed-coefficient FIR and IIR filters that achieves high sample rates with compact layout. The proposed RASSP system described in this paper features a MATLAB-based design program that generates synthesizable VHDL input for logic synthesis.The filter synthesis portion of the program contains several major innovations over current FIR generation systems, such as FIRGEN. Included are coefficient-by-coefficient optimization of Canonical Signed Digit (CSD) representation and a new minimal-hardware IIR pipelining technique which dramatically increases sampling rates. Logic synthesis is obtained with a structured algorithmic description style, which yields an implementation with improved speed and reduced layout area.

1 INTRODUCTION

High performance digital filters have applications in digital communications systems. Real-time digital filtering demands high sample rates, while the VLSI implementation of digital filters provides practical constraints on the design. While compiler tools exist for low sample rate applications [1, 2, 3], an efficient logic generation system does not exist for compact low power mobile applications of both FIR and IIR digital filters which also require high throughput and hardware minimization.

The design and automatic generation of fixed-coefficient FIR and IIR digital filters can be optimized to reduce the overall hardware requirements, thus also reducing power consumption and in most instances latency. Most low-cost and low-power digital filters use fixed-point arithmetic. After an input-output transfer function of a digital filter is developed, the effects of finite-precision arithmetic must be evaluated in order to determine the shortest possible word length to achieve the desired response. In addition, the hardware implementation may be pipelined in order to increase the maximum sampling rate and the specific arrangement of operators may be conducive to hardware minimization.

Canonical signed digit (CSD) multipliers have been shown to provide a very efficient method for constant fixed-point multiplication by utilizing the redundancy of signed digit code [4]. CSD is a radix-2 signed-digit representation for coefficients for which the permissible digit set is {-1,0,1}. Thus, CSD representation permits subtraction, as well as addition, of shifted data in accomplishing multiplication. The feature of redundancy in this representation allows a coefficient implementation to be selected which in general requires fewer adders/subtractors, and thus yields a faster compact multiplier.

A system for Rapid prototyping of Application Specific Signal Processors (RASSP) is proposed which accepts either desired frequency response characteristics or filter transfer function coefficients and produces synthesizable VHDL input for logic synthesis. At the architecture-determination stage of the design process, the proposed RASSP system moves from the abstract transfer function description of the digital filter to more concrete issues of how the filter will be implemented. This system would employ several methods of achieving fast minimal hardware digital filters, while also being highly flexible for support of ongoing research in digital signal processing and communications.

The RASSP system to be implemented would be an extension of a pipelined digital filter design system [5], and will utilize built-in utilities of MATLAB [1] 4.1 to provide a intuitive, visually appealing graphical user interface. Interactive design and analysis would be easily performed by streamlined pull-down menu selections and pop-up windows. The MATLAB signal processing and control system toolboxes provides many of the filter design algorithms necessary. The design system is to be established on an HP-700 Apollo host in an X Windows environment.

[1] MATLAB© is a high-performance numeric computation and visualization software product of The MathWorks, Inc., 24 Prime Parkway, Natick, MA 01760

2 DIGITAL FILTER DESIGN

A system for developing ASIC implementations of digital filters must not only satisfy performance specifications, but also design constraints. It is critical that digital filter design parameters such as filter length, maximum sample rate, and word length be balanced in order to provide the optimal solution. For fixed-coefficient digital filters, our proposed RASSP system would seek a minimal hardware solution which would satisfy the filter specifications.

For any given FIR digital filter performance, a relationship between filter length and data word size exists [6]. This relationship can be further extended to consider minimized CSD bits for a coefficient set. Thus, if for some fixed data word size the hardware complexity of a digital filter is constrained, then the aggregate number of CSD bits used for the coefficient set must be balanced with the filter length in order to meet performance requirements.

Any increase in hardware capacity can be utilized for either an increase in filter order, an increase in the number of CSD bits, or both. Increasing non-zero CSD bits improves the accuracy of finite precision coefficients. However, if the corresponding infinite precision coefficients produce a filter of some arbitrary length which narrowly meets the desired filter performance, then available hardware capacity would be better utilized for increasing the order of the digital filter to accomodate improved performance — at perhaps a coefficient resolution which is equal to or *less* than that originally. Frequently, hardware complexity can be reduced by increasing filter order rather than increasing the number of CSD bits. This result not only decreases overall hardware requirements, but also generally permits a faster sampling rate.

2.1 MDM Pipelining Technique

Although pipelining FIR filters is very straightforward, IIR digital filters require that the corresponding transfer function accomodate the necessary pipeline delay introduced by the latches in the feedback path [7]. In general, recursive IIR digital filters require less hardware than non-recursive realizations to achieve a given set of filter specifications. However, when pipelining is applied to a digital filter for the purpose of increasing the sampling rate, the degree of hardware augmentation of IIR filters is significantly higher than that of non-recursive counterparts [8]. This is because superfluous poles are introduced, which must then be cancelled by a corresponding cascaded FIR filter section.

We have introduced the canonical *minimum denominator-multiplier* (*MDM*) technique that is based on finding the minimum augmentation under the constraint that stability of the pipelined IIR filter is achieved only through increasing the pipeline delay without adding non-zero denominator coefficients [7]. Thus, the number of denominator multipliers will never be more than the order of the IIR filter being realized. The MDM pipelining technique provides a unifying approach that considers the low-augmentation advantages of clustered look-ahead from the time-domain pipelining technique while providing a worst-case stable pipelined solution in the form of symmetrical scattered look-ahead [7]. This generalized method emphasizes a minimization in the overall computational complexity of the resulting pipelined recursive digital filter.

The MDM pipelining method is an exhaustive search procedure which examines a finite set of pipelined filter realizations and guarantees a stable solution with minimal computational complexity. This method can be easily and efficiently programmed for deriving pipelined recursive digital filter transfer function coefficients on a computer. An algorithm was developed for use with MATLAB in order to obtain the results presented in this work.[2]

The RASSP system proposed would utilize the MDM technique for pipelining IIR filters, which has been shown to consistently yield minimal computational complexity over all other recursive digital filter pipelining methods. In the case where the augmented hardware exceeds that of a linear phase FIR filter designed with the Remez exchange algorithm to meet the desired filter specifications, the latter is utilized.

2.2 Optimized CSD Coefficient Quantization

The proposed RASSP system would efficiently implement fixed-coefficient FIR digital filters using CSD to represent the filter coefficients [9]. The CSD coefficients would be scaled to prevent overflow, thus all partial products will essentially be the input data signal that is either unshifted or shifted towards the LSB. Optimizations of FIR coefficient sets have also been developed that rely on scaling, approximation, and shift-delay redundancy [9, 10]. The result is a drastic reduction in hardware complexity when compared with using standard binary multipliers to realize the filter coefficients. However, these optimizations also result in both an approximation to the filter coefficients and an increase in the noise generated due to roundoff error in the computation process. The modified coefficient set must be verified to satisfy the desired filter response.

[2] This MATLAB algorithm is available by anonymous ftp at ftp.ece.ucdavis.edu in /pub/dsplab/iir_pipelining/MDM.

The coefficients of a digital filter need not have a uniform number of non-zero CSD bits, since coefficient sensitivity will vary [11]. The total number of CSD bits for the entire coefficient set yields a measure of hardware complexity which takes into account filter length and non-uniform coefficient resolution.

3 DIGITAL FILTER ARCHITECTURE

A digital filter is a network of multiply-add-delay operations. Using CSD fixed-point coefficient representation, multiplication can be accomplished for a fixed-coefficient filter by hard-wired shifts, addition, and subtraction. Furthermore, signals that are to be distributed at high fan out should be supported by adequate buffering. This includes the driving system clock signal and the MSB of the input signal replicated in two's complement shifts. Thus, a digital filter can be specified by a fundamental set of buffers, registers, and two's complement adders and subtractors. Optimizations such as that suggested by Hartley [9] add low level specifications on the precedence of addition, subtraction, and delay. Furthermore, data path optimization would yield the most speed for an economical design.

These operations should be organized in a manner which groups partial products of similar relative degree of shift first in order to reduce hardware requirements [3]. In the case where the adder/subtractor tree is not completely balanced, it is best to combine the terms with the greatest degree of shift first.

Partial products would be truncated before being added in order to further minimize computational complexity. This truncation would be performed in a manner which would produce equal truncation errors for all partial products, regardless of shift. This reduces the number of adder bits inefficiently used for the minority of partial products which experience the largest degree of shift, and this also serves to simplify roundoff noise analysis.

The data word length not only determines the width of the data path, but also the complexity of the essential buffer, register, adder/subtractor, and ADC/DAC elements used. Uniform data word length can greatly simplify the synthesis of a digital filter, yet hardware can be reduced significantly by optimizing the word-length for each signal developed [12]. This however results in a system for which fixed-point performance must be verified exclusively by simulation, rather than analytical methods. The performance measure which is commonly used for this is SQNR, or Signal to Quantization Noise Ratio.

Before trying to optimize the word length of signals in the algorithm, the RASSP system needs to verify performance for a minimized uniform word length. Following this, a search of further-reduced word length combinations for the signals of the digital filter system would be performed as prescribed by Sung [12]. This permits a bit reduction of operators beyond that of the uniform CSD partial product truncations.

4 STRUCTURAL DESCRIPTION METHODOLOGY

Digital filters are charaterized by a very regular and heirarchical structure. This is amenable to an algorithmic design style which could be easily programmed for automatic silicon compilation. A fixed-coefficient filtering function is implemented in an optimally efficient way with an architecture obtained from a fully parallel mapping of its signal flow graph. This is because coefficients are fixed leading to a drastic simplification of multipliers in fixed-point arithmetic[2].

The state-space description of a filter completely describes the multiply-add-delay construction of the desired implementation. The state-space description for transposed direct form can be derived from the transfer function coefficients, which can be further decomposed into additions and subtractions of shifted binary partial products. However, a further enhancement in description must be made in order to specify the decomposition of multiplications into a series of additions of binary shifts, as well as the operator arrangement and signal word length which minimizes the hardware implementation. An input-output structural sequence can be constructed from this in order to completely define the resulting hardware implementation.

In order to support retargetable technological mapping, behavioral circuit descriptions should be established for this finite set of hardware elements. Coding of this behavioral description is critical in order to ensure minimal hardware construction of the overall filter, which would be described structurally.

MATLAB contains many functions that are useful in digital filter design and simulation, as well as providing a programming language with which to develop applications. Several additional tools have been developed in MATLAB that further support pipelined CSD filter design. It is proposed that a VHDL code generator be developed in MATLAB that follows the guidelines of an algorithm structural design style described above.

A VHDL code generator would start with a finite set of behavioral descriptions for the fundamental buffer, delay, add, and subtract elements stored in ascii files. The design system would then

methodically construct a structural description of the digital filter system based on the considerations above which is written to an output ascii file. This VHDL file could then be used with a logic synthesis tool, such as Synopsys, to generate a VLSI layout or an FPGA bit stream.

By imposing a modular hierarchical methodology in specifying the hardware description, a regular compact layou can be attained. Logic synthesis based on a structured algorithmic design style has been demonstrated to have speed and layout area advantages over utilizing "blind" behavioral models [13]. As much as 25% increase in speed and 8% reduction in area have been yielded in such experiments. Thus, an intelligent logic synthesis tool applied to signal processing applications like digital filtering has tremendous advantages.

5 CONCLUSION

A RASSP system was described which will be used to develop fast digital filters for which hardware implementation is minimized. It is to be a practical extension of a pipelined digital filter design system, and will developed on an HP-700 Apollo workstation in an Xwindows environment. This system is to complement ongoing research in digital signal processing and communications, and will implement hardware specifications for digital filters using existing methods while also permitting experimentation with new design methodologies and filter structures.

References

[1] R. Jain, P. Yang, and T. Yoshino, "FIRGEN: a computer-aided design system for high performance FIR filter integrated circuits," *IEEE Transactions on Signal Processing*, vol. 39, pp. 1655–1668, July 1991.

[2] F. Balestro, A. Chianale, G. Privat, *et al.*, "Design of digital filters for advanced telecommunications ASICs using a special-purpose silicon compiler," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1047–1055, July 1991.

[3] R. Hartley, P. Corbett, P. Jacob, and S. Karr, "A high speed FIR filter designed by a compiler," in *Proceedings IEEE Custom Integrated Circuits Conference*, (San Diego, CA), pp. 20.2/1–4, May 1989.

[4] K. Hwang, *Computer Arithmetic : Principles, Architecture, and Design.* New York: Wiley, 1979.

[5] E. Q. Wong, A. E. de la Serna, and M. A. Soderstrand, "Pipelined digital filter design system," in *MATLAB Conference* (K. Tuttle and C. Moler, eds.), (Natick, MA 01760), The Math Works, Inc., Oct. 1993.

[6] D. Kodek and K. Steiglitz, "Filter-length word-length tradeoffs in FIR digital filter design," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 739–44, Dec. 1980.

[7] M. A. Soderstrand and A. E. de la Serna, "Minimum denominator-multiplier pipelined recursive digital filters." Accepted for publication in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing.*

[8] A. de la Serna, "Stability of time-domain pipelined IIR digital filters," Master's thesis, University of California, Davis, CA, 95616, September 1993.

[9] R. Hartley, "Optimization of canonic signed digit multipliers for filter design," in *Proceedings IEEE Internationl Symposium on Circuits and Systems*, (Singapore), pp. 1992–1995, June 1991.

[10] D. Kodek and K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital filters," *IEEE Transactions on Circuits and Systems*, vol. 28, pp. 28–32, Jan. 1981.

[11] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 1044–7, July 1989.

[12] W. Sung and K.-I. Kum, "Word-length determination and scaling software for a signal flow block diagram," in *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, (Adelaide, Australia), Apr. 1994.

[13] V. Oklobdzija, "An algorithmic and novel design of a leading zero detector circuit: comparison with logic synthesis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, pp. 124–8, Mar. 1994.