An Optimal Structure for Implementation of Digital Filters

S. Rahmanian[1], E. Rahmani[1], A. Nasiri Avanaki and S. M. Fakhraie[1]
Silicon Intelligence and VLSI Signal Processing Laboratory[1],
School of ECE, University of Tehran, Tehran, Iran
{s.rahmanian, nasiri, e.rahmani}@ece.ut.ac.ir, fakhraie@ut.ac.ir

***Abstract*-In this paper, different structures for an elliptic filter with fixed point arithmetic are implemented and compared. The filter must be quantized for hardware implementation. This quantization is done in two steps. First the coefficients of filter are quantized and then the accuracy of internal nodes are limited. According to the simulation results, lattice and DF2-parallel structures have minimal sensitivity to coefficient quantization. Also, the area (gate count) that each of the structures occupy on the chip are computed. We show that overall, the DF1-parallel structure is the optimal structure for hardware implementation that requires minimal chip area at a reasonable precision.**

Keywords: round-off noise, bit-true modeling digital filter implementation.

I. Introduction & Background

A particular linear time-invariant discrete-time system can be implemented by a variety of computational structure. One motivation for considering alternatives to the simple direct form structures is that different structures that are theoretically equivalent may behave differently when implemented with finite numerical precision.

We are almost always interested in implementations that require the least amount of hardware or software complexity. However, we cannot find the optimal structure on this criterion alone, since some of the minimal hardware structures are very sensitive to quantization noise (the effect of finite-precision computations is modeled with this noise).

Much work has been devoted to estimation of quantization noise and its reduction. One of the appropriate structures in finite word length implementation is δDF2t. [1] and [2] modified δDF2t second order section in which the δ at different branches are separately optimized to suppress the round-off noise further. In [4], an efficient infinite impulse response (IIR) structure is produced via spectral transformation of an appropriate finite impulse response (FIR) prototype design. Hence, the only coefficient quantization required, is for the original FIR coefficients, which are relatively insensitive. The round-off noise for the fixed-point implementations is also quite low compared to conventional IIR designs. However, to our best knowledge, no work in the literature addressed the optimal structure in insensitivity to quantization and the minimal hardware required for implementation, both at the same time.

In this paper different structures of a digital filter is investigated and the optimal structure with the minimal hardware required for implementation and the minimum round-off noise is introduced.

The rest of paper is organized as follows. In Section II, different structures for implementation of a digital IIR filter are reviewed and our experimental setup is given. In Section III, hardware implementation process is explained. In the next section, bit-true modeling is described. HDL modeling is explained in Section V. Before concluding the article in Section VII, our simulation results and the optimal structure for fixed-point implementation are presented in Section VI.

II. Different Filter Types and Structures

The direct forms (DF) are the simplest structure for implementation of digital filters [6]. DF2 can save up to 50% in the number of required delay elements.

Cascade and parallel forms consist of second order direct form sections. In both cases, each pair of complex conjugate poles is realized independently of all other poles. A DF transposed (DFt) structure is obtained by changing the direction of all branches and the input and output signals position, starting from a DF structure. In cascade and parallel forms, one can use DFt second order sections (a.k.a. biquad blocks).In this paper, an IIR low-pass filter used for voice filtering is studied. In this application, only the amplitude characteristic of the filter is important. The filter parameters are as follows. Attenuation ripple at pass-band Rp =1 dB, attenuation at stop-band Rs =60 dB, sampling frequency Fs =44 KHz, pass-band frequency Fpass =3.125 KHz and stop-band frequency Fstop = 4 KHz.

This filter is designed by Chebyshev, Butterworth and elliptic formulas. The order of the elliptic filter is 6 and the order of Butterworth and Chebyshev are 12 and 30 respectively. A FIR formula (such as Parks-McClellan) could provide a better phase response, at a much higher order, which is neither acceptable nor necessary in our application. It is a well-known fact that the elliptic formula gives the minimum order filter (i.e., the minimum number of delays, adders and multipliers) when amplitude characteristic of the filter matters only. For this reason, we use elliptic formula through the rest

1-4244-0765-6/06/$20.00 ©2006 IEEE

of the paper and we find the optimal structure for its fixed-point implementation.

III. Hardware Implementation Process

Generally, the DSP systems are implemented by two major number representations: fixed point and floating point. Floating point arithmetic offers high precision and wide dynamic range and is used when no loss in precision is tolerated (e.g., in simulation of ideal systems). In the real world signal processing applications, where low-cost and low-power solutions are sought, the fixed-point arithmetic is ubiquitously in use.

For fixed point hardware implementation, the bit true model of the filter must be extracted first. We use Simulink/Matlab for extraction of the model, which specifies the required coefficient and intermediate word lengths for filter implementation. Based on these results, HDL model of filter using one of hardware description language must be devised. Finally, the HDL model is synthesized on ASIC or FPGA platforms. This process is described in the following.

IV. Bit True Modeling

A. Input Signal Quantization

First, we quantize the input signal and determine the suitable word length, depending on the application.

If uniform amplitude quantization is performed, only considering the effect of quantization, SNR is increased by 6 dB for each additional bit [6].

We consider 10-bit accuracy for the input signal. Thus, the input SNR is about 60 dB. Since our filter has unit gain at pass band, the desired output SNR is 60 dB. That is, if using fixed-point arithmetic introduces no additional noise.

B. SNR Calculation Method

Our method of SNR measurement is illustrated in Fig. 1. First, sample input signal is injected to the both of ideal (floating-point) and quantized (coefficient and intermediate value) models. Then, the difference between two outputs is calculated. The ratio of quantization noise (the difference between the outputs of the two models) power and the power of the output of the ideal model is considered as the SNR value.

As an approximation of voice signal, we used sum of several sine waves with different frequencies.

C. Coefficient Quantization

Quantization of filter coefficients causes deviation of poles and zeros from their original (designed) positions.

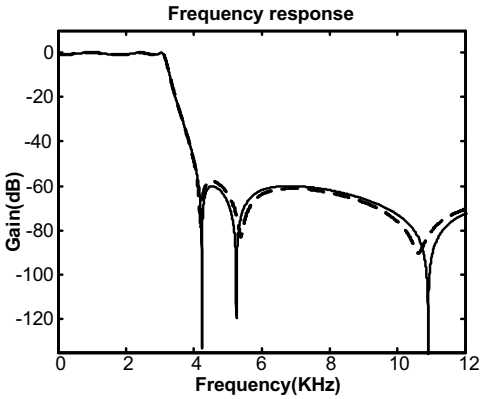

Fig. 2: Comparison of 14-bit quantized (dashed) and the ideal (solid) filters.

Therefore, the frequency response of the quantized filter is changed with respect to the ideal filter. We assume a filter with 64-bit floating point coefficients as our ideal filter. Generally, the closer the poles are to each other, the greater is the deviation. Since a biquad block has only one pair of complex conjugate poles far from each other, parallel or cascade combination of biquad blocks are less sensitive to quantization.

Hence, using parallel and cascade structures we achieve the desired frequency response at a smaller word length. Fig. 2 compares the frequency response of the 14-bit quantized coefficient and the ideal elliptic filters.

To calculate the suitable word length, the SNR of quantized filter is calculated for various word lengths and different structures. By increasing coefficient word length, the output SNR will be increased until it reaches the input SNR. Fig. 3 shows the output SNR versus coefficient word length for parallel, cascade and lattice structures.

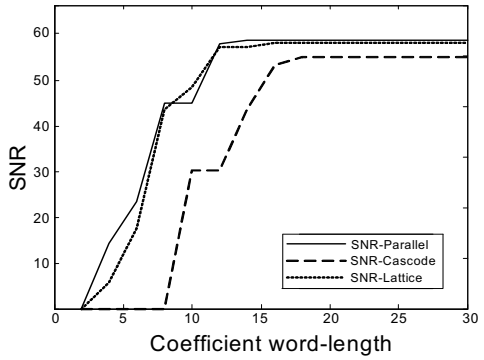

Fig. 3. Output SNR vs. coefficient word length for different structures.

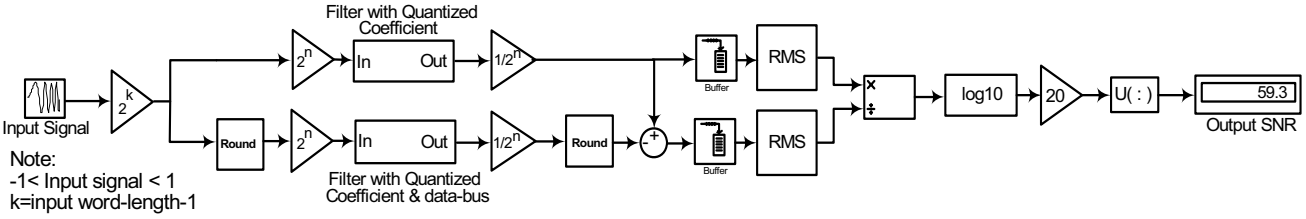

Fig. 1. SNR calculation method

D. Intermediate Word Length Determination

1) Swing Measurement

In finite word length implementation of digital filters, if the word length of the intermediate values is not selected properly, overflow can occur. This causes signal degradation and parasitic oscillations **Error! Reference source not found.**. Using the safe scaling method for intermediate value word length determination, overflow can be avoided. In this method, impulse responses from input to all internal nodes are calculated and the intermediate word lengths (WL) are determined using the following formula**Error! Reference source not found.**:

$$Intermediate\,WL = \log_2 \sum_{n=0}^{\infty} |h_i(n)| + Input\,WL,$$

where $h_i(n)$ is the impulse response from the input to node i. Usually, the maximum calculated word length is used for all of nodes. This method, however, is too conservative[7] (i.e., smaller word length can be used with no significant signal degradation). In practice, the suitable word length can be obtained by monitoring swings at each node:

$$Intermediate\,WL = \log_2(S_i) + Input\,WL,$$

where S_i is the swing of node i. We employed this method for determination of intermediate word lengths.

Table 1 compares the safe scaling with simulation results for various input data such as voice and multi-tone sine signals.

TABLE 1: SAFE SCALING VS. PRACTICAL METHOD.

	Simulation		Safe scaling
	Voice	Multi tone	
Additional Word-length	4 bit	6 bit	8 bit

2) Precision adjustment

In this stage of bit true modeling, the multipliers outputs are quantized using round blocks. Fig. 4 shows the filter with quantized intermediate value.

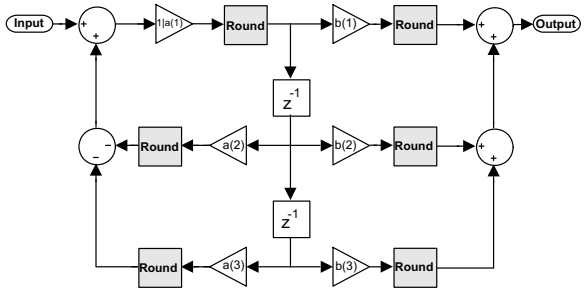

Fig. 4. Model of quantized filter.

Because of the rounding operations, the precision is reduced as compared to the floating-point model. This loss of precision reduces the SNR at the output. To compensate for this effect, the input signal is multiplied by $G = 2^n$ right after input quantization. This way, the input signal is shifted to left by n bits (n zeros are added to the right side of the fixed-point representation). Hence, the minimum possible number in intermediate calculation, and therefore, the precision and the output SNR is increased. After a certain limit, however, increasing n has no effect on the output SNR. Fig. 5 plots the output SNR versus the number of extra bits for three different types of input signals.

V. HDL MODELING AND SYNTHESIS

We assume fully combinational hardware implementation for the filter. In this approach, computational modules such as full-adders and multipliers are all implemented in parallel. Because of the hardware design constraints (area and delay), in this work, array multipliers and carry look-ahead adders are selected for multiplication and addition respectively.

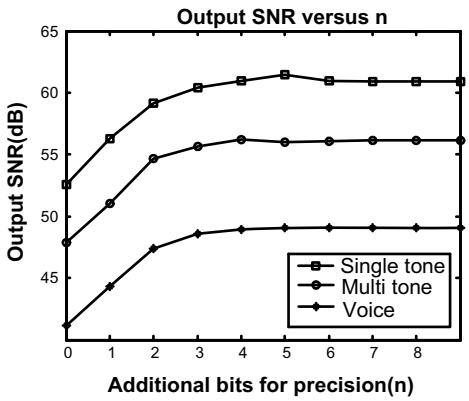

Fig. 5. Output SNR vs. the number of extra bits.

For chip area estimation, one multiplier and one adder are described in Verilog HDL and synthesized for 0.35 μm CMOS ASIC library. Based on synthesis results the chip areas for different structures are estimated.

VI. EXPERIMENTAL RESULTS

In this work, the bit true model parameters for various structure of filter are extracted and then based on this parameter, gate count estimation for filter hardware implementation are accomplished. Proper coefficient word length for different structure as well as additional intermediate word length required for overflow prevention and precision adjustment is given in Table 2, in which gate counts and area estimations are also listed for various structures.

VII. CONCLUSIONS

Bit true models for different structures of a sample elliptic filter are extracted. Based on these models, the required number of gates for various hardware implementations are estimated.

Lattice and parallel DF2 show small sensitivity to coefficient quantization. However, cascade and parallel DF1 require the minimum intermediate word length for implementation.

Regarding the gate count, parallel DF1 is the best structure for hardware efficient implementation and has the lowest cost.

DF2 implementation occupies more chip area than the other structures and costs the most. That is because signals swing widely in this structure, and its implementation requires a long word length.

TABLE 2. BIT TRUE MODEL PARAMETER AND ESTIMATED GATE COUNT FOR DIFFERENT FILTER STRUCTURE

Structure	Proper coefficient word-length	Additional word-length for prevention of overflow	Additional word-length for precision adjustment	Total word-length of intermediate value	Area(gates)
DF1	27	0	16	26	35405
DF1 SOS	15	0	6	16	15078
DF1 Parallel	15	1	5	16	12407
DF1t SOS	15	10	3	23	21675
DF2	27	16	1	27	36727
DF2 SOS	15	2	8	20	18848
DF2 Parallel	12	6	8	24	15029
DF2t SOS	15	2	7	19	17906
Df2t	22	4	11	25	27889
Lattice	12	12	2	24	22002

ACKNOWLEDGEMENTS

The authors would like to thank the Iranian Telecommunication Research Center (ITRC) for supporting this work.

REFERENCES

[1] N. Wong, T. S. Ng, “Improved roundoff noise performance in a direct-form IIR filter using a modified delta operator,” in *Proc. International Symposium on Circuits and Systems*, ISCAS 2001, pp. 773 – 776, vol. 2, May 2001.

[2] N. Wong, T. S. Ng, “Roundoff noise minimization in a modified direct-form delta operator IIR structure,” *IEEE Transaction on Circuits and Systems*, vol. 47, no. 12, pp. 1533 – 1536, December 2000.

[3] G. Li, Z.X Zhao and J.X Hao, “A generalized direct-form II transposed structure for IIR filter implementation with minimal roundoff noise gain,” in *Proc. International Symposium on Circuits and Systems, ISCAS 2003,* vol.4, pp. IV-217 - IV-220, May 2003.

[4] J.A. Lopez, C. Carreras, G. Caffarena and O. Nieto-Taladriz, “Fast characterization of the noise bounds derived from coefficient and signal quantization,” in *Proc. International Symposium on Circuits and Systems, ISCAS 2003,* vol.4, pp. IV-309 - IV-312, May 2003.

[5]

[6] A. V. Openheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing.* Prentice Hall, 1999.

[7] N. Sedaghati, S. Rahmanian and S. M. Fakhraie, “Hardware implementation analysis for digital filters,” in *Proc. Iranian Conference on Electrical Engineering, ICEE2006*, May 2006.