

# An optimal structure for implementation of digital filters

S. Rahmanian<sup>a)</sup> and S. M. Fakhraie

Silicon Intelligence and VLSI Signal Processing Laboratory, School of ECE,  
University of Tehran, Tehran 14395–515, Iran

a) [s.rahmanian@ece.ut.ac.ir](mailto:s.rahmanian@ece.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 the filter are quantized and then the minimum required accuracy of the internal nodes is determined. According to the simulation results, lattice and DFII-parallel structures have minimal sensitivity to coefficient quantization. Also, the chip areas (i.e. gate counts) of different structures are computed. We show that overall, the DFI-parallel structure is the optimal structure for hardware implementation and requires minimal chip area at the needed precision.

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

**Classification:** Integrated circuits

## References

- [1] N. Wong and 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 and T. S. Ng, “Roundoff noise minimization in a modified direct-form delta operator IIR structure,” *IEEE Trans Circuits Syst.II*, vol. 47, no. 12, pp. 1533–1536, Dec. 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] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice Hall, 1999.

## 1 Introduction

A particular linear time-invariant discrete-time system can be implemented by a variety of computational structures. 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.

Much work has been devoted to estimation of quantization noise and its reduction. One of the appropriate structures in finite word-length implementation is  $\delta$ DFII<sub>t</sub>. [1] And [2] modified  $\delta$ DFII<sub>t</sub> second order section in which the  $\delta$  at different branches is 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. 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 are investigated and the optimal structure with the minimal required hardware for implementation and the minimum round-off noise is introduced.

The rest of this 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, our simulation results and the optimal structure for fixed-point implementation are presented in Section V. Finally, Section VI concludes this paper.

## 2 Different filter types and structures

The direct forms (DF) are the simplest structures among others, for implementation of digital filters [5]. 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 signal positions, starting from a DF structure. 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.

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 characteristics of the filter matters only. For this reason, we use elliptic formula through the rest of the paper and we find the optimal structure for its fixed-point implementation.

### 3 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 must be devised. Finally, the HDL model is synthesized on ASIC or FPGA platforms. This process is described in the following.

### 4 Bit-true modeling

#### 4.1 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 power (the difference between the outputs of the two models) and the power of the output of the ideal model is considered as the SNR value. 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.

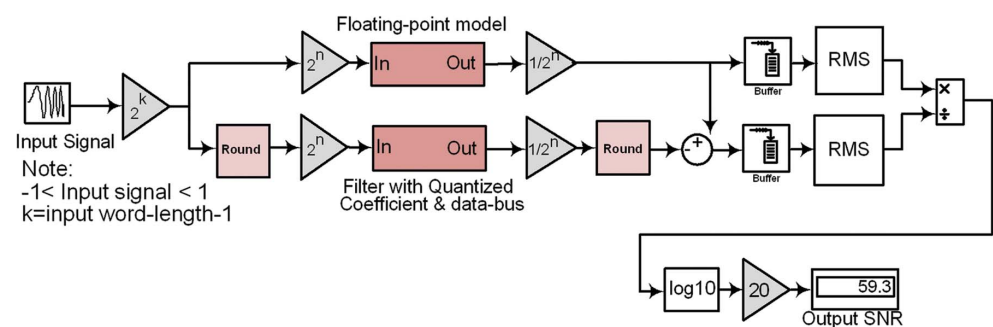


Fig. 1. SNR calculation method.

#### 4.2 Coefficient quantization

Quantization of filter coefficients causes deviation of poles and zeros from their original (designed) positions. 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. Some of the structures are less sensitive to coefficient quantization than others. Similarly,

because different structures have different quantization noise sources and because these noise sources are filtered in different ways by the system. 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. 2 shows the output SNR versus coefficient word-length for DFI, cascade and parallel structures. Generally, the closer the poles are to each other, the greater is the deviation [5]. In parallel and cascade structures each pair of complex-conjugate poles is realized independently of all other poles. For this reason, parallel or cascade combination of biquad blocks are less sensitive to quantization and using these structures, we achieve the desired frequency response at a smaller word-length.

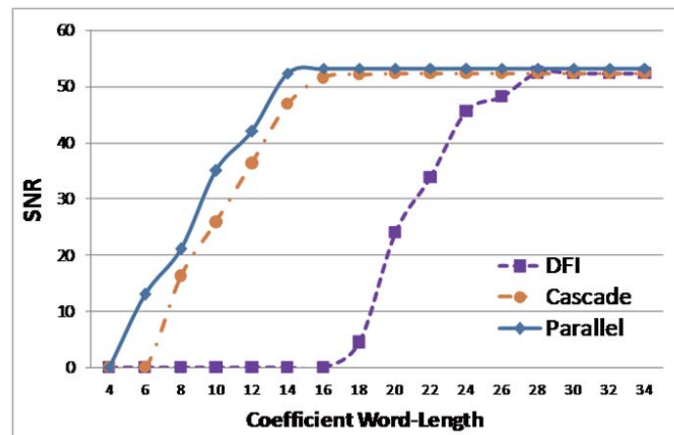


Fig. 2. Output SNR vs. coefficient word-length for different structures.

### 4.3 Intermediate word-length determination

#### 4.3.1 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.

The suitable word-length can be obtained by monitoring swings at each node:

$$\text{Intermediate } WL = \log_2(S_i) + \text{Input } WL \quad (1)$$

where  $S_i$  is the maximum swing of node  $i$ . Usually, the maximum calculated word-length is used for all of nodes.

#### 4.3.2 Precision adjustment

In this stage of bit-true modeling, the multipliers outputs are quantized using rounding blocks. Because of the rounding operations, the precision is reduced as compared to the floating-point model. This loss of precision reduces the

**Table I.** Bit true model parameter and estimated gate count for different filter structures.

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)
DFI	27	0	16	26	35405
DFI Cascade	15	0	6	16	15078
DFI Parallel	15	1	5	16	12407
DFIt Cascade	15	10	3	23	21675
DFII	27	16	1	27	36727
DFII Cascade	15	2	8	20	18848
DFII Parallel	12	6	8	24	15029
DFIIt Cascade	15	2	7	19	17906
DFIIt	22	4	11	25	27889
Lattice	12	12	2	24	22002

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 calculations, and therefore, the precision and the output SNR is increased. After a certain limit, however, increasing  $n$  has no effect on the output SNR.

## 5 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.

For chip area estimation, one multiplier and one adder are described in Verilog HDL and synthesized for  $0.35\mu\text{m}$  CMOS ASIC library. Based on synthesis results the chip areas for different structures are estimated. Proper coefficient word-length for different structure as well as additional intermediate word-length required for overflow prevention and precision adjustment are given in Table I, in which gate counts and area estimations are also listed for various structures.

## 6 Conclusion

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 DFII show small sensitivity to coefficient quantization. However, cascade and parallel DFI require the minimum intermediate word-length for implementation.

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

DFII 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.