ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

ROUNDOFF-NOISE SHAPING IN FILTER DESIGN

G.A. Constantinides, P.Y.K. Cheung, W. Luk

Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, UK
Department of Computing, Imperial College, London SW7 2BZ, UK

ABSTRACT

This paper presents a technique for the spectral shaping of roundoff noise in fixed-point implementations of digital filters. An automated feasibility test is introduced, in order to decide whether a given filter realisation meets user-specified constraints on the roundoff noise power spectrum. This feasibility test is used by an algorithm for optimization of individual signal widths within a filter structure. Some results are presented, illustrating how the optimization produces filters closely meeting the specification, leading to significant improvements in implementation area.

1. INTRODUCTION

This paper presents a technique for computer aided design of fixed-point linear time-invariant digital filters. The technique allows the user to specify bounds on the power spectrum of the roundoff noise introduced by the fixed-point realisation. Our technique has been implemented within the Synoptix high-level synthesis system which, given a Simulink [1] signal flow graph (SFG), and spectral bounds on roundoff noise for each of the outputs, will provide an area-optimized filter implementation in a hardware description language.

By using different wordlengths at different points within the implemented structure, our synthesis tool is able to exploit the various paths through the filter structure (with different spectral properties) in order to shape the roundoff noise towards the specification. The aim of the optimization tool is to minimize the implementation area of the filter, while meeting the user-specified spectral constraints. At the core of this optimization is a feasibility test, to check whether any given realisation meets the spectral constraints. We present such a test in this paper, which will ensure that a noise spectrum is bounded by the specification for all frequencies, given some constraints on the allowable type of specification. In addition, we demonstrate the use of root moments to alleviate the algorithmic accuracy problems associated with the successive deflation of a polynomial used in this feasibility test. This paper also introduces a heuristic used to solve the problem of selecting signal widths within the filter structure, and demonstrates how the approach described allows the error power to closely meet the specification, resulting in significant area savings over a uniform signal width structure.

This work was supported by Hewlett-Packard Laboratories and the Engineering and Physical Sciences Research Council. The authors wish to acknowledge helpful comments from Dr. T. Stathaki.

2. BACKGROUND

The effects of using finite register length in fixed-point systems have been studied for some time. Oppenheim and Weinstein [2] and Liu [3] provide standard models for quantization errors and error propagation through linear time-invariant (LTI) systems. This early work tended to assume a fixed-wordlength computational machine, which leads to the assumption of a single uniform signal width. However there is scope for improvement when we have the freedom to design special purpose hardware to implement DSP functions. In [4] alternative noise models are presented, more appropriate to structures with non-uniform signal widths.

To our knowledge, the problem of optimizing individual signal widths within an implementation structure in order to meet upper bounds on roundoff noise power spectrum has not been addressed in any published work. However, the use of other measures of performance, such as optimization of signal widths in order to achieve a given Signal to Noise Ratio (SNR), has achieved recent attention [5, 6, 7, 8].

Most recent work on addressing the problem of floating-point to fixed-point conversion has considered the issue from a software profiling perspective [5, 6], sometimes in combination with a 'format propagation' and/or user interaction [7, 8]. While these approaches allow the use of nonlinear and time-varying components, the quality of the resulting SNR estimates depend on the input signals used for simulation, or the user help given. In addition, large run-times are necessary for the simulations that form the basis of the optimization routines [5]. In contrast, our approach uses an analytical method which, while restricting the problem domain to LTI systems, does not suffer from these drawbacks. Moreover it allows spectral noise-shaping of the roundoff noise introduced.

Useful work on manipulation of polynomials through the use of so-called 'root moments' has recently appeared in the DSP literature [9]. In this work, root moments are used to determine the minimum phase component of a signal, to form a stability test for LTI systems, and to detect abrupt changes in a signal. In the present paper, we use these root-moment techniques to mitigate the effects of finite wordlength representation when performing deflation of a polynomial during root-finding.

3. NOISE MODEL

The input to our Synoptix tool is a SFG representing the desired filter structure to be implemented. In order to predict the noise PSD, it is necessary to construct a model of how arithmetic roundoff or truncation noise is introduced and propagated through the SFG.

At each point p in the SFG where a signal is truncated, a noise source with variance σ_p^2 is constructed. These sources are assumed

0-7803-5482-6/99/$10.00 ©2000 IEEE

to be uncorrelated with each other, and to have a white PSD (except for truncation of two's complement representation, which is assumed to be white noise together with a Dirac delta at zero frequency to model the truncation bias). The exact model for σ_p^2 used in Synoptix is presented in [4]. The noise PSD at an output of the SFG may then be estimated using the transfer functions from each point p to the output, $H_p(z)$.

$$N(z) = \sum_p \sigma_p^2 |H_p(z)|^2 \quad z = e^{j\theta}$$

Synoptix keeps a representation of $H_p(z)H_p(1/z)$ for each H_p, allowing $N(z)$ to be constructed on the fly for $z = e^{j\theta}$ as σ_p^2 values change during optimization.

4. FEASIBILITY TESTING

Once the noise PSD has been predicted, as above, it is necessary to test whether this satisfies the user-specified constraints on error PSD. Specifically, the feasibility testing algorithm in Synoptix tests whether the following condition holds, given a real constraint function $c(\theta)$.

$$N(e^{j\theta}) \leq c(\theta) \quad \forall\theta$$

We approach this problem by firstly placing a constraint on the type of function that $c(\theta)$ may be. If we restrict $c(\theta) = |C(e^{j\theta})|^2$ with C real and rational in $z = e^{j\theta}$, then our technique may be applied. We do not believe this to be an unreasonable restriction since standard digital filter design packages may be used to construct an appropriate $C(z)$ of a suitable order, close to a given $c(\theta)$.

If $N(e^{j0}) \leq |C(e^{j0})|^2$ and $N(e^{j\pi}) \leq |C(e^{j\pi})|^2$ then as long as neither $N(z)$ nor $C(z)$ have poles on the unit circle, it holds from their smoothness that

$$(\exists\theta_0 : N(e^{j\theta_0}) > |C(e^{j\theta_0})|^2) \Rightarrow$$
$$(\exists\theta_1 < \theta_0, \theta_2 > \theta_0 : N(e^{j\theta_i}) - |C(e^{j\theta_i})|^2 = 0 \quad i = 1, 2)$$

Therefore by viewing the problem in this light, we have reduced the number of points of interest for the feasibility test from the entire unit circle to the finite set of points on the unit circle where $N(e^{j\theta}) - |C(e^{j\theta})|^2 = 0$. In order to solve this equation, only the numerator polynomial $F(z)$ need be considered.

The next step of the feasibility test is to find those roots of $F(z)$ that lie on the unit circle. Due to finite wordlength effects in the algorithm execution, it is necessary to consider not just roots where $|z| = 1$ but those nearby, since the limited accuracy could lead to the movement of roots. This may be done in a number of ways. Firstly, we may recognize that since $F(z)$ has real coefficients, if $z = z_0$ is a root, so is $z = z_0^*$. In addition, $F(z)$ has symmetric coefficients about z^0, hence $z = 1/z_0$ and $z = 1/z_0^*$ are also roots. A straight-forward solution technique is to use an iterative method of polynomial solving, such as Laguerre's method [10], for finding an initial solution z_0 which may then be classified as (a) $z_0 \approx 1$ or $z_0 \approx \pi$ (b) $|z_0| \approx 1, Im(z_0) \neq 0$ (c) $Im(z_0) \approx 0, |z_0| \neq 1$ or (d) $Im(z_0) \neq 0, |z_0| \neq 1$. Before finding the next solution, the polynomial F(z) may then be deflated by the appropriate factor (a) $(z - z_0)$, (b) $(z - z_0)(z - z_0^*)$, (c) $(z - z_0)(z - 1/z_0)$ or (d) $(z - z_0)(z - z_0^*)(z - 1/z_0)(z - 1/z_0^*)$. Any solutions lying in a small annulus around $|z| = 1$, the region of interest, are retained and stored as *possible* unit-magnitude roots of $F(z)$. This small annulus is used in order not to reject solutions with $|z| \approx 1$ which may arise due to finite wordlength effects in the algorithm execution.

This procedure has been implemented within Synoptix, and works well for systems where $F(z)$ has a relatively small order. However if $F(z)$ is of large order, inaccuracies due to polynomial deflation using finite-wordlength arithmetic can easily build up during execution. At each deflation step, these inaccuracies may grow, so that for the final roots extracted, they may significantly change the root locations. For this reason, we present here an alternative algorithmic approach based on root moments and Newton identities [9]. The key point to recognise is that the only roots of $F(z)$ which must be located on the complex plane are those roots with $|z| \approx 1$. We proceed by factorizing $F(z)$ into $F(z) = F_1(z)F_o(z)$ where $F_1(z)$ contains those roots of $F(z)$ within the annulus of interest and $F_o(z)$ contains all other roots. Thus only $F_1(z)$, typically of much smaller order than $F(z)$, need be solved using the deflation-based technique in the previous paragraph. This factorization may be performed efficiently using root moment techniques.

The m'th root moment of a polynomial of order n with roots $\{r_j\}$ is defined as [9]

$$S_m = r_1^m + r_2^m + \cdots + r_n^m = \sum_{i=1}^{n} r_i^m$$

There exists a simple recurrence relationship between the coefficients of the polynomial and S_m, allowing S_m to be easily found from polynomial coefficients and vice-versa. In addition, it may be shown [9] that the root moments of the factor of polynomial $F(z)$ with roots inside closed contour Γ are given by

$$S_m^\Gamma = \frac{1}{2\pi j} \oint_\Gamma \frac{f'(z)}{f(z)} z^m dz \tag{1}$$

The above can be easily and efficiently calculated as a Discrete Fourier Transform when Γ takes the form $z = \rho(\theta)e^{j\theta}$. In addition, if the root moments of polynomial $f_1(z)$ are $S_m^{f_1(z)}$ and the root moments of polynomial $f_2(z)$ are $S_m^{f_2(z)}$, then the root moments of $f(z) = f_1(z)/f_2(z)$ will be $S_m^{f(z)} = S_m^{f_1(z)} - S_m^{f_2(z)}$ [9].

We may express $F_1(z) = F_{1+}(z)/F_{1-}(z)$ where F_{1+} contains all the roots of $F(z)$ which lie within the outer-ring of the annulus, i.e. within the contour Γ_1, defined as $z = \rho^+ e^{j\theta}$, and F_{1-} contains all the roots of $F(z)$ which lie within the inner-ring of the annulus Γ_2, defined as $z = \rho^- e^{j\theta}$ ($\rho^+ > 1, \rho^- < 1$ and $\rho^+, \rho^- \approx 1$). In this way, we may directly derive the root moments of the polynomial of interest, $F_1(z)$ from two evaluations of Eqn. 1, one for the outer ring and one for the inner ring, followed by their subtraction.

Once $F_1(z)$ has been extracted, the deflation-based technique can be employed to find all roots of interest z_ℓ, $1 \leq \ell \leq r$, and extract their arguments θ_ℓ. Once a set θ_ℓ, $1 \leq \ell \leq r$ of such zeros, in order of ascending θ and normalized to lie in the range $0 \leq \theta \leq \pi$ has been found, it is sufficient to test a single point between each θ value to complete the feasibility test. For example, using the mid-point:

$$\begin{array}{rrcll} & N(e^{j\frac{1}{2}(\theta_\ell + \theta_{\ell+1})}) & \leq & |C(e^{j\frac{1}{2}(\theta_\ell + \theta_{\ell+1})})|^2 & 1 \leq \ell < r \\ \Rightarrow & N(e^{j\theta}) & \leq & |C(e^{j\theta})|^2 & \forall\theta \end{array}$$

This entire feasibility test has been automated and incorporated within the Synoptix system as the core of the wordlength optimization procedure, to be described in the next section.

5. WORDLENGTH OPTIMIZATION

Given a signal flow graph and a set of contraints on the output noise spectra, the task of Synoptix is to implement the circuit using the minimum area. We have developed area models of the building blocks used: adders, multipliers, multiply-accumulate blocks and registers. These may then be used to estimate the effect on area of changing the bit-width of each signal in the SFG. The problem is essentially one of optimization, with the vector of signal widths in the SFG, $\mathbf{n}$, being the optimization variable. Given an area estimator based on the building block area models, $A(\mathbf{n})$, the problem is to minimize $A(\mathbf{n})$ subject to $n_j \in Z^+ \ \forall j$ (where n_j is the j'th element of $\mathbf{n}$), and also subject to the feasibility constraints described in the previous section. This is a complex optimization problem for two reasons. Firstly the variables are constrained to be positive integers. Secondly, while area generally rises with $\mathbf{n}$ and output error generally falls with $\mathbf{n}$, it is not true that either effect is monotonic in any n_j.

We now describe one of the heuristics, AS, designed to solve this optimization problem. A pseudo-code for AS is set out below.

```
Algorithm AS
Input: A SFG, a constant k>1.
Output: A set of signal widths n.

S1) Run algorithm UNIF to find optimal
    uniform signal width u
S2) Set w(j) := ku for all j
S3) Improve the solution
  S3.1) improved := FALSE
  S3.2) n := w
  S3.3) Reduce each signal width in turn
    S3.3.1) Reduce signal j by one bit
    S3.3.2) If solution feasible and
      smallest area so far, b := j and
      improved := TRUE. Goto S3.3.1.
      If solution infeasible move on to
      next signal and Goto S3.3.1.
      If there are no more signals
      Goto S3.4
  S3.4) If improved = TRUE, reduce w(b) by
    one bit and Goto S3. Otherwise set
    n := w and STOP.
```

Firstly a binary search is performed (algorithm UNIF) in order to find the optimal uniform signal width u (i.e. when all signals in the SFG are assumed to have identical bit-width). This is because S3 assumes an initial feasible set of signal widths in $\mathbf{w}$. The algorithm only reduces widths, so widths are initialized to ku in order to allow them to take values above u. The next phase of the algorithm is to reduce signal widths one bit at a time. The signal width to reduce at each step is chosen by S3.3. The intuition is that there must come a stage in reducing each signal at which the SFG is no longer feasible. At this stage, some reduction in area over the original solution is likely to have occurred. This area gain is compared to that achieved by reducing all other signals, and the one with the most gain wins.

In this way both the constraint and the objective function play a role in determining the direction of movement towards the solution. The algorithm is neither based solely on local information, nor is it a global algorithm. It looks some way into the future to 'keep an eye' on the constraint surface.

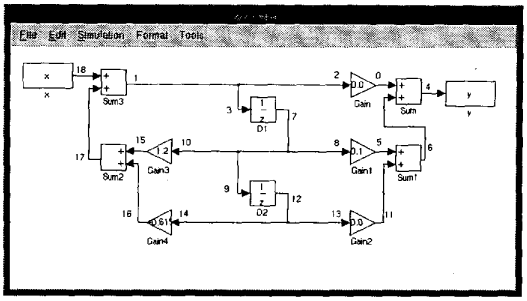

Figure 1: A Simple Signal Flow Graph

6. RESULTS

The technique for roundoff-noise shaping described in this paper has been implemented within the Synoptix high-level synthesis tool. This tool accepts a Simulink [1] signal flow graph as input and produces Hardware Description Language output suitable for direct synthesis and implementation with FPGAs.

Fig. 1 shows one such SFG of a simple second-order filter. Shown in Fig. 2 are the spectral profiles of the different paths that exist from each point within this filter's structure to its output. Synoptix maintains an internal representation of these profiles and uses them to construct the estimated roundoff-noise spectrum for a given fixed-point implementation of the filter. Shown in Figs. 3(a,b) are two spectral noise specifications and the corresponding noise spectra of the area-minimal filters produced by Synoptix. It is clear that Synoptix exploits the possibility of using different wordlengths at different points within the structure in order to achieve a tight-fitting implementation. Comparison with Fig. 3(b,c), where the same specifications are applied but a uniform wordlength has been used over all signals, shows this to be the case. Comparing Figs. 3(a) and (c) and Figs. 3(b) and (d) demonstrates that the optimization has been able to 'stretch' the output noise PSD to closely meet the specifications. This in turn translates into significant area savings: for this example the uniform wordlength structure requires 810 logic cells in an Altera Flex10K device, compared to 636 and 663 logic cells for the shaped designs, a 22% and 18% area reduction respectively.

7. CONCLUSIONS AND FUTURE WORK

We have presented a technique for exploiting the possibility of shaping roundoff noise within filters. This technique has been implemented within the Synoptix high-level synthesis system for synthesis to FPGAs, leading to significant area reductions in real circuits. As far as we are aware, there has been no other published work which attempts to adjust the wordlengths in a signal flow graph structure in order to meet fully specified constraints on roundoff noise PSD at the filter outputs.

A dedicated resource binding [11] was assumed in order to form area estimates. We are currently investigating the interaction between the resource-sharing problem for high-level synthesis and the precision optimization presented in this paper. No consideration has been given to maximum clock-rate in the synthesised filters. More analysis must be performed in to see whether this

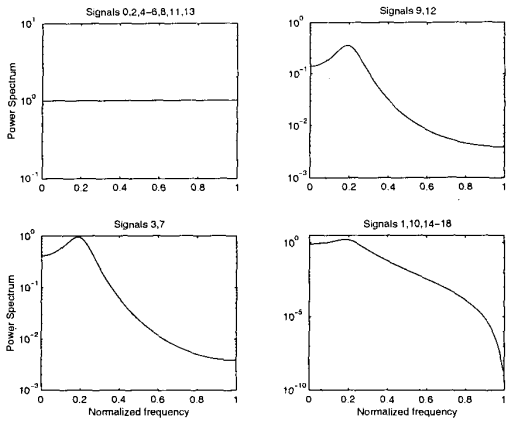

Figure 2: Spectral profiles through the filter

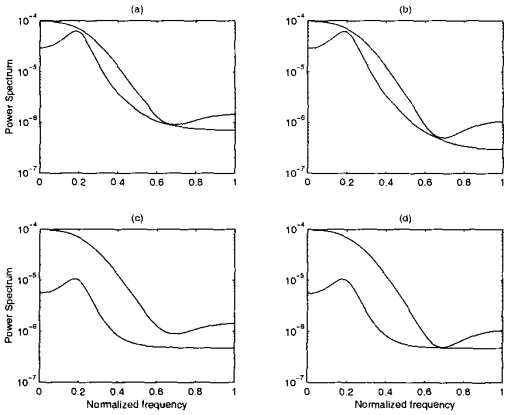

Figure 3: Two specifications (upper-curve) and their optimized (a),(b) and uniform-width (c),(d) implementations

is a factor that would affect the constraint surface significantly. The system, as presented, will not synthesise non-linear or time-varying systems. If analytical techniques are to be extended into non-linear and time-varying domains, a significantly different and more general theoretical framework needs to be laid. However, it is entirely possible to extend the present method to some restricted classes of non-linear or time-varying system, for example adaptive FIR filters, if some statistical knowledge of the variation of FIR coefficients is known.

8. REFERENCES

[1] MathWorks, *Simulink*, http://www.mathworks.com

[2] A.V. Oppenheim and C.J. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform", *Proc. IEEE*, 60(8):957–976, 1972.

[3] B. Liu, "Effects of Finite Word Length on the Accuracy of Digital Filters - A Review", *IEEE Trans. Circuit Theory*, CT-18(6):670–677, 1971.

[4] G.A. Constantinides, P.Y.K. Cheung and W. Luk, "Truncation Noise in Fixed-Point SFGs", *IEE Electron. Lett.*, 35(23):2012–2014, 1999.

[5] S. Kim, K. Kum, and W. Sung, "Fixed-Point Optimization Utility for C and C++ Based Digital Signal Processing Programs," *IEEE Trans. on Circuits and Systems II*, 45(11):1455–1464, 1998.

[6] J. Choi, H. Jun and S. Hwang, "Efficient hardware optimization algorithm for fixed-point digital signal processing ASIC design," *IEE Electronics Letters*, 32(11), 1996.

[7] M. Willems, V. Bürsgens, H. Keding, T. Grötker and M. Meyr, "System-level fixed-point design based on an interpolative approach," in *Proc. 34th Design Automation Conference*, June 1997.

[8] R. Cmar, L. Rijnders, P. Schaumont, S. Vernalde and I. Bolsens, "A Methodology and Design Environment for DSP ASIC Fixed Point Refinement," in *Proc. DATE-99*, München, 1999.

[9] T. Stathaki, "Root moments: a digital signal-processing perspective", *IEE Proc.-Vis. Image Signal Process.*, 145(4):293–302, 1998.

[10] W. H. Press, B. P. Flannery, S. A. Teukolskyand W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1988.

[11] G. DeMicheli, *Synthesis and optimization of digital circuits*, McGraw-Hill, New York, 1994.