Analytical Optimization of Bit-Widths in Fixed-Point LTI Systems

Omid Sarbishei, Katarzyna Radecka, *Member, IEEE,* and Zeljko Zilic, *Senior Member, IEEE*

***Abstract*—Analyses of range and precision are important for high-level synthesis and verification of fixed-point circuits. Conventional range and precision analysis methods mostly focus on combinational arithmetic circuits and suffer from major inefficiencies when dealing with sequential linear-time-invariant circuits. Such problems mainly include inability to analyze precision when quantization of constant coefficients is taken into account, and lacking efficient word-length optimization algorithms to handle both variables and constants, while satisfying the error metrics. The algorithms presented in this paper solve these problems. Experiments illustrate the efficiency and robustness of our algorithms.**

***Index Terms*—Fixed-point linear-time-invariant (LTI) circuits, precision analysis, range analysis, word-length-optimization.**

I. Introduction

DISCRETE linear-time-invariant (LTI) systems are widely used in many digital signal processing (DSP) applications. Such systems include finite impulse response (FIR) filters, infinite impulse response (IIR) filters, as well as fast Fourier transform (FFT) and discrete cosine transform (DCT) units. Implementing such circuits using a fixed-point data representation is a common approach [1], and is still gaining importance as many designs migrate to FPGAs, where floating-point arithmetic is disadvantageous.

The analysis of range and precision is an important part of the high-level optimization and verification process, as it is a basis for determining integer (IB) and fractional (FB) bit-widths for all the variables. Range analysis allows avoiding the overflow, while the precision analysis helps to provide error bound in terms of either maximum mismatch (MM), maximum mean-square-error (MSE), or signal-to-quantization-noise-ratio (SQNR). Given the reference y and its fixed-point realization y_{fixed}, error metrics MM and MSE represent the absolute difference: $\max(|y - y_{\text{fixed}}|)$, and the expectation $E(|y - y_{\text{fixed}}|^2)$, respectively. The MM metric provides the largest *spot* error, while MSE/SQNR deals with the classical signal-to-noise notion. The application of MM to the analysis of LTI circuits was addressed in [14] and [30], while papers [21] and [30] target MSE/SQNR, which are particularly important in DSP applications [21]–[25].

Several methods have been introduced to compute the range and precision of arithmetic circuits. Dynamic analysis [1], [7]–[9], [35] involves simulations, and hence is slow and nonrobust, which confines its applicability. Static analysis, on the other hand, has gained major interest in recent years [2]–[5], including arithmetic transform [6] or Taylor series [10]. Much of the effort has focused on the static analysis of MM for direct-flow-graph designs that can provide safe, but possibly pessimistic results. However, such solutions may not always be adequate, since they cannot handle recursive datapaths, such as IIR filters.

In the case of LTI circuits, conventional static approaches for analyzing ranges are either based on the computation of the L_1 norm of impulse responses [14], [26], [30] or the utilization of affine arithmetic (AA) [31]. Both solutions are robust, and provide overestimations of the exact range.

Conventional static analyses of precision in terms of MM or MSE/SQNR for LTI circuits either ignore the effect of coefficient quantization error originating from scaling the constant coefficients in the reference model to particular FB values, or make use of approximations such as perturbation and sensitivity analysis to handle coefficient errors. Both of these cases result in major underestimations of MM and MSE, which is not safe and robust as explored in Section II.

On the other hand, the MM and MSE/SQNR analyses presented in this paper alleviate the problems that exist with previous methods. This paper addresses four particular issues.

1) Provides a new error model for LTI circuits, for all quantization error sources originating from scaling of variables or constants to acceptable FB values.
2) Proposes new static analyses of MM and MSE/SQNR for LTI circuits. The analysis is safe and robust, unlike conventional methods that underestimate error and ignore coefficient quantization.
3) Demonstrates a novel efficient analytical optimization to set the FB values of variables and constants, while satisfying a given bound on MM or MSE/SQNR. Our solution is more efficient than the conventional methods that do not provide means for simultaneous control of the bit-widths, including the coefficients.
4) Proposes a more proficient range for LTI circuits compared to the previous work.

The rest of this paper is organized as follows. Section II provides the background. Section III addresses the range

Manuscript received September 11, 2010; revised March 9, 2011 and July 24, 2011; accepted September 15, 2011. Date of current version February 17, 2012. This paper was recommended by Associate Editor N. Ranganathan.

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada (e-mail: omid.sarbishei@mail.mcgill.ca; katarzyna.radecka@mcgill.ca; zeljko.zilic@mcgill.ca).

Digital Object Identifier 10.1109/TCAD.2011.2170988

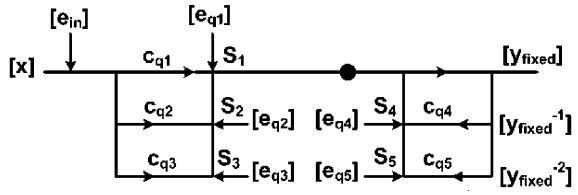


Fig. 1. Implementation of a fixed-point direct form second-order IIR filter.

analysis for an LTI system; the computation of MM is presented in Section IV, and an algorithm is introduced to set FBs. Section V extends the precision analysis to MSE (SQNR) and presents an algorithm for setting FBs under given MSE bound. Section VI gives the experimental results.

II. Definitions and Background

The discrete LTI circuit can be expressed as $[y] = [x] * [h]$, where $[x]$ and $[y]$ are the discrete input and output respectively, while $[h]$ is the impulse response. Through the rest of this paper we represent the discrete variable $[x]$ as x. Further, note that $[h]$ depends on some constant coefficients that have to be quantized to a suitable FB.

The process of implementing a fixed-point LTI circuit from the specification traditionally takes two steps as shown in Table I. First, the coefficients are quantized, resulting in the quantized impulse response h_q and the output y_q. Then, the quantization errors of the input x, i.e., e_{in}, and the intermediate variables that are scaled, i.e., $e_{q1},\ldots, e_{qN}$, are considered. Note that the intermediate variables depend on the LTI topology, e.g., direct or parallel form IIR filters. The output y_{fixed} is presented in the third row of Table I, where h_{qi} is the transfer function from e_{qi} to the output y_{fixed}.

As an example, assume the second-order fixed-point IIR in Fig. 1. The values of $c_{q1}, \ldots, c_{q5}$ are the quantized coefficients, while $S_1, \ldots, S_5$ are the intermediate variables with the corresponding quantization errors $e_{q1}, \ldots, e_{q5}$.

Based on Table I, the error metric MM can be defined as

$$MM = \max(|y - y_{\text{fixed}}|) \tag{1}$$

where the *maximum* function in (1) is defined over all the feasible combinations of input patterns, as well as all the possible values of quantization errors. The precision analysis aims to verify that the maximum mismatch MM does not exceed a given error bound $E_{\max}$

$$MM < E_{\max}. \tag{2}$$

Similarly, MSE can be obtained as

$$\begin{aligned} MSE &= E(|y - y_{\text{fixed}}|^2) \\ &= \lim_{M\to\infty} \tfrac{1}{M} \sum_{k=0}^{M} \mathrm{E}(|y[k] - y_{\text{fixed}}[k]|^2) \end{aligned} \tag{3}$$

where M is the length of the input sequence that is infinite in the case of feedback circuits and finite for the nonfeedback

TABLE I
Implementation of a Fixed-Point LTI System

Design	Impulse Responses	Output
Reference	h	$y = x * h$
Coefficient quantized	h_q	$y_q = x * h_q$
Final fixed-point	$h_q, h_{q1}, \ldots, h_{qN}$	$y_{fixed} = (x + e_{\text{in}}) * h_q + e_{q1} * h_{q1} + \cdots + e_{qN} * h_{qN}$

LTI circuits. Extending this notation to a p-output system $[y_1]$, $[y_2]$, $\ldots$, $[y_p]$ results in

$$MSE = \frac{1}{p} \sum_{j=1}^{p} MSE(y_j). \tag{4}$$

A given maximum bound on MSE, i.e., E_{bound}, is satisfied, if $MSE < E_{\text{bound}}$. The MSE can be used to obtain SQNR as

$$SQNR = \frac{E(|y|^2)}{MSE} = \frac{E(|x|^2) \times \sum_{j=0}^{\infty} |h[j]|^2}{MSE} \tag{5}$$

where $E(|y|^2)$ is the power of the original specification that is either given or easy to compute. As SQNR is directly derived from MSE, in this paper we only address the computation of MSE.

There have been many developments aiming to analyze range [14], [26], MM [12]–[15], or MSE/SQNR [21], [31] of fixed-point LTI circuits. However, tighter ranges and better precision analysis than the ones present in the literature can be obtained using our method implementing simultaneously the two steps in Table I.

Regarding range analysis of LTI circuits, conventional approaches are either based on the L_1 norm of impulse responses [14], [26], [30] or on the utilization of AA [31]. Both solutions are robust and provide overestimations of the exact range.

In terms of precision analysis for LTI circuits, conventional methods either ignore the effect of coefficient quantization, e.g., [11], [21], and [26], or make use of approximations like perturbation and sensitivity analysis to handle such errors [14], [27], [30], and [33]. The approaches that ignore the effect of quantizing coefficients compute MM and MSE with an error defined as $y_q - y_{\text{fixed}}$ (Table I), i.e., they assume that $h = h_q$. This can result in major underestimations of the *exact* error, given as $y - y_{\text{fixed}}$ (Table I), which is neither safe nor robust. The design process initially chooses very high FBs for constant coefficients, to keep the reference characteristics, such as group delay, still acceptable. The error analysis including the impact of quantizing input and intermediate variables is taken into account independently, since those error sources do not change the original impulse response and the position of its zeros and poles. Word-length optimization heuristics are then applied for the input and intermediate variables, such that a given bound on MSE or SQNR is satisfied [32]. The above solution is not efficient in terms of hardware costs. In fact, work in [20] has formally proved that for a suitable low-resolution variable or a coefficient in a polynomial datapath, if its FB is increased by one bit only, it becomes possible to widely reduce the FB of several other variables. Hence, lack

of flexibility in controlling the FB of constant coefficients and an initial selection of safe high FB values [11], [21], [26], [32] puts restrictions on the efficiency of the design optimization.

However, there are solutions like the ones in [14], [27], [30], [33], and [39] which take into account the effect of coefficient quantization in their error analysis. Particularly, the method in [33] assumes the coefficient errors as a source of noise by adding a random jitter to the coefficients and determining the MSE/SQNR w.r.t. such noise. The approaches in [13] and [31] make use of a perturbation and sensitivity analysis w.r.t. the coefficient quantization errors to track their impact on the position of zeros and poles. The analysis involves MM. The major drawback of the methods in [30] and [33] lies in their inability to deal with feedback circuits of orders higher than 2, and with LTI circuits in general. Furthermore, these solutions are based on a linear approximation, which limits robustness.

The method in [15] offers an alternative precision analysis in terms of MM rather than MSE/SQNR that is applicable to an arbitrary IIR filter. However, the importance of MSE/SQNR is further confirmed by methods like [31] and [34], where the analysis of MSE/SQNR has been extended to nonlinear and time-varying circuits. In particular, the scheme in [31] utilizes AA to compute range, MM, and MSE/SQNR. However, neither [31] nor [34] takes into account the effect of coefficient quantization error.

In contrast to previous methods, this paper provides a static analysis for range, MM, and MSE/SQNR for LTI circuits, which does not suffer from the problems encountered in conventional solutions. In particular, our range analysis can lead to tighter results compared to previous methods. Furthermore, our MM and MSE/SQNR analysis takes into account a more comprehensive representation of an error, by including all the error sources, i.e., variables and coefficient quantization errors. Further, it is applicable to all LTI circuits unlike methods in [14], [27], [33], and [39]. Our error analysis provides a negligible overestimation of the error. Hence, the computed values of MM and MSE are robust. Note that the conventional analyses ignoring the effect of coefficient quantization are nonrobust, since they underestimate the error. Finally, our analysis leads to an optimization of FBs and IBs of both variables and constant coefficients, while satisfying the error bounds.

III. Range Analysis

In this section, we propose a range computation algorithm for obtaining minimum IBs in an implementation of the bounded input bounded output (BIBO) stable causal LTI system $y = x * h$. This problem, crucial for the discrete system design, has been addressed previously, where the solutions in [14], [16], and [26] utilize the L_1 norm of $[h]$, i.e., $|||[h]|||_1 = \sum_{k=0}^{\infty} |h[k]|$ to compute an overestimation of the range using the following inequality:

$$\max(|y|) \leq |||[h]|||_1 \times \max(|x|). \tag{6}$$

The norm $|||[h]|||_1$ can be computed using a simple numerical method [14]. The convergence condition in (6) is governed by the position of poles in the system. Namely, the closer the poles are to the unit circle (stability condition), the higher the number of summations L is required to estimate $|||[h]|||_1$, that is

$$|||[h]|||_1 = \sum_{k=0}^{\infty} |h[k]| \approx \sum_{k=0}^{L} |h[k]|.$$

The reason is that for BIBO stable LTI systems we have $\lim_{k\to\infty} h[k] = 0$, and hence, if L is high enough, then the changes in $|||[h]|||_1$ become negligible and the summation loop converges.

Example 1: Consider an LTI circuit with a single dominant pole at $z = z_0$, i.e., $y[n] = x[n-1] + z_0 y[n-1]$. The value of $|h[k]|$ decreases (increases) w.r.t. k, if $|z_0| < 1 (|z_0| > 1)$ with the complexity of $O(|z_0|^k)$. Now assume that the resolution ranges from 2^{-S} to 2^S. Under such conditions, the samples $h[k]$ converge to zero ($<2^{-S}$) or infinity ($>2^S$) after $L = \lceil |S \log_{|z_0|} 2| \rceil$ iterations, where $\lceil . \rceil$ is the ceiling function and rounds its input to the closest higher integer. For instance, if $S = 64$, then for the stable circuit with $z_0 = 0.99$, converging to the condition $|h[L]| > 2^{-64}$, is obtained after $L = 4414$ iterations, while for the unstable circuit with $z_0 = 1.01$, the condition $|h[L]| > 2^{64}$ is satisfied at $L = 4459$. Hence, even with poles very close to the unit circle, the convergence is obtained relatively fast. □

The inequality used for the bound in (6) may result in a major overestimation of the range, and lead to the allocation of superfluous IBs for the fixed-point variables.

A tighter range with less overestimation is obtained as

$$\max(y[n]) \leq \sum_{k=0}^{n} \max(x \times h[k]) \tag{7}$$

$$\Rightarrow \max(y) \leq \sum_{k=0}^{\infty} \max(x \times h[k]) \approx \sum_{K=0}^{\hat{L}} \max(x \times h[k]). \tag{8}$$

The variable $\hat{L}$ in (8) represents the number of summations required to estimate $\sum_{k=0}^{\infty} \max(x \times h[k])$. The simple numerical scheme in [14], which is utilized to compute (6), can be used to compute $\max(y)$ in (8) as well. The variable $\hat{L}$ is dependent upon the position of poles like in (6). The closer the poles are to the unit circle, the higher value of $\hat{L}$ is required for convergence. Note that if the input sequence $x[n]$ is uncorrelated, then the inequalities in (7) and (8) reduce to equality conditions, which means that the computed range is exact. We can make use of (8) to compute the minimum value of y as well.

Example 2: Consider the following IIR filter:

$$y[n] = 2.487x[n] + 0.131x[n-1] - 0.42x[n-2] \\ -0.141y[n-1] + 0.492y[n-2] - 0.087y[n-3].$$

The input bound is [0, 100]. Using (8), after $n = 178$ samples, the computation of range converges to the interval $[-196.39, 495.03]$, which corresponds to an overestimated integer range of $[-197, 496]$. Ten bits of IB including a sign bit are required to represent this interval. Using (6) to compute the range [14], [16], [26] gives the coarser result of $[-692, 692]$, which requires the IB of 11. Note that AA applied

to compute the range [31] gives the coarse result of [−394, 692], which requires the IB of 11 as well. □

IV. MM Analysis and FB Allocation

The analysis of precision in terms of MM verifies that the condition in (2) is satisfied. Moreover, it is required to find acceptable FBs for all constants and variables, while satisfying the error bound given by (2). In this section, we present an efficient analysis of MM for LTI circuits, which leads to a simple formula to set the FB of all the variables, while satisfying the condition in (2).

Three error sources input quantization, coefficient quantization, and scaling error (generated by scaling of intermediate variables) have to be taken into consideration when representing the fixed-point output y_{fixed} (Table I). In this paper, we only address truncation, and round-to-nearest types of quantization. The truncation is just a shift-right operation and does not require additional hardware, while round-to-nearest necessitates the shift, add and comparison operations. The scaling errors, i.e., $e_q = x_{\text{scaled}} - x$, have the following ranges for truncation and rounding-to-nearest:

$$\begin{array}{ll} \text{Truncation} & -2^{-FB} \le e_q \le 0 \\ \text{Round-to-nearest} & -2^{-FB-1} \le e_q \le 2^{-FB-1} \end{array} \tag{9}$$

where FB is the fractional bit-width of the scaled variable x_{scaled}. The round-to-nearest scheme in (9) provides the exact maximum and minimum bounds for the quantization error. However, regarding truncation, in order to compute bounds on the quantization error we have to replace 2^{-FB} in (9) with $2^{-FB} - 2^{-FB_{\text{old}}}$, where FB_{old} is the FB of nonscaled variable x ($FB_{\text{old}} > FB$) [19]. However, as explored in [20] and indicated by experiments in Section VI, FB_{old} is usually much higher than FB, which makes it reasonable to overestimate $2^{-FB} - 2^{-FB_{\text{old}}}$ with simply 2^{-FB}. As presented in the rest of this section, due to the only slightly overestimated interval in (9) for the truncation case, we are able to devise a fast analytical approach to set the FB values of variables, while minimizing hardware cost.

Regarding quantization of coefficients, other scaling techniques such as rounding to powers-of-2 can be investigated to save hardware cost; however, this method contributes to very high errors, which can make the circuit unstable. For coefficients, we assume the round-to-nearest quantization, since it contributes to the lowest bound on error, (9), and furthermore, it does not require the *shift*, *add*, and *comparison* operations.

The fixed-point representation of y_{fixed} (Table I) is

$$y_{\text{fixed}} = (x + e_{\text{in}}) * h_q + \sum_{j=1}^{N} (e_{qj} * h_{qj}) \tag{10}$$

where the quantization errors e_{in} and e_{qj}, $j \in \{1, \ldots, N\}$, may lie within different intervals given by (9). Hence, we can express the mismatch function $y_e = y - y_{\text{fixed}}$ as

$$\begin{aligned} y_e &= y_{\text{fixed}} - y = y_{\text{fixed}} - x * h \\ &= x * (h_q - h) + e_{\text{in}} * h_q + e_{q1} * h_{q1} + \cdots + e_{qN} * h_{qN} \\ &= y_{ex} + y_{e0} + y_{e1} + \cdots + y_{eN} \end{aligned} \tag{11}$$

where $y_{ex} = x * (h_q - h)$, $y_{e0} = e_{\text{in}} * h_q$, $y_{e1} = e_{q1} * h_{q1}$, $y_{eN} = e_{qN} * h_{qN}$.

Given the FB of coefficients, we can compute h_q, $h_{q1}, \cdots, h_{qN}$ based on h and the given topology of the circuit, e.g., direct or parallel form IIR filter. Additionally, if the FB values corresponding to e_{in} and e_{qj} are known, the ranges of e_{in} and e_{qj} can be calculated based on (9). In consequence, the problem of determining the maximum and minimum values (bounds) of (11), i.e., MM can be redeclared as a range analysis issue. It can be then solved using the adjusted (8), where the major amendment to (8) is in handling the input space. In particular, (8) deals with only one input x, whereas now we need to consider several inputs: x, $e_{\text{in}}, e_{q1}, \ldots, e_{qN}$. Furthermore, an individual impulse response corresponds to each input. Hence, the computation of all the impulse responses associated with functions assigned to inputs x, $e_{\text{in}}, e_{q1}, \ldots, e_{qN}$, is required.

Finally, there is a correlation between e_{qj} ($j \in \{1, \ldots, N\}$) and the input x, i.e., y_{ex} and y_{ej}, which can be ignored by making use of the triangle inequality $\max(y_{ej} + y_{\text{ex}}) \le \max(y_{ej}) + \max(y_{\text{ex}})$. Hence, an upper bound on the maximum value of y_e in (11) is obtained as follows:

$$\begin{aligned} &\max(y_e) = \max(y_{\text{ex}} + y_{e0} + y_{e1} + \cdots + y_{eN}) \le \\ &\max(y_{\text{ex}}) + \left(\sum_{j=0}^{N} \max(y_{ej}) \right) \end{aligned}$$

where each maximum term, i.e., $\max(y_{\text{ex}})$ and $\max(y_{ej})$ ($j = 0, \ldots, N$) in the above equation can be obtained based on (8). Note that in order to compute $\max(y_{ej})$, we must replace the range of input x in (8) with the range of the quantization errors given by (9).

The above discussion for computing MM is valid when all the FBs of coefficients and variables are known. In the rest of this section, we present an analysis to set the values of FBs such that after computing MM based on the previous discussion, a given error bound on MM is satisfied, (2).

Definition 1: Upper and lower output bounds after n samples. The upper and lower output bounds (B_{upp} and B_{low}) of the output y reached after the first n samples are

$$\begin{array}{l} B_{\text{upp}}(y[n]) = \max(\max(y[0]), \ldots, \max(y[n])) \\ B_{\text{low}}(y[n]) = \min(\min(y[0]), \ldots, \min(y[n])). \end{array} \tag{12}$$

Equation (12) indicates that $B_{\text{upp}}(y[n])(B_{\text{low}}(y[n]))$ is a monotonically increasing (decreasing) function of n.

The precision analysis in terms of MM leading to the selection of FB values for system variables presented in the rest of this section uses the following sequence in reasoning:

Lemma 1: Determines $B_{\text{upp}}(y_{ej})$, where y_{ej} is given by (11), and ($j = 0, \ldots, N$)
Lemma 2 and Corollary 1: Computes the perturbation of $B_{\text{upp}}(y_{ej})$ w.r.t. the changes in FB values
Lemma 3: Calculates $B_{\text{upp}}\left(\sum_{j=0}^{N} y_{ej}\right)$ in (11)
Lemma 4: Sets analytically FB values for variables.

Lemma 1: Consider the mismatch term $y_{e0} = e_{\text{in}} * h_q$ in (11), where e_{in} has a bound based on the type of the

quantization, (9). The upper bound of y_{e0} after n samples can be expressed as $B_{\text{upp}}(y_{e0}[n]) = \max(y_{e0}[n])$. Moreover, the lower output bound of y_{e0} after n samples is $B_{\text{low}}(y_{e0}[n]) = \min(y_{e0}[n])$.

Proof: Since $y_{e0} = e_{\text{in}} * h_q$, by using (7) we obtain

$$\begin{aligned} &\max(y_{e0}[n+1] = \\ &\max(y_{e0}[n]) + \max(e_{\text{in}} h_q[n]). \end{aligned} \tag{13}$$

For the bound on the input e_{in} coming from the round-to-nearest or truncation quantization types, (9), we have

$$\begin{aligned} \max(e_{\text{in}} h_q[n]) &= \max(|e_{\text{in}}|) \times (|h_q[n]|) \geq 0 \\ \max(e_{\text{in}} h_q[n]) &= -\max(|e_{\text{in}}|) \times (|h_q[n]|) \leq 0. \end{aligned}$$

By applying this to (13) we obtain

$$\max(y_{e0}[n+1] \geq \max(y_{e0}[n])$$

and hence, $B_{\text{upp}}(y_{e0}[n]) = \max(y_{e0}[n])$, (12). ∎

Lemma 1 can be extended to the other quantization terms $y_{ej}\,(j = 1, \ldots, N)$ in (11). In particular, it indicates that for term y_{ej} originating from a quantization noise within the interval in (9), functions $\max(y_{ej}[n])$ and $\min(y_{ej}[n])$ are, respectively, monotonically increasing/decreasing in n.

Lemma 2: Consider the mismatch term $y_{e0} = e_{\text{in}} * h_q$, (11), where e_{in} corresponds to FB fractional bits for a given quantization type, (9). Further, assume that the value of $\max(y_{e0}[n]) = A$ at the nth sample is computed based on (7). If the original FB is increased by p bits, i.e., $FB_{\text{new}} = FB + p$, where $p \geq -FB$, then the new maximum bound on the mismatch of the nth sample can be obtained as $\max(y_{\text{e\_new}}[n]) = 2^{-p} \times A$. Note that for negative values of p it is simply a reduction of number of FBs by $-p$ bits.

Proof: According to (7), the MM of $y_{e0}[n]$ is a linear function of the upper and lower bounds of e_{in}, which are given by (9). Based on that and the fact that the upper and lower bounds of e_{in} in (9) can be represented as $2^{-FB} \times c$, where c is either $0, -1$, or $\pm 1/2$ depending on the type of quantization, we deduce that for both types of quantization given by (9), we have $\max(y_{e0}[n]) = 2^{-FB} \times cte$. Here, cte is a constant positive value, which can be obtained by (7). Hence, if we change FB to $FB_{\text{new}} = FB + p$, then the new maximum bound on mismatch becomes $\max(y_{\text{e0\_new}}[n]) = 2^{-FB-p} \times cte = 2^{-p} \times A$. ∎

Corollary 1: Assume that we established the upper bound of $y_{e0} = e_{\text{in}} * h_q$, (11) to be C, i.e., $B_{\text{upp}}(y_{e0}) = \lim_{n\to\infty} \max(y_{e0}[n]) = C$, using (8). Based on Lemma 1, Lemma 2, and (12), if e_{in} corresponds to FB fractional bits, increasing FB by p bits results in the new upper bound of $2^{-p} \times C$.

The validity of Corollary 1 results from Lemma 2, which states that the value of $\max(y_{e0}[n])$ is multiplied by 2^{-p}, if FB is increased by p. Since the relation $B_{\text{upp}}(y_{e0}[n]) = \max(y_{e0}[n])$ is true due to Lemma 1, we deduce that the upper bound is also multiplied by 2^{-p}. Now that we have established how changing FB impacts the value of $B_{\text{upp}}(y_{ej}[n])$ $(j = 0, \ldots, N)$, we move to represent $B_{\text{upp}}(\sum_{j=0}^{N} y_{ej})$ w.r.t. the FB of variables.

Lemma 3: Consider the mismatch part of (11), which originates from $e_{\text{in}}, e_{q1}, \ldots, e_{qN}$

$$y_{eq} = y_{e0} + \cdots + y_{eN} = e_{\text{in}} * h_q + \sum_{k=1}^{N} e_{qk} * h_{qk}. \tag{14}$$

Based on Corollary 1, let the upper bounds of $e_{\text{in}} * h_q$, $e_{q1} * h_{q1}, \ldots, e_{qN} * h_{qN}$ be equal to $2^{-FB_{\text{in}}} \times A_0, 2^{-FB_1} \times A_1 \ldots, 2^{-FB_N} \times A_N$, respectively, where $A_j\,(j = 0, \ldots, N)$ is a positive value obtained by (8) to compute the upper bound for y_{ej}. Then, the bound on MM of y_{eq} in (14) is the summation of the bounds

$$B_{\text{upp}}(y_{\text{eq}}) = 2^{-FB_{\text{in}}} \times A_0 + \sum_{k=1}^{N} 2^{-FB_k} \times A_k. \tag{15}$$

Proof: First, note that the maximum values of $y_{e0}[n], \ldots, y_{eN}[n]$ all occur at the same time when $n \to \infty$ as they are all monotonically increasing functions of n, Lemma 1. Additionally, y_{eq} is a linear function of $y_{\text{eq}}, \ldots, y_{eN}$, (14). Therefore, the MM of y_{eq} in (14) is obtained by adding the individual bounds. ∎

The analysis provided by Lemmas 1–3, as well as Corollary 1, forms a basis for the following lemma, which proposes an efficient formula to efficiently set the FB values, while satisfying (2) and minimizing hardware cost.

Lemma 4: Assume that $FB_{\text{in}}, FB_1, \ldots, FB_N$ are selected such that the upper bounds of $e_{\text{in}} * h_q, e_{q1} * h_{q1}, \ldots, e_{qN} * h_{qN}$, i.e., $B_0 = 2^{-FB_{\text{in}}} \times A_0$, $B_1 = 2^{-FB_1} \times A_1, \ldots, B_N = 2^{-FB_N} \times A_N$ all satisfy the following condition:

$$\frac{E_{\max}}{2 \times (N+1)} < B_i \leq \frac{E_{\max}}{(N+1)}\,(i = 0, \ldots, N) \tag{16}$$

where $E_{\max}$ is the given error bound. Then, based on (15), the bound on MM of y_{eq} in (14) is equal to $B_{\text{upp}}(y_{\text{eq}}) = \sum_{i=0}^{N} B_i \leq E_{\max}$, which is the summation of individual bounds. If one of the fractional bit-widths, e.g., FB_{in}, is reduced by one, then it is necessary to increase by one at least two other arbitrary FB values FB_i and $FB_j, i, j \in \{1, \ldots, N\}$ in order to keep the upper bound on y_{eq} not larger than $B_{\text{upp}}(y_{\text{eq}}) = \sum_{i=0}^{N} B_i$.

Proof: If FB_{in} is reduced by 1, then according to (15), $B_{\text{upp}}(y_{\text{eq}})$ is changed to $B_{\text{upp}}(y_{\text{eq\_new1}}) = B_0 + \sum_{i=0}^{N} B_i$. In order to keep the new bound on MM equal to $\sum_{i=0}^{N} B_i$, we choose to increase FB_1 by 1. This reduces the MM by $\frac{B_1}{2}$

$$B_{\text{upp}}(y_{\text{eq\_new2}}) = B_0 + \left(\sum_{i=0}^{N} B_i\right) - \frac{B_1}{2}.$$

In consequence, the bound on MM is increased by $B_0 - \frac{B_1}{2}$. In the best case scenario, when B_0 and B_1 are respectively at their minimum and maximum values given by (16), i.e., $B_0 = \frac{E_{\max}}{2\times(N+1)} + e$ and $B_1 = \frac{E_{\max}}{(N+1)}$, where e is a positive small value according to (16), the additional error is equal to $B_0 - \frac{B_1}{2} = e > 0$. This means that the new bound on mismatch is still higher than $\sum_{i=0}^{N} B_i$ even when considering the best-case scenarios for B_0 and B_1. Hence, it is necessary to reduce another FB, e.g., FB_2, by 1 to further reduce the new bound on mismatch. ∎

Lemma 4 indicates that in terms of hardware cost, the most efficient approach to set the FB of input/intermediate variables

is to select equal limits for the terms B_i ($i = 0, \ldots, N$) in (16). Note that in some applications we do not have a freedom to choose arbitrary FBs for the input variables. In such cases FB_{in} is fixed; however, $FB_1, \ldots, FB_N$ of the intermediate variables can be selected by (16).

We now present the optimization algorithm for selecting FB values for variables and coefficients in an LTI circuit H. Our scheme guaranties that a given error bound $E_{\max}$ is satisfied for the mismatch function (2). The outputs of the algorithm are the MM of the fixed-point LTI circuit, as well as the FB of variables and constant coefficients.

The algorithm starts from setting the FB_c, i.e., the FB of coefficients, initially to a minimum value, i.e., 1. In Step 2, a *while* loop is invoked, which is responsible for finding the suitable FB_c. In this loop, at Step 5, the MM is computed using (7), (8), and (11), while assuming the input and intermediate quantization errors all equal to zero. If MM exceeds the error bound, the algorithm iteratively increases FB_c by 1 until a minimum possible FB_c that satisfies $E_{\max}$ is found. After setting FB_c, the FB of input and intermediate variables, i.e., $FB_{\text{in}}, FB_1, \ldots, FB_N$ are addressed in Steps 8 to 13. Note that (16) is used to allocate suitable values of FBs to the variables sequentially. If FB_{in} of the inputs is given as a fixed value with no freedom for changing it, then the procedure based on Lemma 4 and (16) has to be modified to address only $FB_1, \ldots, FB_N$.

Note that for the same values of FB_c and FB_i ($i = 1, \ldots, N$), contribute to higher hardware cost, as FB_i is for a fixed-point variable, while FB_c is for constant coefficients. Hence, in Step 14 the value of FB_c is compared with the rest of FBs. If FB_c is not of the highest value, then FB_c is increased by 1 and the second *while* loop (Step 7) is re-executed to achieve lower values of FB for other variables using (16) once again. Otherwise, the final FB values and MM are returned. Note that by increasing FB_c, the error term $y_{\text{ex}} = x * (h_q - h)$ in (11) is reduced, and hence, it leaves us a degree of freedom to choose higher values for the error terms y_{ej} ($j = 0, \ldots, N$) in (11). Due to (16), this results in lower FB values for the variables without exceeding $E_{\max}$. Therefore, the hardware cost is improved.

Example 3: Consider the direct form IIR filter in Fig. 3. The goal is to find suitable values of FBs for the input variable x, the intermediate variables c to f and the constant coefficients $\{0.1, -0.4, -0.1, 0.46, -0.08\}$ such that the error bound $E_{\max} = 0.1$ is satisfied. The quantization by truncation is chosen for the variables, while coefficients are treated with round-to-nearest method. The computation of the bounds on MM based on (7) and (8) converges after $n = 156$ samples with $MM = 0.088$. The following FBs for the variables are obtained after making use of the algorithm in Fig. 2:

$$FB_c = 14,\ FB_x = FB_{a-c} = 7,\ FB_{d-j} = FB_y = 8$$

where FB_m is the fractional bit-width of the variable m in Fig. 3 and FB_c is the fractional bit-width of the coefficients. If we ignored the effect of the coefficient quantization for the above example, which is the case for conventional methods, the maximum mismatch would be $MM = 0.0098$, which is a unsafe underestimation of error by the factor of 9. □

V. Analysis of MSE and Fractional Bit-Width Allocation

The analysis of the precision and MM may not be sufficient for LTI systems in some DSP applications, hence other error metrics are usually applied. Among them MSE (and in consequence SQNR) is one of the most important, as the output error between the fixed-point and reference circuit is assumed to be a random variable (noise). MSE is a suitable measure to indicate the power of noise [21]. In this section we present a robust analysis for computing MSE, which can be beneficial for both optimization and verification of real DSP systems. In particular, we first need to prove the validity of the robustness assumption of the uniform probability distribution for e_{qj} ($j = 1, \ldots, N$) in (11).

Based on (4) and (11) the MSE metric can be rewritten as

$$E(|y - y_{\text{fixed}}|^2) = \lim_{M\to\infty} \frac{1}{M} \sum_{k=0}^{M} E(|y_e[k]|^2). \tag{17}$$

It has been shown in [16] that by assuming a uniform distribution of an error e_q generated by the quantization, i.e., $e_q = x_{\text{scaled}} - x$ over the interval $[A, B]$, the value of $E(|e_q|^2)$ can be computed as

$$E(|e_q|^2) = \sigma_{e_q}^2 + (E(|e_q|))^2 = \frac{(B-A)^2}{12} + \frac{(A+B)^2}{4} \tag{18}$$

where $\sigma_{e_q}^2$ is the variance of e_q. Hence, based on (9) and (18), for the two quantization types we have

$$\begin{aligned} &\text{Truncation :} && E(|e_q|^2) = 2^{-2FB}/3 \\ &\text{Round-to-nearest :} && E(|e_q|^2) = 2^{-2FB}/12. \end{aligned} \tag{19}$$

Equation (19) is valid for continuous variables; however, based on the results in [20], as well as our experiments in Section VI, the FB values and number of truncation bits are typically higher than 8, which makes the approximation in (19) almost accurate [19].

Note that the quantization error of an intermediate variable in a fixed-point arithmetic circuit generally does not have a uniform distribution over its interval given by (9), [17], and [25]. However, the problem can be simplified for LTI systems, [19] and [21]. In particular, the only arithmetic operators in an LTI circuit are *additions* and *constant multiplications*. Hence, if we can prove for both of these operations that the assumption of a uniform distribution on the outputs (when the inputs are uniform) is acceptable, then all the quantization errors can be considered to have uniform distributions as well.

For the constant multiplier $S(x) = c \times x$, where c is a constant coefficient, if the input x has a uniform distribution over the interval $(-a, a)$, then obviously the output S has also a uniform distribution over $(-c \times a, c \times a)$. Hence, the quantizing error generated by scaling S is also uniform.

Although constant multipliers do not change the uniform distribution of outputs, adders do. In fact, the sum has a distribution, which is the convolution of the distributions of the inputs. Hence, in general, it is a complex task to derive a distribution of all the intermediate variables in an LTI circuit using the convolution approach. Furthermore, it is even more challenging to find the exact distribution of the quantization

error of intermediate variables based on the distribution of their corresponding nonquantized values. This problem is resolved by the following lemma, which illustrates that it is acceptable to assume a uniform distribution of the outputs of adders in an LTI circuit. As a result, the quantization errors can be considered uniform.

Lemma 5: Consider an addition operation $S = x_1 + x_2$, where x_1 and x_2 are independent input variables having uniform distributions over intervals $(-a_1, a_1)$ and $(-a_2, a_2)$, respectively. A uniform distribution of the sum S, and as a result, the associated quantization error, overestimates the value of the exact distribution of S, i.e., $E(|S|^2)$.

Proof: The exact value of $E(|S|^2)$ can be computed based on (18) as

$$E(|S|^2) = E(|x_1 + x_2|^2 = E(|x_1|^2) + E(|x_2|^2) = \frac{a_1^2 + a_2^2}{12}.$$

Since the value of S lies within the interval $(-a_1 - a_2, a_1 + a_2)$, then if we assume a uniform probability distribution for the sum S, using (18), we obtain the following new value of $E(|S_{\text{new}}|^2)$:

$$E(|S_{\text{new}}|^2) = \frac{(a_1 + a_2)^2}{12} = E(|S|^2) + \frac{a_1 a_2}{6}.$$

The fact that both a_1 and a_2 are positive numbers implies that $E(|S_{\text{new}}|^2) > E(|S|^2)$. ■

Note that Lemma 5 assumes that x_1 and x_2 are statistically independent, while in general they might be correlated. However, regarding LTI circuits, the degree of all variables is at most 1, and as a result even if there is a correlation between the intermediate variables x_1 and x_2 then $x_1 + x_2$ can be flattened w.r.t. some other primary variables that are statistically independent.

Hence, in a system involving adders and constant multipliers, as in an LTI circuit, the assumption of a uniform distribution for the intermediate quantization errors results in an overestimation of MSE, which is safe compared to the case where MSE is underestimated. The overestimation of MSE due to adders is mostly negligible, as per Lemma 5.

For example, assume an adder with one 12-bit input operand and another 8-bit one with uniform distributions over the intervals $[-2047, 2047]$ and $[-127, 127]$, respectively. The output is quantized by truncating its least significant b bits. Table II illustrates the values of $E(|e_{\text{exact}}|^2)$ and $E(|e_{\text{uniform}}|^2)$ for different values of b. Here, e_{exact} represents the exact quantization error on the output, while e_{uniform} stands for the error when considering a uniform distribution, i.e., (19). As can be seen, the overestimation becomes negligible as b increases. According to [20] and our experiments in Section VI, the typical number of truncation bits in an optimized imprecise circuit is higher than $b = 8$. Hence, the overestimation originated from the assumption of the uniform distribution for quantization errors is mostly negligible.

According to (19) the round-to-nearest scaling contributes to a much lower value of $E(|e_q|^2)$ compared to truncation. Hence, being the worse case, we only consider the round-to-nearest quantization to allocate suitable bit-widths to all the variables and constant coefficients when the goal is to satisfy a specific bound on MSE. In fact, the MSE analysis is simplified for the round-to-nearest quantization, as $E(e_q) = 0$, (9), while $E(e_q) \neq 0$ for truncation.

Finding the value of $E(|y_e|^2)$ in (17) where y_e is given by (11) is not trivial due to the correlation among the variable x and all the quantization errors $e_{q1}, \ldots, e_{qN}$. Note that the term $y_{\text{ex}} = x * (h_q - h)$ in (11) does not appear in the error analysis of conventional methods, which ignore the effect of coefficient quantization error resulting in $h_q = h$ (Table I). This makes the error analysis trivial, since the correlation between the variable x and all the quantization errors $e_{q1}, \ldots, e_{qN}$ does not affect the output error. However, in our analysis of the error defined by (11), we consider the correlation. The solution we propose is to overestimate the error function y_e by another error function, i.e., $\hat{y}_e > y_e$, such that $E(|\widehat{y_e}|^2)$ can be computed more easily. Under such conditions, an overestimated value of MSE is obtained. Lemma 6 establishes the background for the robust computation of MSE by overestimating the error.

Lemma 6: The mismatch function $y_e = y - y_{\text{fixed}}$ in (11) can be overestimated and marked as $\widehat{y_e}$, (20). The new mismatch function $\widehat{y_e}$ is a function of the statistically independent variables $e_{\text{in}}, e_{q1}, \ldots e_{qN}$

$$\begin{aligned}\hat{y}_e = \hat{y}_{ex} + y_{e0} + y_{e1} + \cdots + y_{eN} = \\ \max(|x|) * (h_q - h) + y_{e0} + \cdots + y_{eN}.\end{aligned} \tag{20}$$

Proof: The only term that has been overestimated in (20) compared to (11) is $\hat{y}_{\text{ex}} = \max(|x|) * (h - h_q)$, which is an overestimation of $y_{\text{ex}} = x * (h - h_q)$ ■

The overestimated mismatch function given by (20) includes the term $\hat{y}_{ex} = \max(|x|) * (h - h_q)$, which is independent of x. Therefore, when the round-to-nearest quantization is used, an overestimated value of MSE can be computed as follows based on Lemma 6 and the property

$$\begin{aligned}&E(e_{\text{in}}) = E(e_{q1}) = E(e_{qN}) = 0 : \\ &E(|y_e|^2) < E(|\widehat{y_e}|^2) = \\ &\qquad E(|\hat{y}_{\text{ex}}|^2) + E(|e_{\text{in}}|^2) \times \lim_{M\to\infty} \sum_{j=0}^{M} |h_q[j]|^2 + \\ &\qquad \sum_{i=1}^{N} (E(|e_{qi}|^2) \times \lim_{M\to\infty} \sum_{j=0}^{M} |h_{qi}[j]|^2)\end{aligned} \tag{21}$$

where

$$E(|\hat{y}_{\text{ex}}|^2) = (\max(|x|))^2 \times \left(\lim_{M\to\infty} \sum_{j=0}^{M} (h[j] - h_q[j]) \right)^2.$$

A. Overestimation of Mismatch Function

Lemma 6 introduces an overestimation to the mismatch function (20). In fact, the only term overestimated in (20) is $\hat{y}_{\text{ex}} = \max(|x|) * (h_q - h)$, which is an overestimation of $y_{\text{ex}} = x * (h_q - h)$ in (11). Hence, the worst-case (maximum value) of the overestimation generated for the output error, i.e., $\max(\hat{y}_e - y_e)$, occurs when the other nonoverestimated terms including $y_{e0}, \ldots, y_{eN}$ are all zero, i.e., $\hat{y}_e - y_e = \hat{y}_{\text{ex}} - y_{\text{ex}}$ [(11), (20)]. Assuming a uniform distribution of x over the interval $[A, B]$, the minimum value of $E(|x|^2)$ is computed as

$\frac{\partial E(|x|^2)}{\partial A} = 0 \overset{Eqn.(18)}{\Rightarrow} A = \frac{-B}{2} \Rightarrow \min\{E(|x|^2)\} = \frac{(B-(-B/2))^2}{12} + \frac{((-B/2)+B)^2}{4} = \frac{B^2}{4}$.

Therefore the exact MSE of y_{ex}, and as a result y_e is

$$E(|y_e|^2) = E(|y_{\text{ex}}|^2) = \frac{B^2}{4} \times \hat{H} \tag{22}$$

where $\hat{H} = \{\sum_{j=0}^{\infty}(h[j] - h_q[j])\}^2$. Hence, the overestimated MSE of $E(|\hat{y}_{\text{ex}}|^2)$ can be computed as

$$E(|\hat{y}_{\text{ex}}|^2) = (\max(|x|))^2 \times \hat{H} = B^2 \times \hat{H}. \tag{23}$$

Based on (22) and (23) in the worst-case scenario of computing MSE where no input/intermediate variable quantization exists, the overestimation is at most $10 \times \log_{10} 4 = 6$ dB. However, when other sources of quantization error, including input and intermediate variables quantization are considered, the overestimation becomes lower than 6 dB. Hence, if y_{ex} is much lower than $y_{e0} + y_{e1} + \cdots + y_{eN}$, then $E(|\hat{y}_{\text{ex}}|^2)$, which is the only overestimated term in (20), becomes much smaller compared to the remaining terms in (20). Therefore, the overestimation becomes negligible. The experiments in Section VI provide data to compare our overestimated MSE analysis with the approximate analysis based on [11], [21], and [26], and the exact MSE obtained by exhaustive simulations.

Furthermore, since overestimating error leads to the increase in hardware costs, it is important to find the amount of *FBs* over-allocated by the overestimation. Comparing the worst-case of the overestimation in (22) with the exact value in (23), we observe that to make $E(|\hat{y}_{\text{ex}}|^2)$ equal to $E(|y_e|^2)$, $\hat{H}$ in (23) has to be reduced to $\hat{H}/4$. According to the definition of the LTI system $h[n]$ is a polynomial of a finite number of constant coefficients, e.g., $c_1, \cdots, c_L$, we can re-write $h[n]$ as

$$h[n] = \sum_{j=1}^{T} a_j \times \left(\prod_{i=1}^{L} c_i^{p_{i,j}} \right)$$

where a_j is a complex constant, which can be computed based on the particular topology of the LTI circuit, e.g., direct/parallel form IIR filters, T is the number of monomials $m_j = \prod_{i=1}^{L} c_i^{p_{i,j}}$ realizing the polynomial $h[n]$, while $p_{i,j}$ is a nonnegative integer representing the degree of the coefficient c_i in the jth monomial m_j. Consequently, $h_q[n]$ is a polynomial of L quantized coefficients $\hat{c}_1, \cdots, \hat{c}_L$, with $\hat{c}_i = c_i + e_{ci}$ $(i = 1, \cdots, L)$ and e_{ci} being the coefficient quantization error of c_i. Therefore, we can re-write the expression $h[n] - h_q[n]$ in $\hat{H}$, (22), as

$$h[n] - h_q[n] = \sum_{j=1}^{\hat{T}} (\hat{a}_j \times \hat{m}_j) \tag{24}$$

where $\hat{a}_j$ is a complex constant, $\hat{T}$ is the number of monomials $\hat{m}_j = \prod_{i=1}^{L} e_{ci}^{\hat{p}_{i,j}}$ realizing $h[n] - h_q[n]$, and $\hat{p}_{i,j}$ is a nonnegative integer representing the degree of the variable e_{ci} in the jth monomial $\hat{m}_j$. Since $h[n] = h_q[n]$, if $e_{c1} = \cdots = e_{cL} = 0$, we deduce that $h[n] - h_q[n]$ does not include a zero degree monomial, i.e., $\sum_{i=1}^{L} \hat{p}_{i,j} \neq 0$. The fact that e_{ci} is very small for high values of FB_c (FB of constant coefficients), e.g., $e_{ci} < 0.001$ for $FB_c \geq 10$, it justifies the claim that if FB_c is large enough we can ignore the monomials $\hat{m}_j$ in $h[n] - h_q[n]$ that have a degree higher than 1, i.e., $\hat{m}_j = \prod_{i=1}^{L} e_{ci}^{\hat{p}_{i,j}}$, with $\sum_{i=1}^{L} \hat{p}_{i,j} > 1$. Note that monomials $m_{\text{deg-high}}$ are much smaller compared to the first degree monomials $m_{\text{deg\_1}}$, i.e., $m_{deg\_high} \ll m_{deg\_1}$. Hence, $h[n] - h_q[n]$ can be estimated as

$$h[n] - h_q[n] \approx b_1 e_{c1} + b_2 e_{c2} + \cdots + b_L e_{cL} \tag{25}$$

where b_j $(j = 1, \cdots, L)$ is a constant complex number, which can be obtained using a first-order Taylor approximation of (24) w.r.t. e_{cj} $(j = 1, \cdots, L)$. It can be observed from (25) that if the FBs of all coefficients are increased by 1, then all e_{ci} values are approximately divided by 2 and consequently, $\hat{H}$ is divided by 4. Hence, the overestimation of our analysis results in allocating approximately one additional fractional bit to coefficients.

LTI_Precision_FB (E_{max}, H, Rx)
{ /* *Inputs: Impulse Response H, Error bound* E_{max}
Range of Input x: $Rx = [x_{low}, x_{upp}]$
Outputs: FBs, maximum imprecision MM */
1. $FB_c = 1$; // *minimum point to find the best FB for coefficients*
3. while (1) //*1st while loop: initially sets* FB_c
4. { FB_c++; //*Computation of MM using Eqns. (7), (8), (14)*
5. MM = Compute_MM ($H, Rx, FB_c, e_{in} = e_{q1} = \ldots = e_{qN} = 0$);
6. If ($MM < E_{max}$) break; }
//*Using Eqn. (16) to set the FBs of input and intermediate signals*
7. while (1) //*2nd while loop*
8. { $E_{q\_max} = E_{max} - MM$;
9. Set FB_{in} so that: $\max(y_{e0}) \leq E_{q\_max}/(N+1)$;
10. $E_{q\_max} = E_{q\_max} - \max(y_{e0})$;
11. For ($m = 1; m \leq N; m++$)
12. { Set FB_m so that: $\max(y_{em}) \leq E_{q\_max}/(N+1-m)$;
13. $E_{q\_max} = E_{q\_max} - \max(y_{em})$; }
14. If ($FB_{in} < FB_c$) & ($FB_1 < FB_c$) & … & ($FB_N < FB_c$)
15. { break; } //*Converging by breaking the 2nd while loop*
16. Else { FB_c++;
17. MM = Compute_MM ($H, Rx, FB_c, e_{in} = e_{q1:N} = 0$); }
}
18. Return $FB_{in}, FB_1, \ldots, FB_N, FB_c, MM = E_{max} - E_{q\_max}$; }

Fig. 2. Proposed algorithm for finding suitable FBs in an LTI circuit to satisfy a given error bound on MM.

B. Optimization Algorithm

The algorithm to determine FB values of both variables and constants, which satisfies a specific bound on MSE is presented in Fig. 4. The pseudo-code of the algorithm is similar to that of the precision analysis and *FB* allocation in Fig. 2. At Step 4, the value of $E(|\hat{y}_{\text{ex}}|^2)$ is computed. Since for a BIBO stable system $\lim_{n\to\infty} h[n] = \lim_{n\to\infty} h[n] = 0$, we need to choose n to be high enough to provide accurate results for $E(|\hat{y}_{\text{ex}}|^2)$. This can be done by gradually increasing n until the condition $h[n] \approx h_q[n] \approx 0$ is satisfied. As discussed in Section V-A the convergence of this process depends on the position of poles, and how close the circuit is w.r.t. the stability condition. Experimental results in Section VI show that even for benchmarks with poles very close to the unit circle, i.e., the most negative case, our algorithms converge in just a few seconds.

If, after completing one execution of the *while* loop (Step 2, Fig. 4), the value of $E(|\hat{y}_{ex}|^2)$ is higher than the maximum allowed MSE (E_{square}), then this indicates that FB_c is not high enough, and the *while* loop is re-executed increasing FB_c by 1.

TABLE II
COMPARISON OF $E(|e_{\text{uniform}}|^2)$ AND $E(|e_{\text{exact}}|^2)$ FOR AN ADDER

b	$E(\|e_{\text{uniform}}\|^2)$	$E(\|e_{\text{exact}}\|^2)$	Overestimation
6	1365.3	1333.5	2.38%
8	21 845	21 717	0.589%
9	87 381	87 122	0.297%
10	349 530	348 990	0.154%
11	1 398 100	1 396 900	0.0859%

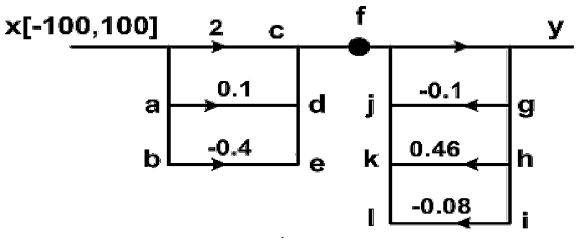


Fig. 3. Direct form third-order IIR filter for example 3.

Otherwise, a procedure based on Lemma 4 is performed to set the FB values of other variables (Steps 9–13). This process, similar to the one in Fig. 2, terminates after checking whether the final values of FBs are all smaller than FB_c. If this is the case, then the algorithm returns the FB values as well as the maximum MSE of the fixed-point circuit, i.e., MSE_{fixed}. Otherwise, the *while* loop in Step 7 is re-executed to increase FB_c and achieve lower values for FB_{in} and $FB_{1:N}$ to save hardware cost.

Note that for computations of range, MM, and MSE/SQNR, all our algorithms have the complexity of $O(n_{\max}N)$, where $n_{\max}$ is the maximum number of samples required for the algorithm to converge, and N is the number of intermediate variables. The value of $n_{\max}$ depends on the position of poles, and how close the circuit is w.r.t. the stability condition. Experimental results in Section VI show that even for specific benchmarks with poles very close to the unit circle, i.e., the most negative case, our algorithms converge in less than 1200 iterations. Note that the process of determining the FB of variables in the algorithms in Figs. 2 and 4, which is based on Lemma 4, has the complexity of $O(N)$, since for each intermediate variable, (16) can easily be utilized to find suitable fractional bit-widths.

The optimization algorithms in Figs. 2 and 4 deal with the error metrics MM and MSE/SQNR separately, since most applications require the satisfaction of either MM or MSE/SQNR. However, it is also possible to address a multiobjective optimization, such that it reaches simultaneously two separate error bounds on MM and MSE. A straightforward way to achieve this is to first find the solution for each error metric independently. Afterwards, for each variable/coefficient we choose the highest obtained fractional bit-width between the two individual solutions to guarantee that both error metrics are satisfied. This approach might not be the optimal in terms of hardware cost; however, it guarantees to satisfy both error metrics.

Example 4: Consider the IIR filter in Fig. 3. The goal is to find suitable values of FBs for the input x, the intermediate variables a to l, and the constant coefficients such that the maximum MSE of -40 dB is satisfied. Note that the condition of $MSE = -40$ dB is equivalent to setting the value of E_{square} in Fig. 4 to 0.0001. The value of MSE_{fixed} is -42.24 dB. The algorithm in Fig. 4 converges after $n = 135$ iterations. In fact after $n = 135$ iterations all the impulse response samples converge to zero, and hence, $n = 135$ is high enough to stand for the upper limit (∞) in all the sigma in Fig. 4 aiming to compute all the impulse response samples. The execution time is less than a second resulting in the following FBs for the variables:

$$FB_c = 10,\ FB_x = FB_{a-c} = 7,\ FB_{d-j} = FB_y = 9.$$

□

LTI_MSE_FB (E_{square}, H, x_{abs})
{ /* *Inputs: Impulse Response H, Error bound* E_{square}
Maximum magnitude of Input x: $x_{abs} = (\max(|x|))^2$
Outputs: FBs, maximum MSE: MSE_{fixed} */
1. $FB_c = 1$; // *start point to find the best FB for coefficients*
2. while (1) //*1st while loop: sets* FB_c
3. { FB_c++;
//Finding the MSE of y_{ex} *from Eqn. (20)*
4. $E(|\hat{y}_{ex}|) = x_{abs} \times \lim_{n\to\infty} \sum_{j=0}^{n}(h[j] - h_q[j])^2$;
5. If ($E(|\hat{y}_{ex}|^2) > E_{square}$) continue; //*Continue the 1st while*
6. Else break; }
//Using Lemma 4 to set the FBs of input and intermediate signals
7. while (1) //*2nd while loop*
8. { $MSE_{max} = E_{square} - E(|\hat{y}_{ex}|^2)$;
9. Set FB_{in} so that:
$E(|e_{in}|^2) \times \sum_{j=0}^{\infty}|h_q[j]|^2 \le MSE_{max}/(N+1)$;
10. $MSE_{max} = MSE_{max} - E\left(|e_{in}|^2\right) \times \sum_{j=0}^{\infty}|h_q[j]|^2$;
11. For ($m = 1; m \le N; m++$)
12. { Set FB_m so that:
$E\left(|e_{qm}|^2\right) \times \sum_{j=0}^{\infty}|h_{qm}[j]|^2 \le MSE_{max}/(N+1-m)$;
13. $MSE_{max} = MSE_{max} - E\left(|e_{qm}|^2\right) \times \sum_{j=0}^{\infty}|h_{qm}[j]|^2$; }
14. If ($FB_{in} < FB_c$) & ($FB_1 < FB_c$) & … & ($FB_N < FB_c$) break;
15. Return FB_{in}, FB_{1-N}, FB_c, $MSE_{fixed} = E_{square} - MSE_{max}$; }

Fig. 4. Algorithm for finding FBs for given MSE error bound.

C. Extension to Nonlinear Circuits

Some recent work, like the approach in [31], handles the accuracy analysis of nonlinear fixed-point circuits with possible feedbacks by making use of linear approximations of the error function. Using such an approximation, the analytical FB optimization technique based on Lemma 3 becomes applicable to nonlinear designs as well. However, the main drawback is that the initial linear assumption of the nonlinear circuit based on [31] may result in unrealistic analyses of error. More accurate computation of error for nonlinear circuits with possible feedbacks requires further study.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the robustness and efficiency of our range and precision analysis algorithms compared to previous work. The algorithms have been implemented in MATLAB and run on an Intel 2.8 GHz Pentium 4 with 2 GBs of main memory.

TABLE III
BENCHMARKS

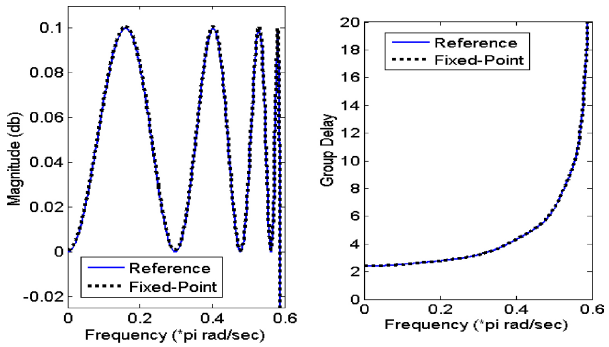
Bench	Order	Nominator Full-Precision Coefficients				Denominator Full-Precision Coefficients			
0 (HPF) [14]	2	101.8, −203.4, 101.6				1, −1.967, 0.968			
1 (LPF)	4	0.03752, 0.150086, 0.22513, 0.150086, 0.03752				1, −1.1839, 1.366039, −0.7782356, 0.2671877			
2 (HPF)	4	0.570925344, −2.280504, 3.41916, −2.2805, 0.5709253442				1, −3.07156, 3.68147, −2.03462, 0.4474244			
3 (LPF)	6	−1.5608, 3.07, 3.07, −1.5608, 1				1, −2.547, 4.2203, −4.3179, 3.0547, −1.3498, 0.3168			
4 (BPF)	8	0.024261154, 0, −0.0970446, 0, 0.145567, 0, 0.024261154, 0, −0.0970446, 0, 0.145567, 0, −0.0970446, 0, 0.02426				−0.00000000000000004441, 1.7459282353828, −0.00000000000000004441, 1.7459282353828, −0.0000000000000016653, 1.0200446024435135, 0.000000000000000053497, 0.30737575513825638			
5 (Quad BPF)	8	1, 2, 1	1, −2, 1	1, 2, 1	1, −2, 1	1, a, b	1,−a, b	1, c, d	1,−c, d
						a = 0.47583613785934908, b = 0.63399428536347535 c = 1.0921588046377746, d = 0.87447915380668007			
6 (NTSC [36])	8	Cascaded form of four 2nd-order direct-form IIR filters							

A. Benchmarks

To obtain suitable IIR benchmarks, we have generated *direct* (DR), *parallel* (PRL), and *cascade* (CS) *forms* of arbitrary order floating-point IIR filters using the *fdatool* MATLAB toolbox based on typical indicators such as sample frequencies and 3 dB bandwidths. The filters have then been optimized by our algorithms, and after finding the suitable bit-widths for fixed-point variables and constant then been optimized by our algorithms, and after finding the suitable bit-widths for fixed-point variables and constant coefficients, a register-transfer-level Verilog code is written for the fixed-point circuits. Finally, the Xilinx ISE v11 synthesis tool is utilized to map the circuits into FPGAs. The benchmarks are addressed in Table III, where HPF, LPF, and BPF refer to high-pass, low-pass, and band-pass filters. Bench#5 is a bi-quad eighth-order cascaded structure of four 2nd-order direct-form IIR filters. Hence, the nominator and denominator coefficients are separated into four subcolumns corresponding to the direct-form second-order IIR filters. The last benchmark is also a National Television Systems Committee (NTSC) channel cascaded eighth-order LPF IIR filter with the cutoff frequency of 4.74 MHz [36].

The first experiment explores the robust convergence of our algorithms (Table IV). The maximum error bound $E_{\max} = 0.1$ has been chosen, while the range of $[-100, 100]$ is selected for both, integer and fractional parts of the input x in all the test-cases in Table IV. Equation (8) and the algorithm in Fig. 2 are used to compute the range (and therefore IB) and MM with FBs setting, such that to $MM < E_{\max}$ is guaranteed. The IB of output and FB of the coefficients, input x and intermediate variables are given in Column 7, Table IV. Columns 3 and 4 indicate the number of iterations (samples n) that are required for the computation to converge. As an example, for Bench#0 it is sufficient to put the maximum number of samples as $n = 1140$ such that all the impulse response samples converge to zero. The position of the dominant poles, which are the closest ones to the unit circle, is shown in Column 8. The Bench#0 includes the dominant poles (stability condition), and as a result it requires the highest number of iterations (samples) to converge compared to the other benchmarks (Table IV).

Fig. 5 represents the frequency domain (pass-band) behavior of the reference model NTSC channel IIR filter in Bench#6, as well as its fixed-point implementation addressed in the last row of Table IV. We have also considered the case where the coefficients are rounded to the closest powers of 2 to save hardware cost. For such a case, the quantized system consists of the pole $-0.0516 \pm 1.2468j$, which is outside the unit circle and makes the filter unstable. Hence, we deduce that rounding to powers-of-two is not robust. Note that some combined quantization approaches can be investigated for coefficients to provide a trade-off between the accuracy of round-to-nearest quantization and the low hardware cost of round-to-powers-of-two technique, which are not discussed in this paper. Fig. 5(a) illustrates the magnitude of the transfer function $H(j\omega)$ in dB versus the frequency ω in rad/s. Furthermore, Fig. 5(b) depicts the group delay, i.e., $-\frac{d\{\angle\ H(j\omega)\}}{d\omega}$, versus ω, where $\angle H(j\omega)$ is the angle of $H(j\omega)$. As can be seen, the fixed-point implementation, which quantizes coefficients using round-to-nearest almost matches the reference model in terms of the frequency domain behavior. Further, the fixed-point filter also matched the behavior of its reference model in the stop-band, which is not captured in Fig. 5. The rest of fixed-point circuits in Table IV also exhibit almost the same behavior in both pass-band and stop-band frequency domains compared to their reference models.



(a) (b)

Fig. 5. Frequency domain (pass-band) behavior of Bench#6 and its fixed-point implementation. (a) Magnitude. (b) Group delay.

As discussed in Section V-A, the proposed analysis of MSE provides a safe and robust overestimation compared to the exact results. The second experiment in this section demonstrates that typically the amount of MSE overestimation is much less than 6 dB, which is the worst-case scenario as discussed in Section V-A.

TABLE IV
EXPERIMENTAL RESULTS OF EVALUATING THE PROPOSED ALGORITHMS ON SEVERAL IIR FILTER BENCHMARKS

Bench	Type	Range Iterations to Converge (# of Samples n)	Precision Iterations to Converge	Obtained Range	Obtained MM, $E_{\max}$	IB/FB_c/FB_{in}/ FB_1/FB_2/...	Dominant Poles in the Z plane	Range Runtime	Precision Runtime (s)
0	DR	1140	553	(−27 090, 27 090)	0.0826	16/26/12/18	0.9835±0.027j	0.42	0.34
1	DR	184	158	(−192, 192)	0.0972	9/13/7/12	0.1432±0.8579j	0.032	0.078
2	DR	377	298	(−293, 293)	0.0934	10/17/7/16	0.9118±0.2953j	0.078	0.172
3	DR	651	494	(−37 323, 37 323)	0.0955	17/24/13/13	0.2999±0.9212j	0.2	0.28
	PRL		387		0.0922	17/20/13/15			0.26
4	DR	276	228	(−185, 185)	0.0937	9/13/7/12	±0.5461±0.7591j	0.063	0.093
5	CS	332	210	(−7624, 7624)	0.0998	14/19/12/13/11/10/9		0.079	0.203
6	CS	379	252	(−27 512, 27 512)	0.0987	16/19/15/14/13/12/10	−0.285±0.8985j	0.094	0.17

TABLE V
COMPARISON OF DIFFERENT STATIC MSE/SQNR ANALYSES AND EXHAUSTIVE SIMULATION FOR THE MULTIPLIER $S = x \times \sqrt{2}/2$

Approach	MSE	SQNR (dB)	Runtime (s)
Simulation (exact)	1.9707×10^{-5}	39.2723	154.7
Analysis [11], [21], [26]	7.6294×10^{-6}	43.3936	<1
Our analysis	2.2331×10^{-5}	**38.7293**	<1

As computing the exact SQNR is not possible for large designs, we target a constant multiplier $S = x \times \sqrt{2}/2$, which is required for realizing an 8-point DCT or FFT. In the reference model the coefficient $\sqrt{2}/2$ has a 64-bit floating-point format. Furthermore, for the input x we have $|x| < 1$, and in the reference model comprises $FB_{x\_\text{ref}} = 23$ plus 1 sign bit. The probability of each bit in the reference model of x to be equal to 0/1 is chosen to be $\frac{1}{2}$ (uniform distribution for x). Regarding fixed-point implementation, the input and coefficient bit-widths are set to $FB_x = FB_c = 7$. The output of the fixed-point multiplier is also quantized using $FB_s = 7$. Computing the exact value of SQNR requires considering all 2^{24} possible cases of x in the reference model. Table V shows the comparison among the exact computations of MSE/SQNR using simulations, the approximate analysis based on [11], [21], [26], and our analysis. The methods in [11], [21], and [26] all underestimate the error by ignoring the coefficient quantization, which makes the analysis unsafe, with more than 4 dB of underestimation for MSE. On the other hand, our analysis overestimates the error, which is safe. Furthermore, following the discussion in Section V-A, the amount of overestimation/underestimation in the proposed MSE/SQNR analysis is much lower than the worst-case of 6 dB (less than 0.6 dB here), which indicates our tighter computation of error compared to previous work. Although simulations can be used to compute the SQNR of nonfeedback circuits like $S = x \times \sqrt{2}/2$ with few I/Os, they are not feasible to find the SQNR for circuits with feedbacks.

In the third experiment, we evaluate the impact of coefficient quantization error on the amount of unsafe overestimation of SQNR provided by the approximate analysis in [11], [21], and [26], which ignores coefficient errors, in comparison with our robust analysis. The eighth-order NTSC IIR filter in Bench#6 is chosen for this experiment, where all the input and intermediate variables are quantized with $FB = 9$, while

TABLE VI
EVALUATION OF OUR MSE/SQNR ANALYSIS AND CONVENTIONAL METHODS W.R.T. DIFFERENT VALUES OF FB_c ON BENCH#6

	SQNR (dB) with $FB = 9$				
Approach	$FB_c = 11$	$FB_c = 15$	$FB_c = 17$	$FB_c = 18$	$FB_c = 19$
Our analysis (safe)	68.5545	88.8503	95.8448	97.7169	99.249
Analysis in [11], [21], [26]	99.2906 (unsafe overestimation)				
Overestimation in nonlog domain	*741.8%*	*106.2%*	*27%*	*11.5%*	*0.29%*

FB_c varies from 11 to 19 bits. We have observed in MATLAB simulations that with the coefficient bit-width of 12, i.e., $FB_c = 11$, the position of zeros and poles in the Z-domain almost matches to the case with full-precision coefficients. Hence, to illustrate the impact of the overestimation, we consider the coefficient bit-widths that are higher than 12. The results are summarized in Table VI. The last row shows the amount of unsafe SQNR overestimation in nonlogarithmic domain originated from the analysis by conventional methods in comparison with our approach. As expected, the amount of overestimation gets lower as FB_c increases; however, this amount of unsafe overestimation is drastically high for some typical values of FB_c, i.e., $11 \leq FB_c \leq 18$, for which the position of zeros and poles in the quantized system has already been shown to be acceptable using simulations in MATLAB. The experiments in Table VI indicate that the methods, which ignore the effect of coefficient quantization, are nonrobust.

Next, in Table VII we compare our MM analysis versus methods in [14] and [15] using the IIR filters. Benchmarks 0, 2, 3 and 6 in Table III have been chosen with $E_{\max}$ set to 0.01. The solution in [14] can only support the precision analysis of second-order IIR filters, and, as a result, it cannot be used for *FB* allocation of Bench#1. Moreover, it provides the overestimations of range and precision leading to assigning additional *IB* and *FB* values. The solution in [15], which offers a more efficient analysis, is applicable to an arbitrary order IIR filter, selecting the similar *FB* values for all the variables. Our method, on the other hand, sets separate values of *FBs* for all of the above variables leading to the superior implementation in terms of area (Table VII).

In the final experiment, we address the efficiency of our MSE optimization algorithm in Fig. 4 compared to the method

TABLE VII
COMPARISON OF OUR MISMATCH ALGORITHM AND CONVENTIONAL METHODS WHEN $E = 0.01$

Bench	Ref	Range	MM (E_{max}=0.01)	IB/FB_c/FB_{in}/ FB_1/...	# Gates
0	[14]	(−27 090, 27 090)	0.0022	16/29/ 29/29	22 338
	[15]	(−13 545, 13 545)	0.0011	15/29/ 29/29	22 118
	This paper	(−13 545, 13 545)	0.008	15/29/ 15/21	**16 762**
2	[14]	(−293, 293)	?	10/?	?
	[15]	(−147, 147)	0.0089	9/20/ 20/20	16 072
	This paper	(−147, 147)	0.0083	9/21/ 9/19	**13 602**
3	[14]	(−37 323, 37 323)	?	17/?	?
	[15]	(−18 129, 19 195)	0.0047	16/24/ 24/24	13 566
	This paper	(-18 129, 19 195)	0.0093	16/24/ 17/17	**10 480**
6	[14]	(−27 512, 27 512)	?	16/?	?
	[15]	(−9072, 18 440)	0.009	16/22/ 22/22/ 22/22/ 22	27 528
	This paper	(−9072, 18 440)	0.0099	16/22/ 20/20/ 19/18/ 15	**24 550**
Average saving compared to [15]					**18%**

TABLE VIII
COMPARISON OF OUR MSE OPTIMIZATION ALGORITHM AND THE APPROACH IN [29] ON AN 8 K FFT UNIT

Parameter		FFT in [28]+ Optimization in [29]	FFT in [28] + Our Optimization
Runtime (s)		∼20	
MSE (dB)	Our analysis (safe)	26.6491	**26.6357**
	[29] (unsafe)	25.3421	25.3357
Datapath logic gates		80 387	**32 268**
Critical path (ns)		14.788	**8.774**
Input/coef bit-width		8/10	8/11
Pipeline stage bit-widths (IB+FB)		20/25/30/35/36	11/13/15/16/17

in [29]. As an example for the bit-width optimization, we consider the bit-widths of variables and constants in the 8 K FFT unit in [28]. The method in [29], which is only applicable to FFT units, ignores the effect of coefficient quantization in both accuracy analysis and optimization process. Therefore, it does not provide a robust MSE computation (Table VIII). Furthermore, it cannot provide an efficient optimization, since coefficient bit-widths are not flexible to be set. The original FFT unit in [28] makes use of 10 bit coefficients, and since the approach in [29] can only set the bit-widths of intermediate variables and does not have flexibility in controlling the bit-widths of constant coefficients, the same bit-width of 10 is used for coefficients, while the bit-widths of intermediate variables have been optimized using the method in [29]. Our optimization approach, on the other hand, provides flexibility in setting the bit-widths of both constant coefficients and variables, and hence, results in a much more efficient implementation (Table VIII). Furthermore, analysis of MSE is robust and safe. The last row of Table VIII shows the bit-widths of the intermediate pipeline stages realizing the 8 K FFT unit.

VII. CONCLUSION AND FUTURE WORK

In this paper, an efficient analysis of range and precision including MM and MSE has been presented for fixed-point LTI circuits. In general, conventional methods for analyzing MM, MSE/SQNR cannot handle the coefficient quantization errors, and hence, result in underestimations of error, which is not safe and robust. Moreover, the underestimation can be relatively large. The analyses of MM and MSE/SQNR in this paper take into account the quantization error of both constants and variables, and always overestimate the exact error, which is safe and robust. We also showed that the amount of overestimation in our precision analysis is mostly negligible, which indicates our tighter computation of error compared to previous work. Furthermore, an analytical word-length optimization, while satisfying the error metrics and avoiding the overflow, is explored. Experimental results illustrate the fast convergence, robustness and efficiency of the proposed algorithms compared to the previous work.

As the next step, we will extend the MM and MSE/SQNR analyses in this paper to focus only on a particular range of frequencies that exist in the pass-band of filters, by hybrid precision analysis in time and frequency domains. We planned to extend the analyses to handle nonlinear circuits and include the transforms that allow don't cares [37] and combinations with dynamic methods [38].

REFERENCES

[1] C. Shi and R. Brodersen, "Automated fixed-point data-type optimization tool for signal processing and communication systems," in *Proc. Des. Autom. Conf.*, 2004, pp. 478–483.
[2] Y. Pang, K. Radecka, and Z. Zilic, "An efficient hybrid engine to perform range analysis and allocate integer bit-widths for arithmetic circuits," in *Proc. ASP-DAC*, 2011, pp. 455–460.
[3] Y. Pang, K. Radecka, and Z. Zilic, "Optimization of imprecise circuits represented by Taylor series and real-valued polynomials," *IEEE Trans. Comput.-Aided Design*, vol. 29, no. 8, pp. 1177–1190, Aug. 2010.
[4] D. Lee, A. A. Gaffar, R. C. C. Cheung, O. Mencer, W. Luk, and G. A. Constantinides, "Accuracy-guaranteed bit-width optimization," *IEEE Trans. Comput.-Aided Design*, vol. 25, no. 10, pp. 1990–2000, Oct. 2006.
[5] A. B. Kinsman and N. Nicolici, "Bit-width allocation for hardware accelerators for scientific computing using SAT-modulo theory," *IEEE Trans. Comput.-Aided Design*, vol. 29, no. 3, pp. 405–413, Mar. 2010.
[6] K. Radecka and Z. Zilic, "Using arithmetic transform for verification of datapath circuits via error modeling," in *Proc. IEEE VLSI Test Symp.*, May 2000, pp. 271–277.
[7] K. Kum and W. Sung, "Combined word-length optimization and highlevel synthesis of digital signal processing systems," *IEEE Trans. Comput.-Aided Design*, vol. 20, no. 8, pp. 921–930, Aug. 2001.
[8] A. Nayak, M. Haldar, A. Choudhary, and P. Banerjee, "Precision and error analysis of MATLAB applications during automated synthesis for FPGAs," in *Proc. IEEE DATE*, Mar. 2001, pp. 722–728.
[9] A. Gaffar, O. Mencer, W. Luk, and P. Cheung, "Unifying bit-width optimization for fixed-point and floating-point designs," in *Proc. IEEE Symp. Field-Programmable Custom Comput.*, Mar. 2004, pp. 79–88.
[10] Y. Pang, K. Radecka, and Z. Zilic, "Arithmetic transforms of imprecise datapaths by Taylor series conversion," in *Proc. Int. Conf. Electron. Circuits Syst.*, 2006, pp. 696–699.

[11] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "The multiple wordlength paradigm," in *Proc. IEEE Symp. Custom Comput. Mach.*, Mar.–Apr. 2001, pp. 51–60.
[12] P. Taneli Harju, "Finite wordlength implementation of IIR polynomial predictive filters," in *Proc. Instrum. Meas. Technol. Conf.*, vol. 1. May 1997, pp. 60–65.
[13] L. D. Milic and M. D. Lutovac, "Design of multiplierless elliptic IIR filters with a small quantization error," *IEEE Trans. Signal Process.*, vol. 47, no. 2, pp. 469–479, Feb. 1999.
[14] J. Carletta, R. Veillette, F. Krach, and Z. Fang, "Determining appropriate precisions for signals in fixed-point IIR filters," in *Proc. Des. Autom. Conf.*, Jun. 2003, pp. 656–661.
[15] O. Sarbishei, Y. Pang, and K. Radecka, "Analysis of range and precision for fixed-point linear arithmetic circuits with feedbacks," in *Proc. IEEE HLDVT*, Jun. 2010, pp. 25–32.
[16] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, Oct. 1995.
[17] G. D. Kim and D. M. Chibisov, "Distribution of rounding error in multiplication of two numbers on a fixed-point computer," *Math. Notes*, vol. 1, no. 2, pp. 150–155, 1967.
[18] A. B. Sripad and D. L. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 25, no. 5, pp. 442–448, Oct. 1977.
[19] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Truncation noise in fixed-point SFGs," *Electron. Lett.*, vol. 35, no. 23, pp. 2012–2014, Nov. 1999.
[20] O. Sarbishei and K. Radecka, "Analysis of precision for scaling the intermediate variables in fixed-point arithmetic circuits," in *Proc. IEEE ICCAD*, Nov. 2010, pp. 739–745.
[21] D. Menard, R. Rocher, and O. Sentieys, "Analytical fixed-point accuracy evaluation in linear time-invariant systems," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 10, pp. 3197–3208, Nov. 2008.
[22] S. Kim, K. Kum, and S. Wonyong, "Fixed-point optimization utility for C and C++ based digital signal processing programs," *IEEE Trans. Circuits Syst. II: Analog Digital Signal Process.*, vol. 45, no. 11, pp. 1455–1464, Nov. 1998.
[23] H. Keding, F. Hurtgen, M. Willems, and M. Coors, "Transformation of floating-point into fixed-point algorithms by interpolation applying a statistical approach," in *Proc. Int. Conf. Signal Process. Appl. Technol.*, 1998, pp. 270–276.
[24] G. Constantinides, P. Cheung, and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Trans. Comput.-Aided Design*, vol. 22, no. 10, pp. 1432–1442, Oct. 2003.
[25] A. Ahmadi and M. Zwolinski, "Symbolic noise analysis approach to computational hardware optimization," in *Proc. IEEE DAC*, Jun. 2008, pp. 391–396.
[26] L. B. Jackson, "On the interaction of round-off noise and dynamic range in digital filters," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 159–184, 1970.
[27] O. Sarbishei and K. Radecka, "Analysis of mean-square-error (MSE) for fixed-point FFT units," in *Proc. IEEE ISCAS*, May 2011, pp. 1732–1735.
[28] R. M. Jiang, "An area-efficient FFT architecture for OFDM digital video broadcasting," *IEEE Trans. Consumer Electron.*, vol. 53, no. 4, pp. 1322–1326, Nov. 2007.
[29] W. H. Chang and T. Q. Nguyen, "On the fixed-point accuracy analysis of FFT algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4673–4682, Oct. 2008.
[30] A. V. Oppenheim and C. J. Weinstein, "Effects of finite register length in digital filtering and the fast Fourier transform," *Proc. IEEE*, vol. 60, no. 8, pp. 957–976, Aug. 1972.
[31] G. Caffarena, C. Carreras, J. Lopez, and A. Fernandez, "SQNR estimation of fixed-point DSP algorithms," *Int. J. Adv. Signal Process.*, vol. 2010, no. 171027, pp. 1–11, 2010.
[32] D. Menard, and O. Sentieys, "Automatic evaluation of the accuracy of fixed-point algorithms," in *Proc. IEEE DATE*, Mar. 2002, pp. 529–535.
[33] C. J. Weinstein, "Quantization effects in digital filters," MIT Lincoln Lab., Lexington, MA, Tech. Rep. 468, Nov. 1969, ASTIA doc.
[34] R. Rocher, D. Menard, O. Sentieys, and P. Scalart, "Analytical accuracy evaluation of fixed-point systems," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2007, pp. 999–1003.
[35] S. S. Bhattacharyya, E. F. Deprettere, R. Leupers, and J. Takala, "Optimization of number representation," in *Handbook of Signal Processing Systems*. New York: Springer, 2010.
[36] J. Radecki, J. Konard, J, and E. Dubois, "Design of multidimensional finite-wordlength FIR and IIR filters by simulated annealing," *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 42, no. 6, pp. 424–431, Jun. 1995.
[37] Z. Zilic and Z. Vranesic, "A multiple-valued Reed-Muller transform for incompletely specified functions," *IEEE Trans. Comput.*, vol. 44, no. 8, pp. 1012–1020, Aug. 1995.
[38] K. Radecka and Z. Zilic, "Design verification by test vectors and arithmetic transform universal test set," *IEEE Trans. Comput.*, vol. 53, no. 5, pp. 628–640, May 2004.
[39] O. Sarbishei and K. Radecka, "On the fixed-point accuracy analysis and optimization of FFT units with CORDIC multipliers," in *Proc. IEEE Symp. Comput. Arithmetic (ARITH)*, Aug. 2011, pp. 62–69.

Omid Sarbishei received the B.Sc. and M.Sc. degrees in electrical engineering from the Ferdowsi University of Mashhad, Mashhad, Iran, and the Sharif University of Technology, Tehran, Iran, in 2007 and 2009, respectively. He is currently a Ph.D. candidate with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada.

He was with the Integrated Circuits Research Center, Sharif University of Technology in 2009 to implement a DVB-T receiver system under sponsorship of the Islamic Republic of Iran Broadcasting and the Ministry of Science in Iran. He has also been a Teaching and Research Assistant. His current research interests include arithmetic circuits, formal verification, and high-level synthesis.

Mr. Sarbishei is the recipient of the prestigious Lorner Trottier and Provost's Graduate named fellowships as a Ph.D. student with the Faculty of Engineering, McGill University.

Katarzyna Radecka (S'00–M'02) received the B.Eng., M.Eng., and Ph.D. degrees from McGill University, Montreal, QC, Canada, in 1995, 1996, and 2003, respectively.

She was with Nortel, Ottawa, ON, Canada, from 1995 to 1996, with Lucent Technologies, Allentown, PA, from 1996 to 1998, and with Concordia University, Montreal, from 2002 to 2007. She is currently with the Department of Electrical and Computer Engineering, McGill University. She has published over 50 publications and has authored the book *Verification by Error Modeling: Using Testing Methods for Hardware Verification* (Norwell, MA: Kluwer). Her current research interests include arithmetic circuits, verification, and test of hardware and software.

Zeljko Zilic (S'93–M'97–SM'06) received the B.Eng. degree from the University of Zagreb, Zagreb, Croatia, and the M.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, all in electrical and computer engineering.

From 1996 to 1997, he was with Lucent Microelectronics, where as a Member of Technical Staff he was involved in the design, test, and verification of Orca FPGAs. He joined McGill University, Montreal, QC, Canada, in 1998, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He used a sabbatical leave from 2004 to 2005 to work with ST Microelectronics, Ottawa, ON. He conducts research on various aspects of the design and test of microsystems including programmable logic cores. He has published over 200 papers, 3 books, and 5 patents for which he received several awards. He has graduated over 40 M.Eng. and Ph.D. students, who have received numerous awards for their theses and have moved on to leading industrial and academic institutions upon their graduation.

Dr. Zilic has been granted the Chercheur Strategique Research Chair from the Province of Quebec. He has also been awarded the Wighton Fellowship for Laboratory Course Teaching by the Sandford Fleming Foundation and the National Council of Deans of Engineering and Applied Science. He is a Senior Member of ACM.