

Article

On the Design of an Energy Efficient Digital IIR A-Weighting Filter Using Approximate Multiplication

Ratko Pilipović ¹ , Vladimir Risojević ²  and Patricio Bulić ^{1,*} 

¹ Faculty of Computer and Information Science, University of Ljubljana, 1000 Ljubljana, Slovenia; ratko.pilipovic@fri.uni-lj.si

² Faculty of Electrical Engineering, University of Banja Luka, 78000 Banja Luka, Bosnia and Herzegovina; vladimir.risojevic@etf.unibl.org

* Correspondence: patricio.bulic@fri.uni-lj.si

Abstract: This paper presents a new A-weighting filter's design and explores the potential of using approximate multiplication for low-power digital A-weighting filter implementation. It presents a thorough analysis of the effects of approximate multiplication, coefficient quantization, the order of first-order sections in the filter's cascade, and zero-pole pairings on the frequency response of the digital A-weighting filter. The proposed A-weighting filter was implemented as a sixth-order IIR filter using approximate odd radix-4 multipliers. The proposed filter was synthesized (Verilog to GDS) using the Nangate45 cell library, and MATLAB simulations were performed to verify the designed filter's magnitude response and performance. Synthesis results indicate that the proposed design achieves nearly 70% reduction in energy (power-delay product) with a negligible deviation of the frequency response from the floating-point implementation. Experiments on acoustic noise suggest that the proposed digital A-weighting filter can be deployed in environmental noise measurement applications without any notable performance degradation.

Keywords: band-pass digital filters; A-weighting filter; approximate computing; approximate filter; noise sensing



Citation: Pilipović, R.; Risojević, V.; Bulić, P. On the Design of an Energy Efficient Digital IIR A-Weighting Filter using Approximate Multiplication. *Sensors* **2021**, *21*, 732. <https://doi.org/10.3390/s21030732>

Academic Editor: Zahir M. Hussain
Received: 25 December 2020
Accepted: 20 January 2021
Published: 22 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Noise pollution is a common problem in urban environments. Humans are continuously exposed to noise as they go about their daily lives. However, exposure to noise in the urban environment or in the workplace can be a source of discomfort leading to health-related problems such as hearing loss if the correct protective actions are not taken. In order to assess the risk of noise for humans, one course of action is to measure the noise level present in the humans' living and working environments. Typical environmental or background noise levels in residential areas range from 30 to 80 dB, and long-term exposure to sound levels over 85 dB causes hearing damage [1]. Studies [2,3] investigated the effects of exposure to office noise and showed that everyday exposure to noise disturbance affected comfort, health, and work performance.

In order to measure human exposure to noise, the measurement equipment must correlate the measured sound pressure level (SPL) to the perceived loudness level of noise using weighting filters, such as an A-weighting filter. An A-weighting filter for sound level meters is defined in the International standard IEC 61672-1 [4] and is required to assess noise levels for legislative regulations. The standard describes the A-weighting filter by tabulating frequency weighting values, and giving an analytical expression for the transfer function of the filter, but it does not define its implementation details. A-weighting is applied to the sound samples to estimate the loudness perceived by the human ear [1].

An A-weighting filter can be implemented as an analog or digital filter. Digital filters can achieve far superior results to those of analog filters, as they do not suffer from parasitic or temperature variations that affect analog filters. Besides, the implementation of

digital filters in digital systems (e.g., SoCs and MCUs), which are omnipresent in today's measurement equipment, sensor nodes, and edge devices, is straightforward.

Coefficients of digital filters obtained from transfer functions of analog filters are real numbers that require an infinite number of bits for their representations, or at least a floating-point representation in digital systems. In practical situations, it is impossible to represent a digital filter's coefficients with an infinite number of bits; hence, designers generally use fixed-point approaches to represent the filter's coefficients. Unfortunately, fixed-point designs degrade the filter frequency response and introduce a theoretical limit of the filter's performance [5]. For example, in the realization of IIR filters in digital hardware, the filter's accuracy is limited by the length of the word used to represent the coefficients and perform arithmetic operations. Additionally, due to quantization, these coefficients are not exact. Consequently, the finite-word filter's frequency response with quantized coefficients is different from the filter's frequency response with exact coefficients. On the other hand, a fixed-point digital filter design can maximize filter performance in terms of area, delay, and power consumption.

The interest in reducing the power consumption of digital filters used in edge computing and sensor networks is growing rapidly. Several techniques are used to achieve significant reductions in power consumption. One of the first papers proposing approximate processing to achieve these goals is [6]; there the authors proposed an algorithm to reduce the total switched capacitance by dynamically varying the filter order based on signal statistics. A recent trend in low-power design is approximate computing for reducing arithmetic activity and average chip power dissipation [7–10]. Multiplication represents a widespread arithmetic operation in DSP; therefore, many DSP applications can benefit from an efficient multiplier design. By relaxing the requirement for exact computation, we can design a power-efficient approximate multiplier [11–15]. The error that emerges from product approximation should be constrained to deliver acceptable results in the application. Therefore, it is essential to find a good compromise between the accuracy of a multiplier and its efficient design.

The effectiveness of approximate multipliers for achieving low-power processing motivated us to apply approximate multiplication inside the A-weighting filter. The main idea was to introduce approximate multiplication in an A-weighting IIR filter and save the area and energy while introducing a neglectable computational error. This work is motivated by some earlier works in which the approximate multiplication was used in finite impulse response (FIR) filters [16–19]. With the optimal placement of approximate multipliers inside the A-weighting filter, we anticipate that the frequency response would be almost identical to the frequency response of the A-weighting filter with exact arithmetic. In this work, we show that we can ensure the minimal influence of approximate multiplication on the performance of the A-weighting filter and achieve power-efficient processing.

The contributions of this paper can be summarized as follows:

- This paper presents a new design for an approximate low-power digital A-weighting filter implemented as a sixth-order IIR filter with approximate multipliers.
- This work provides a thorough analysis of the effects of approximate multiplication on the frequency response of an A-weighting IIR filter. We show how the optimal placement of approximate multipliers across the filter and the appropriate zero-pole pairings ensure minimal degradation of the filter's frequency response in the presence of approximate multiplication.
- Synthesis results indicate that the proposed approximate IIR filter design achieves a nearly 70% reduction in energy (power-delay product) while preserving the required accuracy.

The rest of the manuscript is organized as follows. Section 2 gives some background, and discusses related work and the state of the art. The architecture of a digital IIR A-weighting filter and the effects of coefficient quantization are discussed in Section 3. The proposed approximate multiplication suitable for use in an IIR filter's cascade is presented in Section 4. In Section 5, the impacts of zero-pole pairings and the placement of

approximate multiplication among the filter's characteristics are analyzed, followed by a description of the design of a low power digital A-weighting IIR filter using approximate multiplication. Experimental results are summarized in Section 6. Finally, the paper is concluded in Section 7.

2. Background and Related Work

2.1. Sound Level Measurement Basics

The human auditory system responds to air pressure changes, which are perceived as sound. Therefore, in order to quantify the sound level, it is convenient to measure the pressure of the sound wave at the location of the listener. The sound pressure level is computed as the root-mean-square (RMS) value of the sound pressure, p_{RMS} , relative to the reference pressure $p_0 = 20 \mu\text{Pa}$ and expressed in decibels [20].

$$SPL = 20 \log \frac{p_{RMS}}{p_0} \quad [\text{dB}]. \quad (1)$$

Reference value p_0 is chosen to be approximately the threshold of hearing at 1000 Hz, for a typical human ear. The effective sound pressure is the RMS value of the instantaneous sound pressure p over a given interval of time. The RMS value of the sound pressure is defined as

$$p_{RMS} = \sqrt{\frac{1}{T} \int_0^T p^2(\tau) d\tau}. \quad (2)$$

Since the root mean square computation in Equation (2) involves time averaging, three values for the time constant T are adopted in sound level measurements, namely, impulse (I), fast (F), and slow (S) averaging with time constants equal to 35, 125, and 1 s, respectively [20].

2.2. A-Weighting Filter

The human auditory system has a more pronounced response to signals in the frequency range between 500 and 8 kHz and is less sensitive to very low-pitch and high-pitch noises. To ensure that a sound level meter measures close to what a human hears, the correct frequency weighting related to the response of the human auditory system must be used in sound level measurement. The A-weighting filter [4] is designed with this goal in mind and subsequently has become the most commonly used frequency response in sound level meters. Despite its shortcomings, in many countries, the use of the A-weighting frequency filter is mandatory for the measurement of environmental and industrial noise and assessments of potential hearing damage and health effects of noise.

The A-weighting filter, whose magnitude response is presented in Figure 1, is a band-pass filter designed to simulate the perceived loudness of low-level tones. It progressively de-emphasizes frequencies below 1000 Hz. At 1000 Hz, the filter gain is 0 dB. Between 1000 and about 5000 Hz the signal is slightly amplified, and at about 5000 Hz and higher, the signal is attenuated. The transfer function of an analog A-weighting filter is defined in [4] as:

$$H_a(s) = \frac{4 \cdot \pi^2 \cdot 12194^2 \cdot s^4}{(s + 2\pi \cdot 20.6)^2 (s + 2\pi \cdot 12194)^2 (s + 2\pi \cdot 107.7)(s + 2\pi \cdot 739.9)} \quad (3)$$

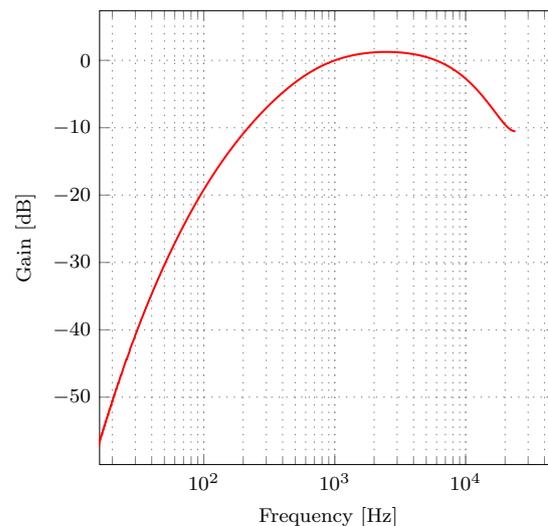


Figure 1. Magnitude response of the analog A-weighting filter given by (3).

2.3. A-Weighting Filter Design

Most of the previous work on noise measurement [21–28], has been done using the analog A-weighting filter defined by (3). Usually, such a filter consists of several active stages implemented with operational amplifiers. Hakala et al. [21] and Kivelä et al. [22,23] presented a sensor node for acoustic noise measurement which uses an analog A-weighting filter. They claim that a digital filter with real-valued coefficients involves excessive floating-point calculations, which surpasses the limit of a small, off-the-shelf integer-based MCU. Consequently, they implemented an A-weighting filter with a cascade of three analog high-pass filters and two analog low-pass filters. The paper by Rimell et al. [1] describes the implementation of the weighting filters as digital IIR filters. It provides all the necessary formulae to calculate the filter coefficients for any sampling frequency directly. The authors used a bilinear transformation to transform the analog equations that are provided in [4]. The downside of using a bilinear transform to convert an analog filter to a digital one is that the transfer function of a digital filter does not strictly follow the analog frequency response at higher frequencies. Risojević et al. [29] proposed a sensor node capable of sound level measurement based on a hardware platform with limited computational resources. Furthermore, to reduce the communication between the sensor node and a sink node and the power consumed by the IEEE 802.15.4 (ZigBee) transceiver, they performed digital A-weighting filtering on the node. The proposed digital A-weighting filter's coefficients were obtained using a matched-z transformation, and the filter was implemented as a cascade of three second-order IIR sections with quantized coefficients. In contrast to [1], Risojević et al. added a low-pass section for correction of the magnitude response at higher frequencies. In such a way, they obtained a digital filter that satisfies the tolerance limits imposed by the IEC 61672-1 standard.

2.4. Approximate Digital Filters

Many DSP applications use distributed arithmetic based approximate structures for efficient implementation of inner products. In the existing literature, most of these approximate architectures are developed by truncating the least significant bits (LSBs) of the inputs or filter coefficients [16,18,19,30,31]. As FIR filters are more tolerant towards computational errors than IIR filters, many attempts to avoid costly multiplications in FIR filters using distributed arithmetic structures have been made in the last four decades [16–19]. On the other hand, to the best of our knowledge, no attempts have been made to implement IIR filters using approximate arithmetics with coefficient quantization and finite word-length. What follows is an overview of the most recent related work on approximate filter design.

The paper [32] tries to reduce the number of adders of the multiplier block to reduce overall chip area and power consumption. It proposes a power-oriented optimization method for linear phase FIR filters. In the proposed algorithm, the average adder depth of the structural adders is used as the optimization objective in the discrete coefficients search. The authors showed that power savings could be as much as 19.6%. Kumm et al. [17] presented two novel optimization methods based on integer linear programming that minimize the number of adders used to implement a direct/transposed FIR filter. The proposed algorithms work by bounding the adder depth used for these products, which can be used to design filters for low power applications. In contrast to previous multiplier-less FIR approaches, the methods introduced in Kumm et al. [17] ensure optimal adder count. In [16], the authors proposed a fixed-point adaptive FIR filter using approximate distributed arithmetic circuits. The radix-8 Booth algorithm was used to reduce the number of partial products. Additionally, the partial products were approximately generated by truncating the input data. The proposed adaptive FIR filter was employed to identify an unknown system. The authors considered 64-tap and 128-tap FIR adaptive filters to assess the proposed design as low and high order applications. Synthesis results showed that the proposed design achieves, on average, a 55% reduction in energy.

Volkova et al. [33] proposed a generic methodology for the construction of IIR filters that behave as if the computation was performed with infinite accuracy, then converted to the low-precision output format with an error smaller than its least significant bit. This generic methodology is detailed for low-precision IIR filters in the Direct Form I implemented in FPGA logic. The authors validated the proposed methodology on a range of IIR filters. In the paper [34], an IIR filter's hardware complexity is iteratively reduced by approximating the IIR filter coefficients to maximize the number of eliminable common subexpressions. The authors showed that by using the proposed algorithm, a high-order lowpass filter with a minimum stopband attenuation of 60dB could be implemented by a 13-tap IIR filter with a group delay deviation of 0.002 only. Logic synthesis showed that the proposed IIR design saves 39.4% of the area and 41.8% of power consumption over the FIR solutions. The work in [35] proposes an IIR filter implementation considering the quantization aspect. The authors have proposed a pipelined IIR filter structure and a novel implementation of the quantizer. Finally, the work in [36] proposes fixed-point hardware architectures for IIR filters, focusing on design specifications for ECG signal processing, using the truncation error feedback to attenuate errors caused by finite word length operations inside IIR recursive structures. The proposed IIR filter architectures were described and simulated using Verilog and synthesized using the 45 nm Nangate Open Cell Library to verify the area, delay, and power metrics.

However, there is no thorough analysis of the effect of approximate multiplication, quantization, and zero-pole pairings in the IIR digital filters, as we show in this work.

3. Digital IIR A-Weighting Filter Architecture and Coefficient Quantization

A digital A-weighting filter is implemented as infinite impulse response (IIR) filter, whose output depends on a finite number of input samples and a finite number of previous filter outputs. Due to the feedback paths, IIR filters are less numerically stable than their FIR counterparts [37] but provide better performance and less computational cost than FIR filters. In this section, we explore a suitable implementation of a digital A-weighting filter and its coefficient quantization.

We follow the approach by Risojević et al. [29]. Using matched- z transformation [37] for the transfer function given in (3) of the analog A-weighting filter, and sampling frequency $F_S = 48$ kHz, the transfer function of the A-weighting digital filter is obtained as:

$$H_d(z) = \frac{(1 - z^{-1})^4}{(1 - 0.9973z^{-1})^2(1 - 0.2025z^{-1})^2(1 - 0.9860z^{-1})(1 - 0.9097z^{-1})} \quad (4)$$

The magnitude response of the filter with transfer Function (4) slightly violates the tolerance limits imposed by [4] for high frequencies. Therefore, we added a first-order

low-pass section to correct the magnitude response. The gain and cutoff frequency of the added first-order section were chosen by trial and error. The resulting transfer function is:

$$H(z) = \frac{(1 + 0.3z^{-1})(1 - z^{-1})^4}{(1 - 0.9973z^{-1})^2(1 - 0.2025z^{-1})^2(1 - 0.9860z^{-1})(1 - 0.9097z^{-1})} \quad (5)$$

The digital filter defined by (5) will be referred as a reference filter in the rest of the paper.

As can be seen from (5), the filter has poles in the unit circle's proximity, which can make the filter unstable in the presence of coefficient quantization. Risojević et al. [29] employed a cascade-form realization of the transfer function given in (5) using second-order sections (SOS) to avoid system instability due to the round-off errors in the fixed-point arithmetic. The main disadvantage of SOS filter implementation is the nonlinear relationship between the filter's coefficients and filter's poles and zeros [37]. Due to this nonlinear relationship, it is hard to determine the effect of quantization of the filter coefficients on its poles and zeros' positions and control the sensitivity of these positions to quantization errors. The SOS's nonlinear relationship between coefficients and poles motivated us to redesign the A-weighting filter as a cascade-form with the first-order sections (FOS). The filter implementation using FOS is characterized by a linear relationship between filter coefficients and its zeros and poles. Hence, we have control of the poles and zeros of the filter with quantized coefficients. Moreover, the A-weighting filter's FOS and SOS implementations have the same number of employed delay elements and arithmetic units (adders and multipliers). Factorization of the numerator and denominator polynomials in the transfer function of the A-weighting digital filter (5) yields the cascade-form implementation with FOS:

$$H(z) = H_1(z)H_2(z)H_3(z)H_4(z)H_5(z)H_6(z), \quad (6)$$

where the transfer functions of the first-order sections are:

$$\begin{aligned} H_1(z) &= \frac{1}{1 - 0.2025z^{-1}} & H_4(z) &= \frac{1 - 1.0000z^{-1}}{1 - 0.9079z^{-1}} \\ H_2(z) &= \frac{1 + 0.3000z^{-1}}{1 - 0.2025z^{-1}} & H_5(z) &= \frac{1 - 1.0000z^{-1}}{1 - 0.9973z^{-1}} \\ H_3(z) &= \frac{1 - 1.0000z^{-1}}{1 - 0.9860z^{-1}} & H_6(z) &= \frac{1 - 1.0000z^{-1}}{1 - 0.9973z^{-1}}, \end{aligned} \quad (7)$$

The proposed filter can also be represented by matrices of its coefficients as:

$$\mathcal{B} = \begin{bmatrix} 1.0000 & 0 \\ 1.0000 & 0.3000 \\ 1.0000 & -1.0000 \\ 1.0000 & -1.0000 \\ 1.0000 & -1.0000 \\ 1.0000 & -1.0000 \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} 1.0000 & -0.2025 \\ 1.0000 & -0.2025 \\ 1.0000 & -0.9860 \\ 1.0000 & -0.9079 \\ 1.0000 & -0.9973 \\ 1.0000 & -0.9973 \end{bmatrix}, \quad (8)$$

where the position of the coefficients (i.e., zeros and poles) within the matrices represents the placement of FOS. Cascade filter realizations can be obtained by different pole-zero pairings and by different orderings of sections. In floating-point arithmetic, pole-zero pairings and the order of sections in the cascade do not affect the filter's frequency response. However, when the filter is applied in digital electronics using the finite number of bits to represent the filter's coefficients and in the presence of approximate multiplication, we cannot presume that the filter's frequency response is unaffected by pole-zero pairings, the ordering of FOS in the cascade and approximate arithmetics.

We tackle this problem in Section 5. Here, we present the proposed quantization used to determine the minimal amount of bits required to represent the filter coefficients

without violating the tolerance limits imposed by the IEC 61672-1 standard. We perform quantization as follows:

$$\beta_{iq} = \frac{\text{round}(\beta_i \cdot 2^Q)}{2^Q}, \quad \alpha_{iq} = \frac{\text{round}(\alpha_i \cdot 2^Q)}{2^Q}, \quad (9)$$

where $\text{round}()$ represents rounding to the nearest integer, β_{iq} and α_{iq} represent quantized coefficients obtained from α_i and β_i , and Q denotes the number of bits used to represent the decimal part of the filter coefficients.

The magnitude responses of the A-weighting filter for different values of Q are depicted in Figure 2. When $Q = 8$, the filter's frequency response violates the IEC 61672-1 standard's tolerance limits, but only in a narrow frequency range from 10 to 100 Hz. For $Q = 9$, the filter's magnitude response has a 0.3 dB higher magnitude response than upper tolerance limits for frequencies smaller than 20 Hz. Finally, an A-weighting filter with $Q = 10$ satisfies the tolerance limits imposed by the IEC 61672-1 standard. Therefore, we represent the coefficients with 11 bits in the two's complement fixed-point format. The quantized filter coefficients for all six FOS of the A-weighting filter multiplied by 2^{10} are:

$$B = \begin{bmatrix} 1024 & 0 \\ 1024 & 307 \\ 1024 & -1024 \\ 1024 & -1024 \\ 1024 & -1024 \\ 1024 & -1024 \end{bmatrix}, \quad A = \begin{bmatrix} 1024 & -207 \\ 1024 & -207 \\ 1024 & -1010 \\ 1024 & -932 \\ 1024 & -1021 \\ 1024 & -1021 \end{bmatrix}. \quad (10)$$

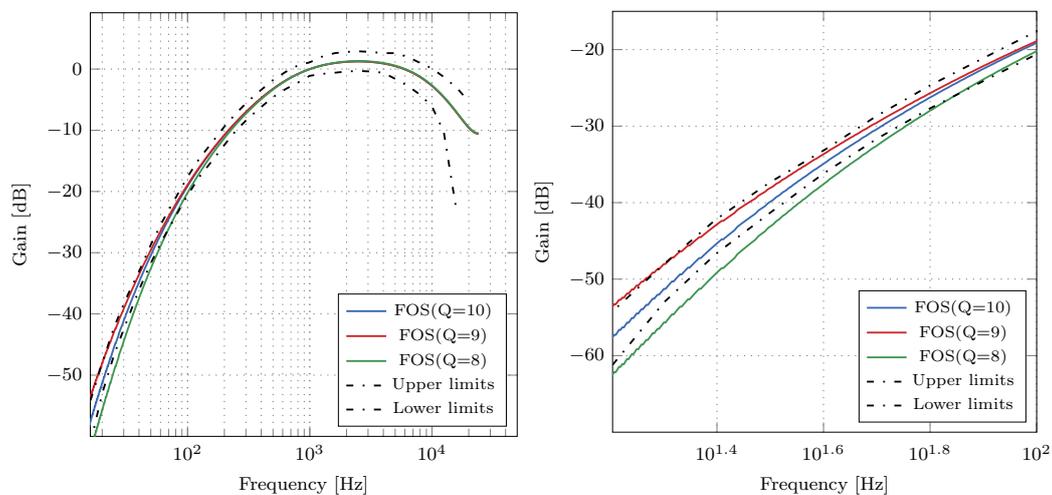


Figure 2. Magnitude responses of the digital A-weighting filter for various values of Q . **Left:** magnitude responses; **right:** enlarged portion of the magnitude responses. The filter with $Q = 10$ satisfies the tolerance limits for the A-weighting filter.

4. The Proposed Approximate Multiplication

In digital filters, the multipliers represent indispensable components that have a strong influence on their area, delay, and energy. If we employ approximate multiplier in digital filters, we can significantly improve energy consumption and area usage. Low energy consumption is a desired property as A-weighting filters are often employed as a part of battery-powered devices. However, approximate multiplication can significantly influence the A-weighting filter's stability and magnitude response. Hence, careful design and placement of approximate multipliers are required. This section first presents an exact multiplier whose design leverages the coefficient's quantization and then proposes an approximate multiplier, which we obtain by simplifying the exact multiplier.

4.1. Exact Radix-4 Multiplier

A radix-4 Booth multiplier [38] consists of two stages: a partial product generation, and a partial product addition stage. Let us illustrate radix-4 Booth encoding for the multiplication of two n -bit integers, i.e., a multiplicand X and multiplier Y in two's complement:

$$X = -x_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i \quad , \quad (11)$$

and

$$Y = -y_{n-1} \cdot 2^{n-1} + \sum_{j=0}^{n-2} y_j \cdot 2^j \quad , \quad (12)$$

where x_i and y_j represent the bits from X and Y , respectively. In the radix-4 Booth encoding, the multiplier Y is divided into overlapping groups of three bits:

$$Y = \sum_{j=0}^{\lceil n/2 \rceil - 1} \hat{y}_j^{R4} \cdot 4^j \quad , \quad (13)$$

where

$$\hat{y}_j^{R4} = -2y_{2j+1} + y_{2j} + y_{2j-1}, \quad \hat{y}_j^{R4} \in \{0, \pm 1, \pm 2\} \quad . \quad (14)$$

Taking into account the radix-4 encoding of Y , we can write the product $P = X \cdot Y$ as:

$$P = \sum_{j=0}^{\lceil n/2 \rceil - 1} PP_j \cdot 4^j \quad , \quad (15)$$

where PP_j represents j -th partial product generated from \hat{y}_j^{R4} group encoding:

$$PP_j = X \cdot \hat{y}_j^{R4}, \quad PP_j \in \{0, \pm X, \pm 2X\} \quad . \quad (16)$$

The previous discussion deals with the general case of an n -bit multiplier. In our case, the filter coefficients of the A-weighting filter are represented with 11-bit integers. If we observe filter coefficients as Y input, the resulting radix-4 Booth multiplier generates six partial products, as shown in Figure 3a). As we can see, the partial product generation stage consists of six Booth encoders, which generate partial products from each radix-4 group \hat{y}_j^{R4} . In the partial product addition stage, we employ the Wallace tree [39] to reduce the number of partial products to two. The final partial product addition is implemented using a prefix (fast) adder [38].

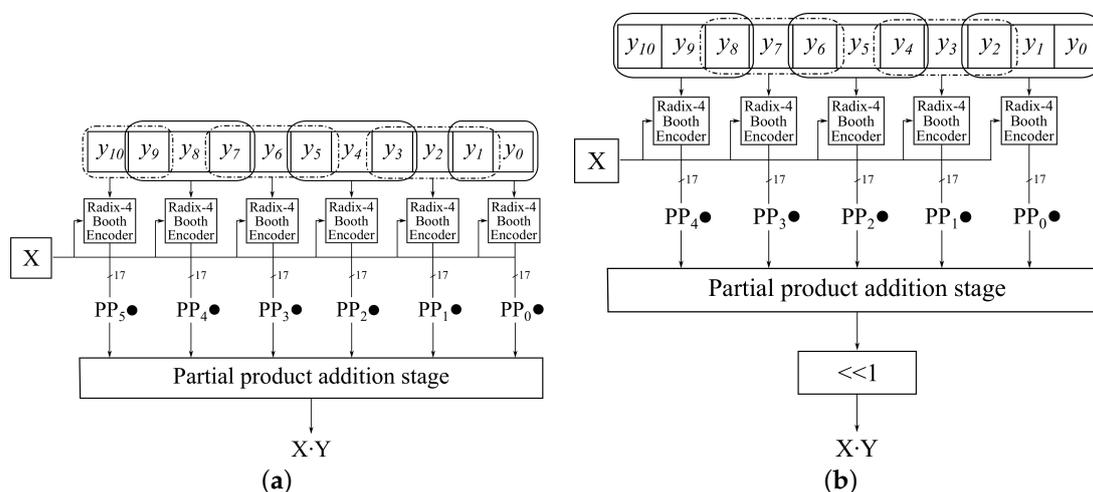


Figure 3. Exact and approximate odd radix-4 multiplier. (a) Exact radix-4 multiplier; (b) Approximate odd radix-4 multiplier.

4.2. Approximate Odd Radix-4 Multiplier

In order to reduce the number of partial products, we propose a slightly modified radix-4 encoding. The main idea behind the proposed encoding is to shift the position of group encodings one place to left, as illustrated in Figure 3b. Let:

$$\tilde{y}_j^{R4} = -2y_{2j+2} + y_{2j+1} + y_{2j} \quad (17)$$

Now, the encoded value, Y_{ODD} , of an n -bit binary number is equal to:

$$Y_{ODD} = \sum_{j=0}^{\lceil n/2 \rceil - 1} \tilde{y}_j^{R4} \cdot 4^j \quad (18)$$

In the case of an 11-bit number, the encoded value Y_{ODD} is:

$$\begin{aligned} Y_{ODD} &= \sum_{j=0}^4 \tilde{y}_j^{R4} \cdot 4^j = (-2y_{10} + y_9 + y_8)4^4 + \dots + (-2y_2 + y_1 + y_0)4^0 \\ &= -y_{10}2^9 + y_92^8 + \dots + y_22^1 + y_12^0 + y_02^0. \end{aligned} \quad (19)$$

By setting $y_02^0 = 2y_02^{-1}$, we can rewrite the above equation as:

$$\begin{aligned} Y_{ODD} &= -y_{10}2^9 + y_92^8 + \dots + y_22^1 + y_12^0 + y_02^{-1} + y_02^{-1} \\ &= (-y_{10}2^{10} + \sum_{i=0}^9 y_i2^i)/2 + y_0/2. \end{aligned} \quad (20)$$

Hence, the idea is to use the encoding from (17) and (19) to encode the multiplier Y :

$$Y = 2 \sum_{j=0}^4 \tilde{y}_j^{R4} \cdot 4^j - y_0 = 2Y_{ODD} - y_0. \quad (21)$$

In such a way, we can decrease the number of partial products by one for binary numbers with odd number of bits.

To avoid costly subtraction, which leads to a more complex circuitry, we propose to neglect the term y_0 and to approximate Y as follows

$$Y \approx \hat{Y} = 2 \sum_{j=0}^4 \tilde{y}_j^{R4} \cdot 4^j, \quad (22)$$

Section 5 shows that neglecting the term y_0 leads to an acceptable error. From (22), we can see that an error arises only when Y is an odd number.

With the proposed approximate odd radix-4 encoding, we can calculate the product $P \approx X \cdot Y_{ODD}$ as:

$$P \approx X \cdot \hat{Y} = 2 \sum_{j=0}^4 \tilde{P}_j \cdot 4^j, \quad (23)$$

where $\tilde{P}_j = X \cdot \tilde{y}_j^{R4}$ represents j -th partial product generated from \tilde{y}_j^{R4} . Note that we employ the same circuitry to obtain \tilde{y}_j^{R4} and \tilde{P}_j as in the design of exact radix-4 multiplier.

To further improve the proposed multiplier design in terms of area, delay, and energy consumption, we propose the omission of the last M bits of multiplier Y . The proposed omission also decreases the number of partial products, leading to even more hardware and energy-efficient design. For example, if $M = 5$, we omit the last two partial products in Figure 3b). Section 5 shows that this error does not affect the filter's response if we select M carefully in each first-order section.

4.3. Error Analysis of the Approximate Odd Radix-4 Multiplier

In this subsection, we present the error analysis of the approximate odd radix-4 (AO-RAD4) multiplier presented in the previous subsection. We analyze the mean relative error (MRE) and the relative error distribution for error assessment. MRE is obtained as an average relative error for all sets of inputs and all possible combinations for a $n \times 11$ bit multiplier.

The calculation of relative error for AO-RAD4 is as follows. Considering (22) and (23), the relative error of AO-RAD4 multiplier for a number pair (X, Y) is obtained as:

$$RE(X, Y) = \frac{|X \cdot \hat{Y} - X \cdot Y|}{|X \cdot Y|} = \frac{|\hat{Y} - Y|}{|Y|}, \quad (24)$$

where \hat{Y} is an approximately encoded operand as in (22). Hence, the relative error depends only on Y . The mean relative error (MRE) is calculated as follows:

$$MRE(X, Y) = \frac{1}{2^{11}} \sum_Y RE(X, Y) = \frac{1}{2^{11}} \sum_{Y=-2^{10}}^{2^{10}-1} \frac{|\hat{Y} - Y|}{|Y|}. \quad (25)$$

Figure 4 illustrates MRE (left) and error distribution (right) for different design instances of the AO-RAD4 multiplier. Error distribution is the probability that the relative error is smaller than a specific value. We can notice that MRE (Figure 4, left) increases exponentially with M . The error distribution (Figure 4, right) shows that the parameter M has a significant impact on error distribution. For example, the number of outputs whose relative error is below 0.1 decreases significantly (from 93% to 86%) when the parameter M increases from 3 to 4.

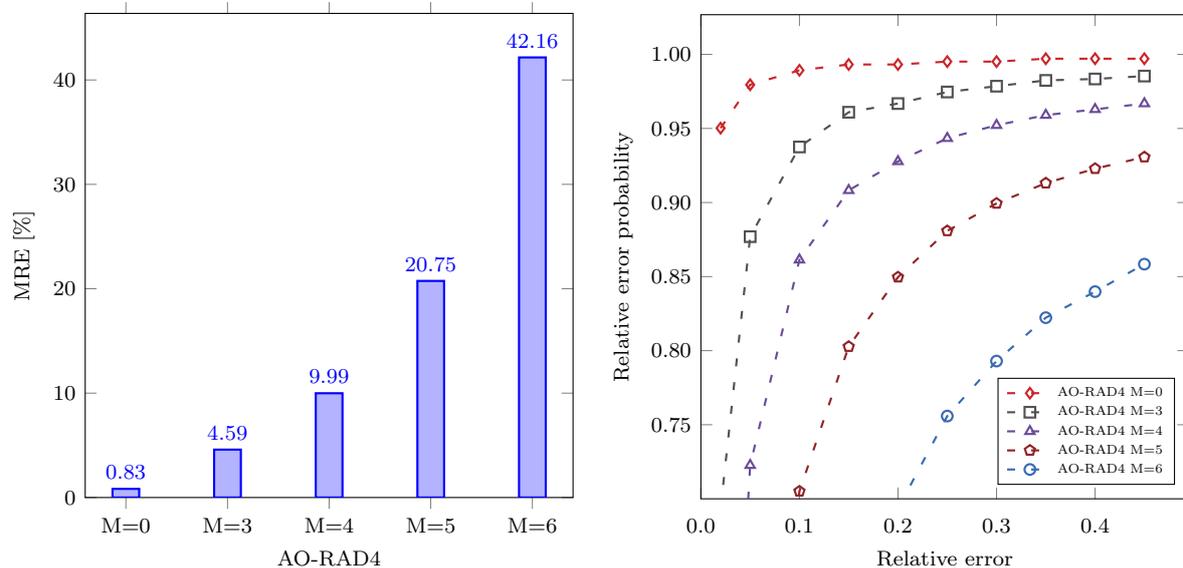


Figure 4. Error analysis of an approximate odd radix-4 (AO-RAD4) multiplier for different values of parameter M . **Left:** mean relative error (MRE); **right:** the probability that the relative error is smaller than a specific value.

5. Hardware Implementation of the Digital A-Weighting Filter with Approximate Multiplication

In this section, we assess the influence of the placement of approximate multipliers inside the digital A-weighting filter and the influence of the zero-pole pairing and ordering of FOS within the digital filter.

5.1. Influence of Approximate Multipliers Placement on the Frequency Response

Employment of approximate multiplication in the A-weighting filter requires careful placement of approximate multipliers across the FOS cascade. The simple substitution of exact multipliers with approximate ones can lead to violation of the filter's requirements or even make the system unstable.

To determine the optimal placement of the AO-RAD4 approximate multipliers within the digital A-weighting filter, we evaluated the magnitude response of the digital A-weighting filter in the presence of approximate multiplication. For every coefficient, we replaced the exact radix-4 multiplier with different instances of AO-RAD4 multiplier while keeping other multipliers exact. Then, we checked whether the proposed digital filter's magnitude response satisfies the criteria for the A-weighting filter. Moreover, we quantitatively assessed the similarity between magnitude responses of the proposed and the reference A-weighting filter, given by (5) using the cross signature scale factor (CSF) [40,41]. The CSF factor is used to quantify the amplitude difference between frequency responses. For a specific frequency ω_k , CSF is defined as:

$$CSF(\omega_k) = \frac{2|H^*(\omega_k)H_R(\omega_k)|}{|H(\omega_k)|^2 + |H_R(\omega_k)|^2}, \quad k = 1, 2, \dots, N \quad (26)$$

where $H_R(\omega_k)$ and $H(\omega_k)$, represent the reference and the proposed frequency responses at frequency ω_k , respectively, and N represents number of frequency points. The CSF ranges from 0 to 1.

Table 1 reports mean CSF for different values of the parameter M when AO-RAD4 multiplier is applied to different coefficients (factors). The combinations under which examined A-weighting filter satisfies tolerance limits for the frequency response are marked in green; otherwise, they are marked in red. As expected, the multiplications with coefficients in the FOS whose poles and zeros are further from the unit circle, are more tolerant to approximation errors and can have larger M . Now, the multiplication with the coefficients from (10) is as follows. As can be observed from Table 1, multiplication with factors 207 and 307 could be replaced with AO-RAD4 with truncation parameter $M = 6$. However, AO-RAD4 multipliers with $M = 5$ and $M = 6$ have the same number of partial products. Hence, it is better to use AO-RAD4 with $M = 5$ as it has significantly better MRE. When multiplying with 932, AO-RAD4 with $M = 5$ can be used. When multiplying with 1010, AO-RAD4 with $M = 4$ can be used. We can also see that multiplication with 1021 is very sensitive to approximation error, and we cannot use the AO-RAD4 multiplier. Hence, we employ the exact radix-4 multiplier for multiplication with 1021.

Table 1. Cross signature scale factor (CSF) for different parameter M values: combinations that satisfy tolerance for A-weighting filtering are marked in green, otherwise in red.

		AO-RAD4				
		$M = 0$	$M = 3$	$M = 4$	$M = 5$	$M = 6$
Factors	207	100.00	99.99	99.95	99.95	99.95
	307	100.00	100.00	100.00	99.99	99.93
	932	100.00	99.95	99.95	99.95	97.35
	1010	100.00	99.77	99.77	92.36	79.26
	1021	98.86	88.16	76.62	66.74	57.64

5.2. Influence of FOS Placement on the Frequency Response

In floating-point arithmetic, the position of sections in a cascade does not affect the filter's impulse response. However, we used fixed-point arithmetic combined with approximate multiplication, so we cannot presume that impulse response is unaffected by FOS's position in the cascade. To find the optimal zero-pole pairings and FOS placement, we evaluated all possible combinations of zeros and poles and the position of FOS in the

cascade. We have calculated the frequency response for every combination and compared it to the reference A-weighting filter frequency response using CSF measure. The evaluation revealed that the following FOS cascade achieves the best CSF:

$$\mathcal{B} = \begin{bmatrix} 1024 & 307 \\ 1024 & -1024 \\ 1024 & -1024 \\ 1024 & -1024 \\ 1024 & -1024 \\ 1024 & 0 \end{bmatrix} \quad \mathcal{A} = \begin{bmatrix} 1024 & -932 \\ 1024 & -1021 \\ 1024 & -1021 \\ 1024 & -1010 \\ 1024 & -207 \\ 1024 & -207 \end{bmatrix} \quad (27)$$

Finally, the proposed digital multiplier with the optimal placement of AO-RAD4 multipliers and the optimal pairings and order of FOS is presented in Figure 5.

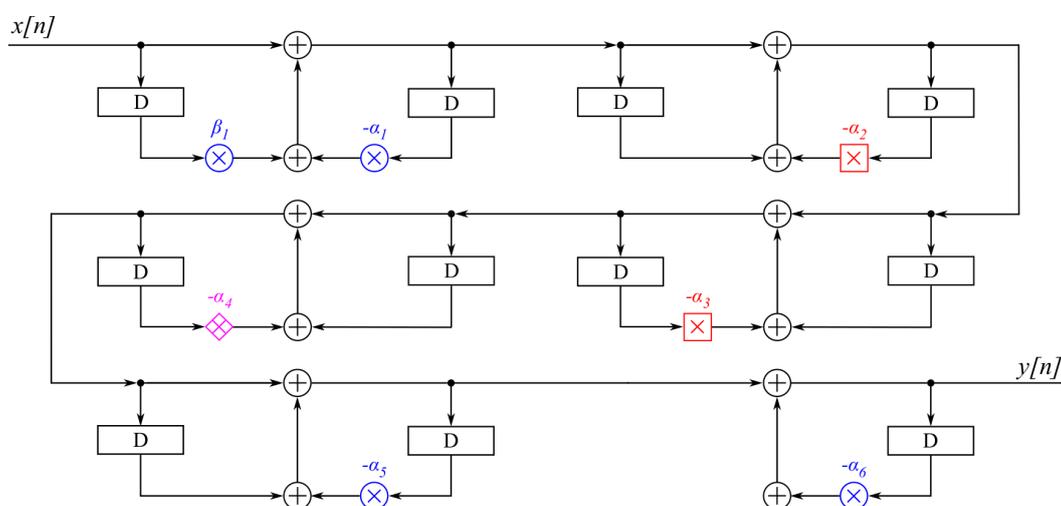


Figure 5. The proposed A-weighting filter. The \boxtimes denotes the exact radix-4 multiplier, \diamond denotes AO-RAD4 with $M = 0$, and \otimes denotes AO-RAD4 with $M = 5$. α_j and β_j represent coefficients α_1 and β_1 of the j -th first-order sections (FOS).

5.3. The Stability of Proposed Filter

In terms of poles and zeros, a digital filter is stable if and only if all poles of the filter's transfer function reside inside the unit circle in the z -plane. Two poles that correspond to coefficients with the value -1021 (27) are unaffected by approximate multiplication, as the proposed filter employs exact multiplication for these coefficients. To determine the influence of approximate multiplication on the remaining poles, we should first analyze the effects of product approximation on the coefficients. The approximate multiplication alters operand Y , and the operand X remains unchanged (see (22) and (23)). The \hat{Y} in (22) is always smaller than Y , so the approximate product is always smaller than the exact product. When we apply approximate multiplication in the filter, we select the filter's coefficients as operand Y . As we perform the exact addition, the computational error solely depends on the multiplication. The approximate multiplication leads to a decrease of coefficients, which decreases the absolute pole values and moves poles away from the unit circle. Therefore, the proposed approximate multiplication cannot lead to an unstable filter.

In addition to pole analysis, we have also evaluated the impulse response of the proposed filter. We have calculated the upper and lower impulse response envelopes using the Hilbert-transform FIR filter [42]. We chose the Hilbert-transform FIR filter to calculate the envelopes because it produces the most accurate envelope estimation. Figure 6 depicts the impulse response of the proposed filter, together with the envelopes for the first 50 samples. As we can see, both envelopes and the impulse response $h(n)$ rapidly decay to zero. From the standpoint of the impulse response, we can conclude that the proposed filter is stable.

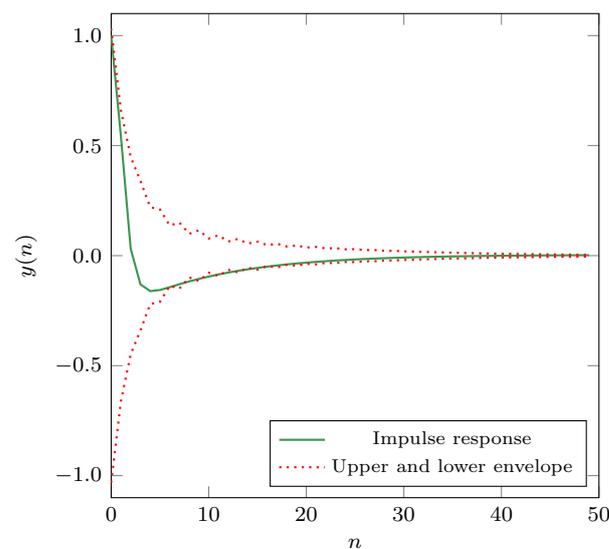


Figure 6. The impulse response of the proposed filter and its envelopes.

6. Simulation and Synthesis Results

We performed the experiments in three steps to verify the proposed approach for implementing an IIR A-weighting filter. Firstly, MATLAB simulations are described and presented to assess the fixed-point A-weighting IIR filter's behavior with and without approximate multiplication. MATLAB simulation consists of comparing the frequency responses of the filters, filtering a set of environmental noise recordings, and comparing the filters' outputs in terms of normalized root mean square error (NRMSE) and mean absolute error of sound pressure level (SPL). Secondly, we have used Verilog to implement the filters and synthesize them to 45 nm Nangate Open Cell Library. The resulting values of the area, delay, and power performance are reported. Finally, we have implemented the filter in Zynq-7000 SoC on the ZYBO Z7 FPGA development board to verify the filter's operation in a real environment.

6.1. Magnitude Response of the Proposed Digital A-Weighting Filter

In this section, we present the MATLAB simulations of the proposed and reference A-weighting filters to observe the influence of approximate multiplications on the frequency response of the filter. We observe how much the frequency response of the proposed filter deviates from the exact frequency response given in the standard.

Figure 7 shows the magnitude responses of the proposed digital A-weighting filter from Figure 5 and the reference digital A-weighting filter whose transfer function is given by (5). Note that in MATLAB simulation, we use IEEE754 double-precision format to represent the reference filter's coefficients. It can be observed from Figure 7 that the magnitude response of the proposed A-weighting filter satisfies the tolerance limits imposed by IEC 61672-1 standard. Moreover, the magnitude responses of the proposed and reference digital A-weighting filters are almost identical to each other.

To quantitatively assess the two magnitude responses, we used the CSF measure. Figure 8 shows CSF for the frequency range [10 Hz, 20 kHz]. The high values of CSF for the examined frequency range suggest that the implemented and reference A-filter have nearly identical frequency responses. For the examined frequency range, the average CSF equals 99.43 %, which indicates a high similarity between frequency responses of the reference and the proposed A-weighting filters. Therefore, we can conclude that employed approximate multipliers have a negligible influence on the filter's frequency response.

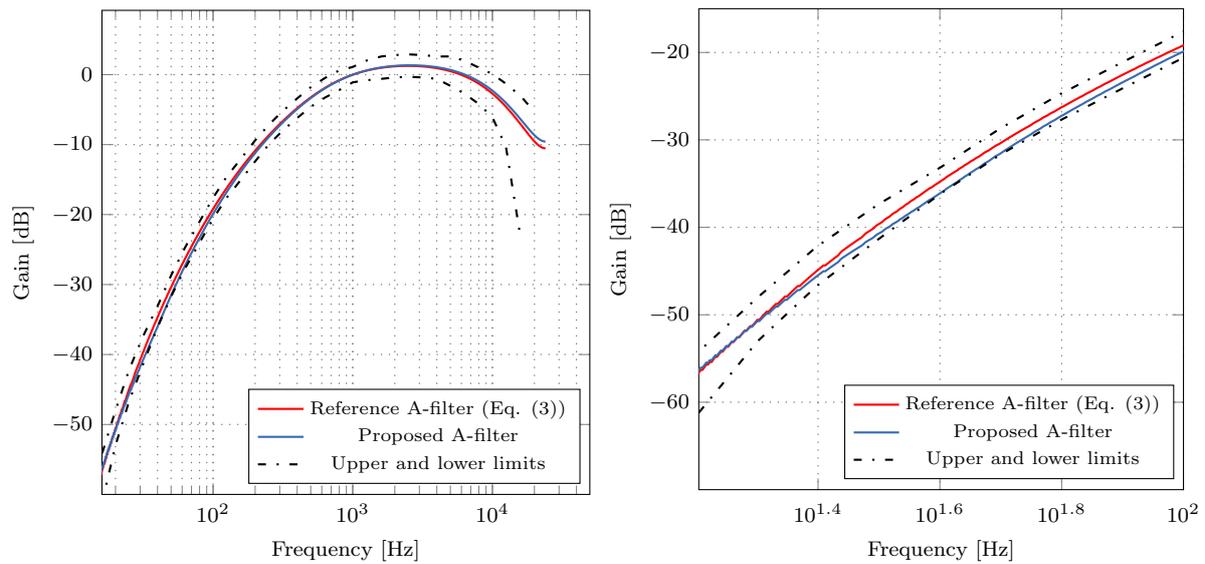


Figure 7. Magnitude responses of the proposed and reference digital A-weighting filters. **Left:** Magnitude response; **Right:** enlarged portion of the magnitude response.

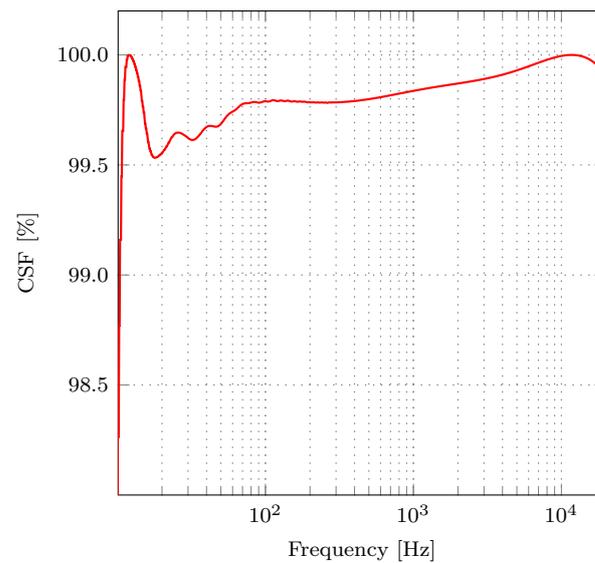


Figure 8. Cross signature scale factor (CSF) of frequency responses of the proposed and reference A-weighting filters.

6.2. Acoustic Noise Level Measurement

To assess the proposed A-weighting digital filter's performance with approximate multipliers, we used the DEMAND collection of acoustic noise in diverse environments [43,44]. For acoustic noise level measurement, we have calculated each recording's sound pressure level according to Equation (1) using fast averaging. Each recording is frequency A-weighted before we calculate the SPL value to take into account the impact of frequency on human perception of loudness. The DEMAND collection of recordings comprises four indoor environments categories, with three recordings within each category. The indoor categories are Domestic, Office, Public, and Transportation. The Domestic category consists of DKITCHEN (inside a kitchen during the preparation of food), DLIVINGR (inside a living room), and DWASHING (domestic washroom with washing machine running) recordings. The Office category consists of OHALLWAY (a hallway inside an office building with occasional traffic), OMEETING (a meeting room), and OOFFICE (a small office with three people using computers) recordings. The Public category consists of PCAFETER (a busy

office cafeteria), PRESTO (a university restaurant at lunchtime), and PSTATION (the main transfer area of a busy subway station) recordings. Finally, the Transportation category consists of the following recordings: TBUS (a public transit bus), TCAR (a private passenger vehicle), and TMETRO (a subway).

Figure 9 shows the normalized root mean square error (NRMSE) between the signal from the reference filter and the signal from the proposed filter for each of the recordings in the DEMAND collection. Normalized root mean square error is defined as:

$$NRMSE = \frac{\sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x_r[n] - x_a[n])^2}}{(x_{r,max} - x_{r,min})}, \quad (28)$$

where x_r is the signal obtained from the reference digital filter, x_a is the signal obtained from the proposed filter, $x_{r,max}$ and $x_{r,min}$ are the maximum and minimum values of the signal x_r , respectively, and N is the number of samples in each signal. It can be observed from Figure 9 that the NRMSE values between the signal from the reference filter and the signal from the proposed filter are very small. To statistically assess the range of estimates for mean NRMSE, we have calculated a 95% confidence interval (95% CI) from the obtained NRMSE on the DEMAND dataset. The CI determines the range of plausible values for mean NRMSE. The CI is calculated as follows:

$$\hat{X} \pm t_c \left(\frac{s}{\sqrt{n}} \right), \quad (29)$$

where \hat{X} represents the mean value of observed samples, t_c represents the critical value from the Student's t-distribution, s represents the standard deviation of observed samples, and n represents the number of samples. We have obtained 95% CI of $(26.85 \pm 11.28) \cdot 10^{-4}$ for the estimate of mean NRMSE. Hence, our method would exhibit NRMSE between $15.57 \cdot 10^{-4}$ and $38.13 \cdot 10^{-4}$, which implies that the proposed filter can be deployed in sound pressure level measurement without noticeable performance degradation.

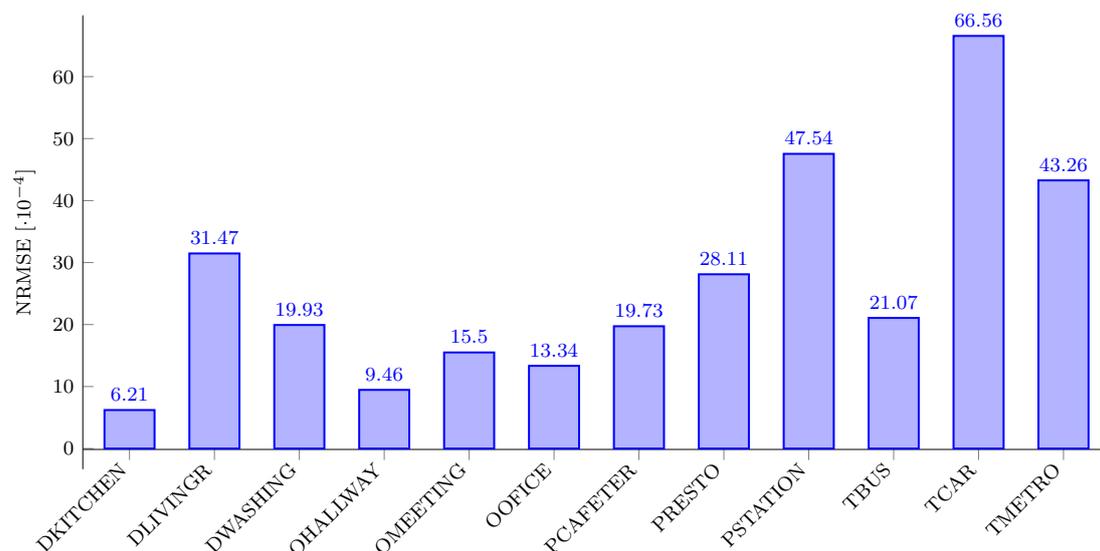


Figure 9. NRMSE between the signal from the reference filter and the signal from the proposed filter for different recordings in the DEMAND database.

We have calculated two sound pressure levels for each recording: one with the proposed and one with the reference A-weighting filter. The loudness was calculated using the “fast” response (window size of 250 ms). The mean error ($\bar{\Delta}_{SPL}$) is also reported for each recording. Figure 10 shows the loudness profiles for each of the recordings in dB SPL (A-weighted).

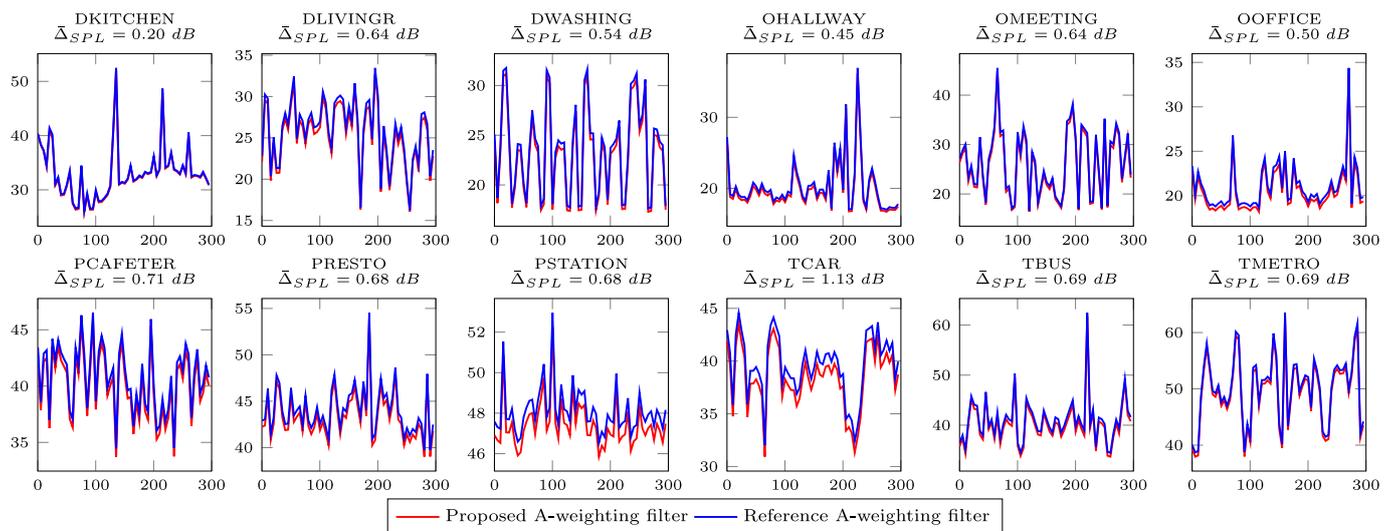


Figure 10. Sound pressure level profile for each of the recordings in the DEMAND collection. $\bar{\Delta}_{SPL}$ denotes the mean absolute error between the SPL values obtained with the proposed and the reference A-weighting filters.

As can be observed from Figures 9 and 10, NRMSE between the signal from the reference filter and the signal from the proposed filter is in strong correlation with the mean absolute error ($\bar{\Delta}_{SPL}$) between the SPL values obtained with the proposed and the reference A-weighting filters. For example, the DKITCHEN recording has the smallest NRMSE and $\bar{\Delta}_{SPL}$, and the TCAR recording has the highest NRMSE and $\bar{\Delta}_{SPL}$.

To understand the underlying distribution of Δ_{SPL} , we have calculated the histogram for the DEMAND dataset and presented it in Figure 11. From Figure 11, we can conclude that a significant amount of the Δ_{SPL} concentrates on interval [0.6,0.8] dB. Through the histogram analysis, we concluded that 91% percent of obtained Δ_{SPL} is smaller than 1 dB. Keeping in mind that professional SPL meters tend to have ± 1 dB error tolerance, these results indicate that the proposed filter offers satisfiable performance for SPL measurement. Finally, we can see that the maximal Δ_{SPL} is equal to 1.4 dB. This suggests that the proposed filter can comply with an Type 2 sound level meter [4].

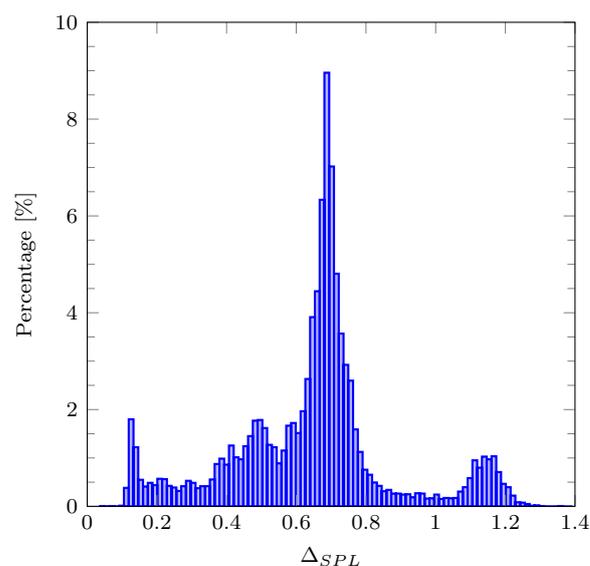


Figure 11. Histogram of the Δ_{SPL} values from the DEMAND dataset. Δ_{SPL} denotes the absolute error between the SPL values obtained with the proposed approximate and the reference A-weighting filters.

Finally, we have assessed the proposed filter's decibel range using pink noise sequences. We have generated several pink noises with different noise levels and calculated two sound pressure levels for each sequence: one with the proposed approximate and one with the reference A-weighting filter. Figure 12 shows the correlation between the noise level of pink noise and $\bar{\Delta}_{SPL}$. As we can see, the proposed filter gives satisfactory results for the examined pink noise sequences.

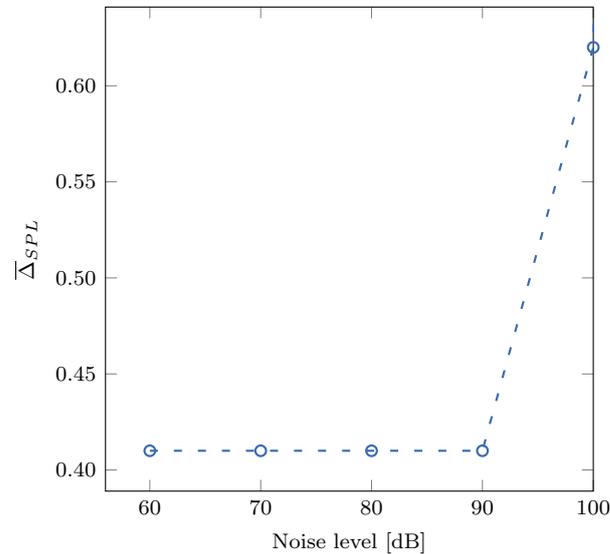


Figure 12. Correlation between the mean absolute error $\bar{\Delta}_{SPL}$ and the pink noise level.

6.3. CMOS Synthesis

In this subsection, we analyze and compare the proposed digital A-weighting filter's hardware performance in terms of power, area, delay, and power-delay-product (PDP). We compare the synthesis results of two digital A-weighting filters: the proposed digital filter with AO-RAD4 multipliers as in Figure 5, and the reference filter with exact RAD-4 multipliers (5). The filters were implemented in Verilog and synthesized to 45 nm Nangate Open Cell Library. For Verilog to GDS synthesis flow, we employed OpenROAD Flow [45], a full RTL-to-GDS flow built entirely on open-source tools. We used timing with 10 MHz virtual clocks to evaluate the power with a 5% signal toggle rate and output load capacitance equal to 10 fF. The synthesis conditions aim to compare different filters while keeping equal conditions for all experiments. The synthesis results are listed in Table 2 and consist of cell area in μm^2 , delay or critical path in nanoseconds, total power (leakage plus dynamic) in μW , and energy or power-delay-product (PDP) in fWs.

Table 2. Synthesis results.

Filter	Area [μm^2]	Delay [ns]	Power [μW]	PDP [fWs]
Proposed approximate	7588.71	1.97	203.75	401.39
Exact 32×32	18,518.92	2.39	526.19	1257.59

As can be observed from Table 2, the proposed digital filter with the approximate AO-RAD4 has substantially smaller area utilization and energy consumption compared to the reference digital filter with exact RAD-4 multipliers. The proposed filter occupies only 41% of the area of the reference filter. The power consumption for the digital filter with exact multipliers is 63% higher than the power consumption for the proposed digital filter with AO-RAD4 approximate multipliers. The proposed digital filter consumes 70% less energy (PDP) than the digital filter with exact multiplication. Besides, the proposed digital filter can process the samples 1.2 time faster.

The superior hardware performance of the proposed approximate filter originates from the usage of the approximate multipliers. The proposed approximate filter and the reference filter have the same FOS structure and the same number of arithmetic operations. Still, the former employs the approximate AO-RAD4 multipliers, and the latter employs the exact radix-4 multipliers. In this way, we achieved fair comparison and eliminated the influence of the filter structure on the synthesis results. For the exact radix-4 multiplier, the complexity of the product generation stage equals $\mathcal{O}(n^2/2)$ (n bits of multiplicand X , and $n/2$ partial products). In the case of the AO-RAD4 approximate multiplier, the complexity is equal to $\mathcal{O}(n \cdot (q - M)/2)$, where n denotes the bit width of the multiplicand, q quantization factor, and M represents the truncation parameter of approximate odd radix-4 Booth multiplier. In the proposed filter, we chose $n = 32$ bits for representing the multiplicand X , $q = 10$ for the quantization factor, and employed AO-RAD4 multipliers with $M = 4$ and $M = 5$. Therefore, the partial product stage complexity in AO-RAD4 is theoretically reduced by 80% compared to the exact radix-4 multiplier. The exact multiplier and the proposed AO-RAD4 multiplier also differ in the number of partial products. The exact radix-4 Booth multiplier has $n/2$ partial products, and the proposed approximate multiplier has $(q - M)/2$ partial products. The employed approximate multipliers with $M = 4$ and $M = 5$ have only three partial products, and the exact multiplier has 16 partial products. With fewer partial products, the approximate AO-RAD4 multiplier exhibits significantly smaller energy consumption and area utilization, which leads to an overall reduction in area and energy in the proposed filter.

To compare the Verilog model and MATLAB model outputs, we conducted the verification through FPGA prototyping. We deployed the proposed filter to the Zync 7000 SoC on the ZYBO Z7 FPGA development board. For the test inputs, we used impulse sequence and Gaussian white noise (AWGN). Figure 13 shows the filter's outputs from the filter implemented in Zync 7000 SoC in the presence of the environmental noise and MATLAB simulation model. We can notice that the outputs match, and the filter implemented in Zync 7000 SoC has the same functionality as the MATLAB model.

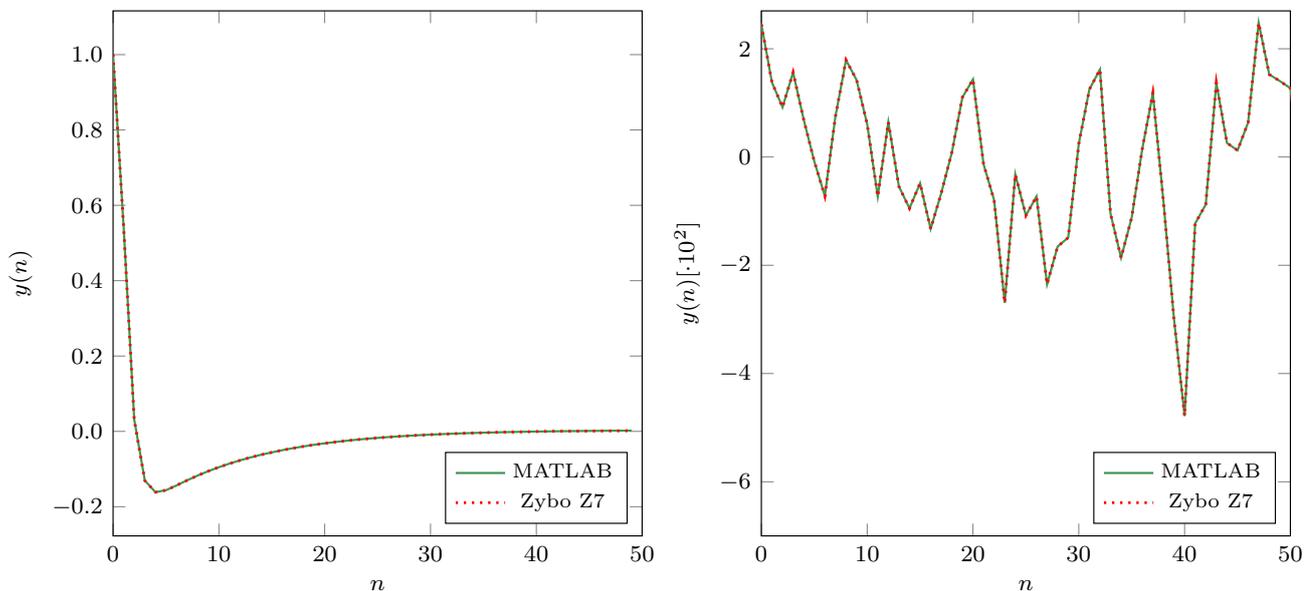


Figure 13. Filter's outputs from the Zync 7000 SoC and of the MATLAB model for two different inputs. **Left:** response to the impulse signal; **Right:** response to AWGN.

6.4. Discussion

Employment of approximate multipliers in the A-weighting IIR filter offers remarkable savings in energy consumption and area utilization, and it has a negligible impact on its accuracy. As the approximate and the reference filter have the same structure, and the

same number of first-order sections, the low area utilization and low energy consumption in the approximate filter comes solely from the employment of the approximate multipliers. The smaller number of partial products in the proposed approximate multiplier leads to a smaller circuit. Hence, the overall area and power consumption of the proposed filter have been reduced. However, careful placement of approximate multipliers in the A-weighting filter is required to meet the A-weighting filter's accuracy, stability, and frequency response. As the criteria for placement of the approximate multipliers, we selected the similarity between magnitude responses of the proposed filter and reference filter. In other words, the optimal choice and placement of the approximate multipliers in the A-weighting filter give the magnitude response, which is almost identical to the magnitude response of the referenced A-weighting filter and satisfies the IEC 61672-1 requirements. Hence, there is an insignificant difference between signals filtered with the proposed and reference filters.

As with every approximation scheme, the one proposed here also has shortcomings and limitations. The proposed approximation scheme applies only to the IIR filters that can be implemented through decomposition on the first-order sections (FOS). We selected the first-order sections as a filter building block because they have a linear relationship between coefficients and poles of a transfer function. On the other hand, we can decompose on FOS only the IIR filters with real poles or near-real poles. Besides, this study solely concentrated on deploying approximate multipliers, and the design of adders was unaltered. To further improve the proposed filters' power consumption, we need to consider the adders' design. To summarize, Figure 14 shows the design flow presented in this paper.

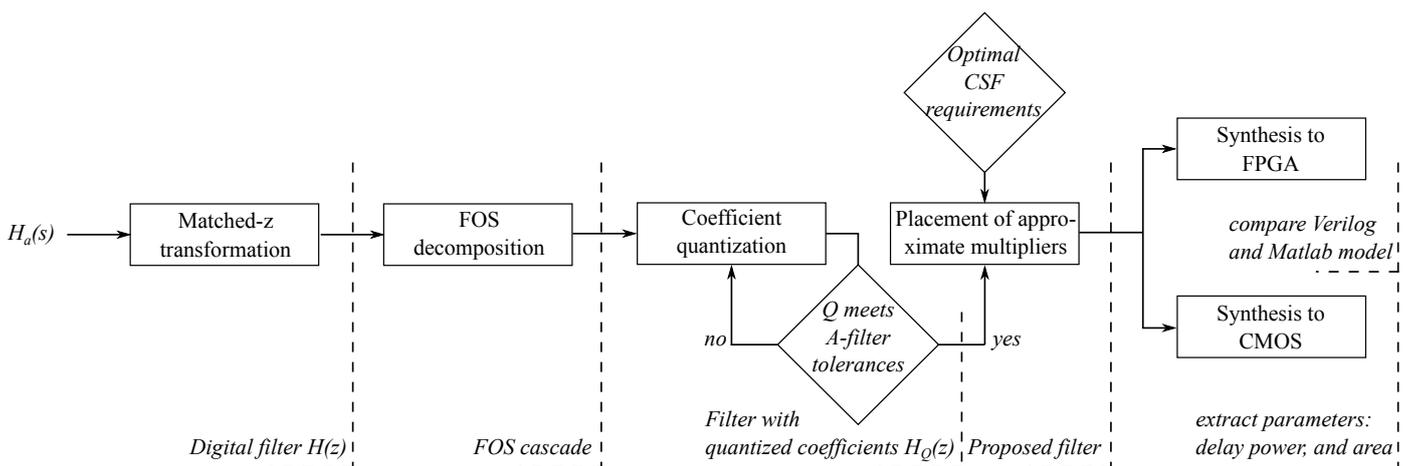


Figure 14. The proposed design flow.

7. Conclusions

In this paper we proposed an energy-efficient A-weighting IIR digital filter that uses approximate multiplications and coefficient quantization. We have thoroughly assessed the impacts of quantization, pole-zero pairings, the positions of the first-order sections in the filter's cascade, and the placement of AO-RAD4 approximate multipliers in the filter's cascade on its performance. The proposed A-weighting IIR digital filter has an almost identical frequency response to the filter with exact multipliers while consuming around 70% less energy. Experiments on acoustic noise suggest that the proposed digital A-weighting filter can be deployed in environmental noise measurement applications without any notable performance degradation. In future work, we will tackle the challenges of employing approximate arithmetic in second-order sections and extending the proposed approach to general digital IIR filter design. Further research will concentrate on the employment of error correction circuits and lowering the error caused by truncation in fixed-point arithmetic.

Author Contributions: R.P., V.R., and P.B. conceived and designed the experiments; R.P., V.R., and P.B. performed the experiments; R.P., V.R., and P.B. analyzed the data; R.P., V.R., and P.B. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Slovenian Research Agency (ARRS) under grants P2-0359 (National research program Pervasive computing) and P2-0257 (System on Chip with Integrated Optical, Magnetic and Electrochemical Sensors), and by the Slovenian Research Agency (ARRS) and Ministry of Civil Affairs, Bosnia and Herzegovina, under grant BI-BA/19-20-047 (Bilateral Collaboration Project).

Acknowledgments: We would like to thank Veselko Guštin, Dušan Kodek, and Ljubo Pipan, for encouraging us to persist in research in computer engineering and hardware. Special thanks go to our colleagues, Iztok Lebar Bajec, Uroš Lotrič, Jure Demšar, Nejc Ilc, and Branko Šter, who carefully read the manuscript and whose comments have further improved the final version of the paper. Any remaining errors are, of course, our responsibility.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SPL	Sound pressure level
DSP	Digital Signal Processing
MCU	Master Control Unit
SoC	System on Chip
IIR	Infinite Impulse Response
FIR	Finite Impulse Response
SOS	Second Order Sections
FOS	First Order Sections
AO-RAD4	Approximate Odd Radix-4
CSF	Cross Signature Scale Factor
NMRSE	Normalized Mean Root Squared Error
CI	Confidence Interval
FPGA	Field Programmable Gate Array
AWGN	Additive White Gaussian Noise

References

- Rimell, A.N.; Mansfield, N.J.; Paddan, G.S. Design of digital filters for frequency weightings (A and C) required for risk assessments of workers exposed to noise. *Ind. Health* **2015**, *53*, 21–27. [[CrossRef](#)] [[PubMed](#)]
- Toftum, J.; Lund, S.; Kristiansen, J.; Clausen, G. Effect of open-plan office noise on occupant comfort and performance. In Proceedings of the 10th International Conference on Healthy Buildings, Brisbane, Australia, 8–12 July 2012.
- Pan, N.L.; Cheung Chan, M. Study on noise perception and distraction in office. In Proceedings of the IASDR 07, Hong Kong, China, 12–15 November 2007.
- IEC. *Electroacoustics-Sound Level Meters-Part 1: Specifications (IEC61672-1)*; International Electrotechnical Commission: Genève, Switzerland, 2013.
- Kodek, D.M. Performance limit of finite wordlength FIR digital filters. *IEEE Trans. Signal Process.* **2005**, *53*, 2462–2469. [[CrossRef](#)]
- Ludwig, J.T.; Nawab, S.H.; Chandrakasan, A.P. Low-power digital filtering using approximate processing. *IEEE J. Solid State Circuits* **1996**, *31*, 395–400. [[CrossRef](#)]
- Agrawal, A.; Choi, J.; Gopalakrishnan, K.; Gupta, S.; Nair, R.; Oh, J.; Prener, D.A.; Shukla, S.; Srinivasan, V.; Sura, Z. Approximate computing: Challenges and opportunities. In Proceedings of the 2016 IEEE International Conference on Rebooting Computing (ICRC), San Diego, CA, USA, 17–19 October 2016; pp. 1–8.
- Mittal, S. A survey of techniques for approximate computing. *ACM Comput. Surv. (CSUR)* **2016**, *48*, 62. [[CrossRef](#)]
- Jerger, N.E.; Miguel, J.S. Approximate Computing. *IEEE Micro* **2018**, *38*, 8–10. [[CrossRef](#)]
- Eeckhout, L. Approximate Computing, Intelligent Computing. *IEEE Micro* **2018**, *38*, 6–7. [[CrossRef](#)]
- Kim, M.S.; Garcia, A.A.D.B.; Oliveira, L.T.; Hermida, R.; Bagherzadeh, N. Efficient Mitchell’s Approximate Log Multipliers for Convolutional Neural Networks. *IEEE Trans. Comput.* **2018**, *68*, 660–675. [[CrossRef](#)]
- Liu, W.; Xu, J.; Wang, D.; Wang, C.; Montuschi, P.; Lombardi, F. Design and Evaluation of Approximate Logarithmic Multipliers for Low Power Error-Tolerant Applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 2856–2868. [[CrossRef](#)]

13. Pilipović, R.; Bulić, P. On the Design of Logarithmic Multiplier Using Radix-4 Booth Encoding. *IEEE Access* **2020**, *8*, 64578–64590. [[CrossRef](#)]
14. Leon, V.; Zervakis, G.; Soudris, D.; Pekmestzi, K. Approximate Hybrid High Radix Encoding for Energy-Efficient Inexact Multipliers. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2018**, *26*, 421–430. [[CrossRef](#)]
15. Liu, W.; Cao, T.; Yin, P.; Zhu, Y.; Wang, C.; Swartzlander, E.E.; Lombardi, F. Design and Analysis of Approximate Redundant Binary Multipliers. *IEEE Trans. Comput.* **2019**, *68*, 804–819. [[CrossRef](#)]
16. Jiang, H.; Liu, L.; Jonker, P.P.; Elliott, D.G.; Lombardi, F.; Han, J. A High-Performance and Energy-Efficient FIR Adaptive Filter Using Approximate Distributed Arithmetic Circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *66*, 313–326. [[CrossRef](#)]
17. Kumm, M.; Volkova, A.; Filip, S.I. Design of Optimal Multiplierless FIR Filters. *arXiv* **2019**, arXiv:1912.04210.
18. Mohanty, B.K.; Meher, P.K. An Efficient Parallel DA-Based Fixed-Width Design for Approximate Inner-Product Computation. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2020**, *28*, 1221–1229. [[CrossRef](#)]
19. Ray, D.; George, N.V.; Meher, P.K. An Analytical Framework and Approximation Strategy for Efficient Implementation of Distributed Arithmetic-Based Inner-Product Architectures. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 212–224. [[CrossRef](#)]
20. Peterson, A.P.G. *Handbook of Noise Measurement*; General Radio Company: Cambridge, MA, USA, 1980.
21. Hakala, I.; Kivela, I.; Ihalainen, J.; Luomala, J.; Gao, C. Design of Low-Cost Noise Measurement Sensor Network: Sensor Function Design. In Proceedings of the 2010 First International Conference on Sensor Device Technologies and Applications, Venice, Italy, 18–25 July 2010; pp. 172–179. [[CrossRef](#)]
22. Kivela, I.; Gao, C.; Luomala, J.; Hakala, J.I.I. Design of Networked Low-Cost Wireless Noise Measurement Sensors. *Sens. Trans.* **2011**, *10*, 171–190.
23. Kivela, I.; Gao, C.; Luomala, J.; Hakala, I. Design of noise measurement sensor network: Networking and communication part. In Proceedings of the SENSORCOMM 2011: Fifth International Conference on Sensor Technologies and Applications, Nice, France, 21–27 August 2011; pp. 280–287. [[CrossRef](#)]
24. Santini, S.; Ostermaier, B.; Vitaletti, A. First Experiences Using Wireless Sensor Networks for Noise Pollution Monitoring. In *Proceedings of the Workshop on Real-World Wireless Sensor Networks*; ACM: New York, NY, USA, 2008; REALWSN '08, pp. 61–65. [[CrossRef](#)]
25. Filippini, L.; Santini, S.; Vitaletti, A. Data Collection in Wireless Sensor Networks for Noise Pollution Monitoring. In *Distributed Computing in Sensor Systems*; Nikolettseas, S.E., Chlebus, B.S., Johnson, D.B., Krishnamachari, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 492–497.
26. Tan, W.M.; Jarvis, S.A. On the design of an energy-harvesting noise-sensing WSN mote. *EURASIP J. Wirel. Commun. Netw.* **2014**, *2014*, 167. [[CrossRef](#)]
27. Segura-Garcia, J.; Felici-Castell, S.; Perez-Solano, J.J.; Cobos, M.; Navarro, J.M. Low-Cost Alternatives for Urban Noise Nuisance Monitoring Using Wireless Sensor Networks. *IEEE Sens. J.* **2015**, *15*, 836–844. [[CrossRef](#)]
28. Segura Garcia, J.; Pérez Solano, J.J.; Cobos Serrano, M.; Navarro Camba, E.A.; Felici Castell, S.; Soriano Asensi, A.; Montes Suay, F. Spatial Statistical Analysis of Urban Noise Data from a WASN Gathered by an IoT System: Application to a Small City. *Appl. Sci.* **2016**, *6*. [[CrossRef](#)]
29. Risojević, V.; Rozman, R.; Pilipović, R.; Češnovar, R.; Bulić, P. Accurate Indoor Sound Level Measurement on a Low-Power and Low-Cost Wireless Sensor Node. *Sensors* **2018**, *18*, 2351. [[CrossRef](#)]
30. Amirtharajah, R.; Chandrakasan, A.P. A micropower programmable DSP using approximate signal processing based on distributed arithmetic. *IEEE J. Solid State Circuits* **2004**, *39*, 337–347. [[CrossRef](#)]
31. Venkatachalam, S.; Ko, S. Approximate Sum-of-Products Designs Based on Distributed Arithmetic. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2018**, *26*, 1604–1608. [[CrossRef](#)]
32. Ye, W.B.; Lou, X.; Yu, Y.J. Design of Low-Power Multiplierless Linear-Phase FIR Filters. *IEEE Access* **2017**, *5*, 23466–23472. [[CrossRef](#)]
33. Volkova, A.; Istoan, M.; De Dinechin, F.; Hilaire, T. Towards Hardware IIR Filters Computing Just Right: Direct Form I Case Study. *IEEE Trans. Comput.* **2019**, *68*, 597–608. [[CrossRef](#)]
34. Deng, G.; Chen, J.; Zhang, J.; Chang, C. Area- and Power-Efficient Nearly-Linear Phase Response IIR Filter by Iterative Convex Optimization. *IEEE Access* **2019**, *7*, 22952–22965. [[CrossRef](#)]
35. Wang, Z.; Pan, C.; Song, Y.; Sechen, C. High-throughput digital IIR filter design. *J. Algorithms Optim.* **2014**, *2*, 15–27.
36. Ott, G.; Costa, E.A.C.; Almeida, S.J.M.; Fonseca, M.B. IIR Filter Architectures with Truncation Error Feedback for ECG Signal Processing. *Circuits, Syst. Signal Process.* **2019**, *38*, 329–355. [[CrossRef](#)]
37. Proakis, J.G.; Manolakis, D.K. *Digital Signal Processing*, 4th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2006.
38. Ercegovac, M.D.; Lang, T. *Digital Arithmetic*, 1st ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2003.
39. Wallace, C.S. A suggestion for a fast multiplier. *IEEE Trans. Electron. Comput.* **1964**, *EC-13*, 14–17. [[CrossRef](#)]
40. Dascotte, E.; Strobbe, J. *Updating Finite Element Models Using FRF Correlation Functions*; SPIE Proceedings Series; In Proceedings of the 17th International Modal Analysis Conference, Kissimmee, FL, USA, 8–11 February 1999; pp. 1169–1174.
41. Lee, D.; Ahn, T.S.; Kim, H.S. A metric on the similarity between two frequency response functions. *J. Sound Vib.* **2018**, *436*, 32–45. [[CrossRef](#)]
42. Oppenheim, A.V.; Buck, J.R.; Schaffer, R.W. *Discrete-Time Signal Processing*; Prentice Hall: Upper Saddle River, NJ, USA, 2001.

43. Thiemann, J.; Ito, N.; Vincent, E. The diverse environments multi-channel acoustic noise database: A database of multichannel environmental noise recordings. *J. Acoust. Soc. Am.* **2013**, *133*, 3591. [[CrossRef](#)]
44. Thiemann, J.; Ito, N.; Vincent, E. DEMAND: A Collection of Multi-Channel Recordings of Acoustic Noise in Diverse Environments. Supported by Inria under the Associate Team Program VERSAMUS. Available online: <https://zenodo.org/record/1227121#.YA1k04sRWUk> (accessed on 15 September 2020).
45. Ajayi, T.; Blaauw, D.; Chan, T.; Cheng, C.; Chhabria, V.A.; Choo, D.K.; Coltella, M.; Dreslinski, R.; Fogaça, M.; Hashemi, S.; et al. OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain. Available online: <http://people.ece.umn.edu/users/sachin/conf/gomactech19.pdf> (accessed on 20 December 2020).