IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 16, 2019, accepted February 3, 2019, date of publication February 13, 2019, date of current version March 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899107

Area- and Power-Efficient Nearly-Linear Phase Response IIR Filter by Iterative Convex Optimization

GELEI DENG[1], JIAJIA CHEN[2], JIAXUAN ZHANG[1], AND CHIP-HONG CHANG[3], (Fellow, IEEE)

[1]Department of Engineering Product Development, Singapore University of Technology and Design, Singapore 487372
[2]College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

Corresponding author: Jiajia Chen (jiajia_chen@nuaa.edu.cn)

This work was supported by the grant IDG31700104 funded by the SUTD-MIT International Design Center, Singapore.

ABSTRACT Low complexity infinite impulse response (IIR) filter design with nearly-linear phase response has attracted considerable attention in recent years due to the substantially high area and power consumption of linear phase finite impulse response (FIR) filter. Compared with the FIR filter, designing an IIR filter with the minimized group delay deviation and low power cost is a challenging topic. In this paper, the non-convex group delay deviation minimization problem for IIR filter design is reformulated into an iterative optimization problem to achieve lower group delay deviation. The hardware complexity of the solution is iteratively reduced by approximating the IIR filter coefficients to maximize the number of eliminable common subexpressions. The headroom for the coefficient adjustment is governed by the gradient of group delay deviation between iterations. Using our proposed design algorithm, a high-order lowpass filter with a minimum stopband attenuation of 60dB can be implemented by a 13-tap IIR filter with a group delay deviation of 0.002 only, as opposed to two linear-phase FIR filters designed by two recent and competitive FIR filter design algorithms with tap number of 51 and 57, respectively. Logic synthesis shows that the proposed IIR design saves 39.4% of the area and 41.8% of power consumption over the FIR solutions. Comparing with the latest nearly-linear phase IIR filter design algorithms, the group delay deviation of the solutions generated by our proposed algorithm are on average 25.5% lower, along with an average area and power savings of 20.5% and 18.4%, respectively, from the logic synthesis results.

INDEX TERMS IIR filter design, digital IC design, digital signal processing.

I. INTRODUCTION

Digital filters are widely used in digital signal processing (DSP), control and communications [1]– [3]. With the increasing number of broadband devices and new applications arising from the Internet of Things and vehicular technologies [4], increased bandwidths and improved spectral confinement are required to support low latency communications and heterogeneity of services for 5G and beyond [5]. These emerging developments call for more filters with aggressively reduced size and power consumption for signal acquisition and conditioning. Infinite impulse response (IIR) filter, with lower memory and logic cost than finite impulse response (FIR) filter [6], has resurged as a promising solution to address the more stringent expectation of area-power efficiency. Due to the non-linear phase response, IIR filter is not unconditionally stable. Nevertheless, robust stability of IIR filter can still be achieved by minimizing the group delay deviation to ensure that all frequencies within the input signal are delayed by approximately the same amount of time. Achieving a good phase response linearity without substantially increase the filter order has thus become the main focus of recent research in IIR filter design.

One way to alleviate the non-linear phase response is to cascade an all-pass phase equalizer to IIR filter [7]. The equalizer compensates the non-constant group delay so that the phase response of the filter becomes nearly linear. However, the design obtained by this method may inor-

The associate editor coordinating the review of this manuscript and approving it for publication was Yong Xiang.

dinately increases the hardware cost. Another approach is based on model reduction [8]. This method first finds an FIR solution that satisfies all the response specifications and then transforms the FIR filter into an IIR filter by model reduction technique. Non-convex optimization [9] techniques, including Fletcher-Powell optimization [10], impulse-response Gramian optimization [8], Rouche's theorem [11] and iterative optimization [12], are also popularly adopted for nearly-linear phase IIR filter design. Steiglitz-Mcbride (SM) scheme [13] is widely applied to convert the nonconvex problem into a series of iterative convex problems, and argument principle [14]– [16] is commonly utilized to fulfill the stability constraints during the optimization. The solutions obtained by the abovementioned algorithms have good phase linearity but not the lower hardware complexity, because the implementation cost is not directly incorporated into the design algorithm. Most of the latest works [17]– [19] attempt to lower the filter order while minimizing the group delay deviation. However, reduction of filter order alone does not fully harness the cost saving opportunity for physical implementation on application-specific integrated circuit (ASIC) and field programmable gate array (FPGA) platforms. Lower order filter may possess longer coefficient word length and/or reduced number of sharable common subexpressions that deprive them from greater hardware saving opportunity.

In this paper, a new design algorithm is proposed to synthesize nearly-linear phase IIR filter with reduced coefficient word length and logic complexity by incorporating full-adder cost estimate and common subexpression search into the coefficient generation and quantization process. The solution to the non-convex filter design optimization problem is approximated by the solution of a simpler iterative convex function through SM scheme. The best coefficient set from each iteration with the optimized trade-off between phase linearity and hardware implementation cost is selected for further adjustment until the increment in group delay deviation falls below a specified trust region threshold. To improve the algorithmic efficiency and solution quality, a low order IIR filter with good linearity is initialized from an FIR filter by model reduction technique. These new contributions have led to significant reduction in physical implementation costs of IIR filters in ASIC and FPGA with improved phase response linearity.

The remaining sections are organized as follows. Section II introduces the IIR filter preliminaries and the design problem formulation. Section III presents the proposed IIR filter design algorithm. The logic synthesis results and comparison are presented in Section IV. Section V concludes the paper.

II. PRELIMINARIES AND PROBLEM FORMMULATION

A. IIR FILTER PRELIMINARIES

The transfer function of an IIR digital filter is given by:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{j=0}^{M-1} b_j z^{-j}}{\sum_{i=0}^{N-1} a_i z^{-i}} \tag{1}$$

where $z = e^{j\omega}$, and $a_0 = 1$, $a_i \; \forall i = 1, 2, \cdots, N-1$ and $b_j \forall j = 0, 1, \cdots, M-1$ are the filter coefficients.

The ripple magnitude δ between the frequency response $H(e^{j\omega})$ of an IIR filter implemented with finite-precision coefficients and an ideal frequency response $\tilde{H}(e^{j\omega})$ of infinite precision coefficients can be computed by:

$$\delta(e^{j\omega}) = \left\| \left| H(e^{j\omega}) \right| - \left| \tilde{H}(e^{j\omega}) \right| \right\|_\infty \tag{2}$$

where $\|\alpha\|_\infty$ denotes the infinite norm of α. The passband and stopband of the normalized ideal magnitude response are 1 and 0, respectively.

The group delay $\tilde{\tau}$ of an IIR filter is given by

$$\tilde{\tau}(\omega) = -\frac{d}{d\omega} \angle H(e^{j\omega}) \tag{3}$$

where $\angle H(e^{j\omega})$ is the phase response of $H(e^{j\omega})$.

A filter is said to have a linear phase response if the group delay $\tilde{\tau}$ is constant. The phase response linearity can be measured by the group delay deviation $dev_\tau\ (\omega)$ from the required constant group delay τ, i.e.,

$$dev_\tau\ (\omega) = \tilde{\tau}(\omega) - \tau \tag{4}$$

To make the phase response as linear as possible, $dev_\tau(\omega)$ needs to be minimized over the passband.

To have a stable IIR filter, all poles need to fall within the unit circle of the complex plane. Hence, the maximum distance r_{max} between the pole and the origin is theoretically set to be unity. As the coefficients are truncated or rounded for fixed-point or efficient hardware implementation, the pole positions will shift and r_{max} is set to be less than unity at design time to guarantee stability. Rather than setting pole position constraints, Cauchy's argument principle [16] is more commonly used to address the stability problem in practical IIR filter design. Argument principle relates the number of poles and zeros of the denominator $A(z)$ of $H(z)$ to a contour integral of its logarithmic derivative. The IIR digital filter is stable if and only if the total change in argument of $A(z)$ is zero, which can be expressed as

$$\oint_c d\left[\arg A(z)\right] = 0 \tag{5}$$

where arg is the argument operator and $\oint_c d$ denotes the contour integral along the circle c of radius r_{max} in counter-clockwise direction around the origin.

B. PROBLEM FORMULATION

To increase the throughput, the IIR filter is usually implemented in a transposed direct form II structure as shown in Fig. 1, where $A_0, A_1, \cdots, A_{M-1}$ denote the accumulators. As the same input signal is multiplied with a group of constants in the multiplier block, the complexity of this multiple constant multiplication (MCM) block [20] can be significantly reduced by sharing of common subexpressions. The complexity of tap delay-and-accumulate (TDA) block of the IIR filter depends on the word length of the output signal from the MCM blocks. The exact number of total full adders (FAs)

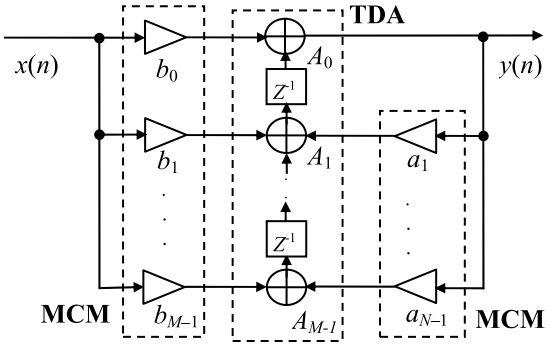

FIGURE 1. IIR filter in transposed direct form II.

TABLE 1. Notations used for specifying passband, transition band and stopband for different filter types.

	Low pass	High pass	Band pass	Band stop
Passband	$[0, \omega_3]$	$[\omega_2, \pi]$	$[\omega_1, \omega_3]$	$[0, \omega_1,]$, $[\omega_4, \pi]$
Transition band	$[\omega_3, \omega_4]$	$[\omega_1, \omega_2]$	$[\omega_1, \omega_2]$, $[\omega_3, \omega_4]$	$[\omega_1, \omega_2]$, $[\omega_3, \omega_4]$
Stopband	$[\omega_4, \pi]$	$[0, \omega_1]$	$[0, \omega_1]$, $[\omega_4, \pi]$	$[\omega_2, \omega_3]$

and registers used for the IIR filter implementation can be calculated at design time. For area-power efficiency, ripple carry adder (RCA) is used because it has the lowest area and average power dissipation [21] compared with other types of adders. The number of FAs in an RCA is proportional to its operand length. As the costs of a FA and an 1-bit register are comparable, they are assumed to be equivalent for simplicity so that the implementation cost can be approximated by the total number of FAs, C_{FA}.

Besides stability, the IIR filter must also fulfill the passband and stopband cutoff frequencies, passband ripple and stopband attenuation. The band edge frequencies for different types of filter can be fully specified by four frequency parameters ω_1, ω_2, ω_3 and ω_4, as shown in Table 1.

Let δ_p and δ_s denote the maximum allowable passband ripple and the minimum stopband attenuation, respectively. The ripple and attenuation constraints can be specified as

$$\delta(e^{j\omega}) \leq \delta_p \quad \text{for } \omega \in \text{passband} \tag{6}$$

$$\delta(e^{j\omega}) \leq 1 - \delta_s \quad \text{for } \omega \in \text{passband} \tag{7}$$

The gain of the transition band is usually limited to be less than one. This constraint can be expressed as

$$\left|H(e^{j\omega})\right| \leq 1 \quad \text{for } \omega \in \text{transitionband} \tag{8}$$

In summary, an IIR filter coefficient synthesis problem can be formulated as follows to minimize its hardware implementation cost and maximizing the phase linearity.

$$\begin{aligned} &\text{Minimize } dev_\tau(\omega), C_{FA} \\ &\text{s.t. } \delta(e^{j\omega_p}) \leq \delta_p \\ &\quad \delta(e^{j\omega_s}) \leq 1 - \delta_s \\ &\quad H(e^{j\omega_t}) \leq 1 \\ &\quad \oint_c d\left[\arg A(z)\right] = 0 \end{aligned} \tag{9}$$

where ω_p, ω_s, and ω_t are the frequencies within the passband, stopband and transition band, respectively.

Two major challenges are identified for the optimization problem in (9). Firstly, according to [17] and [22], the group delay deviation is not convex in the search space. This can be validated experimentally by calculating the derivative of group delay deviation of an arbitrary IIR filter. Multiple local maxima and minima are observed in this optimization process, which also proves its non-convexity. Secondly, the two minimization objectives, namely the group delay deviation and hardware complexity, are not independent. This further increases the challenge of finding an optimal trade-off between the two correlated optimization objectives.

III. THE PROPOSED ALGORITHM

The above two challenges of the IIR filter design problem are addressed in this section.

A. ITERATIVE CONVEX OPTIMIZATION

It is impossible to find the global minimum of a non-convex problem without exhaustive search. To search for an optimal IIR coefficient set with the minimum group delay deviation, Steiglitz-Mcbride (SM) scheme [13], which converts a non-convex optimization problem into an iterative convex optimization problem, is adopted in the proposed algorithm. SM scheme calculates the linear approximation of the objective function iteratively from the given constraints, starting with a prudently selected initial point in the solution space. It has been mathematically proved in [13] that this optimization solution is closer to the actual local minimum, compared with the initial point. The improved solution is used as a new initial point for the next iteration. This process is repeated to refine the solution towards the ideal solution of the non-convex objective function.

The following notations are used to recast the objective function to suit the SM scheme. The optimal point at the i-th iteration consists of the IIR filter coefficient set H_i expressed as $H_i = \left[b_0^{(i)}\, b_1^{(i)} \cdots b_M^{(i)}\, a_1^{(i)} a_2^{(i)} \cdots a_N^{(i)}\right]^T$, where T denotes the matrix transpose, and $a_j^{(i)} \forall j \in [1, M]$ and $b_k^{(i)} \forall k \in [1, N]$ are coefficients of H_i. In addition, the difference between the coefficients for the i-th and $(i+1)$-th iterations is denoted by $\Delta H_i = H_{i+1} - H_i$. Hence, the objective function of group delay deviation for the i-th iteration is given by:

$$\text{minimize} \left\| dev_\tau(H_i, \omega) + \nabla_H dev_\tau(H_i, \omega)^T \Delta H_i \right\|_\infty \tag{10}$$

where ∇ is the gradient operator and $dev_\tau(H_i, \omega)$ denotes the group delay deviation of H_i.

It is possible that the initial point fails to meet the constraints if the ripple of the initial solution is larger than the prescribed maximum ripple for the passband and stopband. This problem can be solved by adding convergence constant to the objective function. For the same constraints given in (9), the iterative optimization problem [23] can be formu-

lated as follows:

$$\begin{aligned}\text{Minimize: } & \left\| dev_\tau(H_i,\omega) + \nabla_H dev_\tau(H_i,\omega)^T \Delta H_i \right\|_\infty \\ & + \lambda_p \left\| \delta(e^{j\omega_p}) + \nabla_H \delta(e^{j\omega_p})^T \Delta H_i \right\|_\infty \\ & + \lambda_s \left\| \delta(e^{j\omega_s}) + \nabla_H \delta(e^{j\omega_s})^T \Delta H_i \right\|_\infty \qquad (11)\end{aligned}$$

$$\begin{aligned}\text{Subject to: } & \delta(e^{j\omega_p}) + \nabla_H \delta(e^{j\omega_p})^T \Delta H_i \le \delta_p \\ & \delta(e^{j\omega_s}) + \nabla_H \delta(e^{j\omega_s})^T \Delta H_i \le 1 - \delta_s \\ & \delta(e^{j\omega_t}) + \nabla_H \delta(e^{j\omega_t})^T \Delta H_i \le \delta_t \\ & \oint_c d\left[\arg A(z)\right] = 0; \quad \|\Delta_H\|_2 \le \rho \qquad (12)\end{aligned}$$

where λ_p and λ_s are convergence constants for the passband and stopband regions, respectively, and ρ is the trust region to limit the step size of each iteration of the optimization.

B. SELECTION OF INITIAL POINT

The final solution of iterative convex problem formulated in Section III.A is the local minimum around the initial point. Hence, selection of initial point is critical to the performance of the final result. In practical scenarios, the group delay is either fixed by the design specifications, such as IIR5 and IIR7 [9], or can simply be kept as low as possible if the group delay is not specified, such as IIR4 in [17] and IIR1 in [9]. The initialization of our algorithm takes into consideration the difference in design space for these two different types of group delay specifications encountered in the design of practical IIR filters.

1) IIR FILTER WITH A SPECIFIED GROUP DELAY

If a group delay is specified, for fast convergence of the initial solution H_1, one good solution is to start with an FIR filter, as the group delay requirement can be easily fulfilled by a symmetric FIR filter with a constant group delay. The IIR solution can then be generated as an approximation to the FIR filter. A common approach to this approximation is model reduction [24]. The model reduction approach based on Hankel-norm optimal approximation [25] is adopted in our algorithm. This algorithm approximates a high order system by a lower order system, and the error bound introduced in [25] is used to evaluate the difference between two systems, which is calculated based on the Hankel singular values [26] of the two systems in comparison. The reduced system order k is calculated by [26], and the descriptor of the input system is extracted from the state-space representation of [25]. Hankel norm transformation is applied on the descriptor to obtain the k-th order truncation system. After the transformation, the IIR filter coefficients are extracted from the equivalent state-space model of the output system, which is obtained through the Hankel matrix operations proposed in [25]. The difference between the model reduced IIR filter and the FIR filter gives rise to the group delay deviation. It has been mathematically proved in this model that the error bound can be reduced with increasing order of approximation. This implies that a lower group delay deviation can be achieved by increasing the order of IIR filter to better approximate the given FIR filter. In our algorithm, the maximum error bound is limited to e_{max}, beyond which the group delay deviation is unacceptable due to the poor approximation. The price to pay for the increased filter order is the higher hardware cost. The algorithm proposed later in Section III.D will further reduce the hardware cost, which lowers e_{max} to obtain a filter solution with higher linearity. To limit the hardware cost of the initial solution H_1, the IIR filter with the least C_{FA} is sought from the IIR coefficient sets bounded by the error between 0 and e_{max}. The pseudo code of the search for a good initial solution is shown in Fig. 2.

```
Init_IIR_τ(τ, δp, δs ω1, ω2, ω3, ω4)
    Initialize emax;  // initialize the maximum error bound
    Initialize H1; // initialize the array for IIR coefficients
    CFA = ∞; // initialize cost of H1
    NFIR = 2×τ;
    FIR = FIR_syn(NFIR, δp, δs, ω1, ω2, ω3, ω4);
    Nmin = order_estimate(FIR, emax);
    for NIIR from Nmin to NFIR
       IIR = model_reduction(FIR, NIIR);
       IIR_coef = get_coeff(IIR);
       FA = FA_count(IIR_coef);
       if FA < CFA
          CFA = FA;   H1 = IIR_coef;
       end if;
    end
   return H1;
}
```

FIGURE 2. Algorithm to search for H_1 with specified group delay.

The function **Init_IIR_**τ generates an initial solution H_1 for the convex optimization problem with a specified group delay τ, where the inputs are the cut-off frequencies, passband ripple, stopband attenuation and the group delay. The FIR filter order N_{FIR} is set to 2 τ for a group delay of τ. This is because the average group delay of the IIR filter generated by the Hankel-norm reduction is half of the original FIR filter order [26]. The function **FIR_syn** uses the Parks McClellan algorithm to synthesize an FIR filter that fulfills the frequency response specifications. The function **order_estimate**uses Hankel-norm approximation [26] to calculate the minimum order N_{min} of IIR filter that can approximate the FIR filter with an absolute error bounded by e_{max}. The filter order N_{IIR} is incremented from N_{min} to N_{FIR}. In each iteration, a candidate IIR filter of order N_{IIR} is generated from an FIR filter by the function **model_reduction**, and the function **get_coeff** extracts the coefficients of the candidate IIR filter from its corresponding state-space representation. The FA cost for the coefficient set is evaluated by **FA_count**. At the end of the iterations, the IIR filter H_1 with the least FA cost and an absolute error lower than e_{max} is returned.

2) IIR FILTER WITH NO IMPOSED GROUP DELAY

If no group delay is specified, there is more freedom to select a better initial solution H_1. Under this circumstance, the FIR filter order needs not be restricted to 2 τ, but can be selected so that the group delay deviation of the approximation IIR filter is minimized. At the same time, the filter order should be

```
Init_IIR(δp, δs, ω1, ω2, ω3, ω4) {
    Initialize emax;  // initialize the maximum error bound threshold
    Initialize H1; // initialize the array for IIR coefficients
    CFA = ∞; // initialize cost of x1
   N_lowest = PM(δp, δs, ω1, ω2, ω3, ω4);
   for NFIR = N_lowest to 2×N_lowest
      FIR = FIR_syn(NFIR, δp, δs, ω1, ω2, ω3, ω4);
      Nmin = order_estimate(FIR, emax);
      for NIIR from N_estimate to 0.5* NFIR
         IIR = model_reduction(FIR, NIIR);
         IIR_coef = get_coeff(IIR); FA = FA_count(IIR_coef);
        if FA < CFA
        CFA = FA;   H1 = IIR_coef;
       end if;
      end
    end
   return H1;
}
```

FIGURE 3. Algorithm to search for H_1 without specified group delay.

kept as low as possible to achieve low hardware complexity. The minimum FIR filter order is firstly determined by the Parks McClellan algorithm [27]. To avoid an unacceptably high order of IIR filter from being generated, the search range is set to be between the minimum FIR filter order and twice as large. For each FIR filter order within this range, one FIR filter is synthesized and the minimal order IIR filter is generated to approximate this filter with the absolute error bounded by e_{max}. The IIR filter order is then increased from the minimum until half of the order of its source FIR filter. This limit imposed on the IIR filter order prevents its complexity from growing beyond that of the source FIR filter. For each IIR filter order, an IIR filter is generated by model reduction. The coefficient set with the lowest hardware complexity is selected. The pseudo code is shown in Fig. 3. Except the function **PM**, which calculates the lowest FIR filter order *N_lowest* that fulfills the frequency response specifications by the Parks McClellan method, other used functions are the same as those in Fig. 2.

C. SOLUTION OF ITERATIVE OPTIMIZATION PROBLEM

After a good initial solution H_1 has been generated, λ_p and λ_s in (11) are set to be $k_p \times \delta(H_1, e^{j\omega_p})$ and $k_s \times \delta(H_1, e^{j\omega_s})$, respectively, where $\delta(H_1, e^{j\omega_p})$ is the passband ripple and $\delta(H_1, e^{j\omega_s})$ is the stopband ripple of the IIR filter with the coefficient set H_1. The convergence constants, k_p and k_s, for the passband and stopband, respectively are empirically determined to be between 10^4 and 10^5 for fast convergence. The trust region is initialized to ρ_{initial}.

With these parameters specified, the optimization problem (11) is solved iteratively. Each iteration generates a new solution H_i with a difference ΔH_i between the new solution H_i and its previous solution H_{i-1}. λ_p and λ_s will also be updated based on the passband and stopband ripples of the new solution H_i before the next iteration. Since the objective function of minimizing the group delay deviation is achieved by linear approximation, the trust region ρ can be adjusted after the second iteration, i.e., $i = 2$, to reduce the linear approximation error. After obtaining ΔH from the previous two iterations, the actual improvement on group delay deviation, $\|dev_\tau(H_i + \Delta H_i, \omega)\|_\infty - \|dev_\tau(H_i, \omega)\|_\infty$, is calculated and compared with the linear approximation group delay deviation improvement $\left\|\nabla_H dev_\tau(H_i, \omega)^T \Delta H_i\right\|_\infty$. The actual percentage improvement should be larger than the linear approximation percentage improvement by a fraction of at least ∇_{min}. Otherwise, the linear approximation is not accurate and the trust region ρ is reduced from ρ_{initial} to reduce the approximation error until the error becomes less than $1 - \nabla_{min}$. If the error is already less than $1 - \nabla_{min}$, ρ is enlarged to expand the search region to accelerate convergence. ρ will be reduced progressively with the number of iterations as the solution gets closer to the actual minimum. The convergence criterion is determined by the trust region threshold ρ_{th}. When ρ is less than ρ_{th}, any further modification with trust region larger than ρ_{th} will not discernibly decrease the group delay deviation.

D. REDUCTION OF HARDWARE COMPLEXITY

To maximize the number of adders shared in the implementation of MCM block, common subexpressions (CS) are detected and eliminated from the coefficient set H_i returned from the i-th iteration of the algorithm presented in Section III.C. We use CS sharing for the design of MCM block because adder-graph based techniques including [2] has a general propensity to increase the critical path delay with little or no clear advantage in hardware saving for low order filter design. The continuous coefficients are quantized and the sensitivity of each quantized coefficient to the filter response is analyzed. The quantization error is evaluated based on coefficient sensitivity so that the quantization is performed only when the quantization error is small enough to keep the frequency response in specification. Otherwise, more quantization levels are added to reduce the error to an acceptable level. Throughout the process of coefficient adjustment which will be introduced later, the frequency response is monitored to ensure that the designed specifications are fulfilled by the resulting filter. The quantized filter coefficients are then transformed into CSD representation [28] to facilitate detection and elimination of common subexpressions (CSs), and the frequencies of occurrence of eight short horizontal CSs of the forms, 101, $10\bar{1}$, $\bar{1}01$, $\bar{1}0\bar{1}$, 1001, $100\bar{1}$, $\bar{1}001$ and $\bar{1}00\bar{1}$, in the CSD coefficients are detected and eliminated by the method described in [29]. The total number of FAs that can be saved by common subexpression elimination (CSE) is given by:

$$S = \sum_{j=1}^{n} (f_j - 1) FA_j \tag{13}$$

where f_j is the frequency of occurrence of the j-th CS, FA_j is the number of FAs needed to generate the j-th CS and n is the total number of CSs of the coefficient set H_i obtained in the i-th iteration.

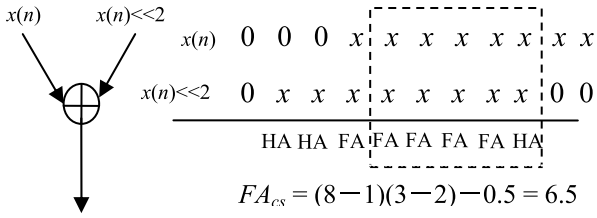

FIGURE 4. Full-adder cost of generating $x(n) \times 101_{CSD}$.

Each CS can be generated by an RCA. The addends of the RCA correspond to the product of input variable x and the weight of the two nonzero digits of the CS. Every zero digit of a CS shifts one addend left from the other by one bit, eliminating one FA from the least significant bit position and introducing one half adder (HA) towards the most significant bit position. Since the cost of a HA is approximately half the cost of a FA, the total number of FAs, FA_{cs}, required to generate a given CS can be calculated by:

$$FA_{cs} = (l_x - 1)(l_{cs} - z_{cs} - 1) - 0.5z_{cs} \tag{14}$$

where l_x is the word length of input signal, l_{cs} is the length of the CS, and z_{cs} is the number of zero bits of the CS.

Fig. 4 illustrates the calculation of FA_{cs} for the CS generated by the multiplication of 101_{CSD} with an input signal $x(n)$ of length 8.

From (14), it is evident that FA_{cs} increases with l_x and decreases with z_{cs}. For a given filter order and coefficient word length, the total hardware cost can be reduced by increasing either f_i or FA_i or both by modifying the coefficients of H_i in each iteration. However, unconditional modification of coefficients may increase the group delay deviation. To increase CS sharing with minimal increment in group delay deviation, we maximize the ratio η of FA cost reduction to group delay deviation increment defined in (15).

$$\eta = \frac{S_{pos} - S_{pre}}{\Delta \left\| dev_\tau(\omega) \right\|_\infty} \tag{15}$$

where S_{pre} and S_{pos} are the number of FAs reduced by CS sharing before and after the adjustment of coefficients, respectively. $\Delta \left\| dev_\tau(\omega) \right\|_\infty$ represents the increment of the group delay deviation caused by the coefficient adjustment. A large η implies the reduction in the number of FAs is more significant than the degradation in linearity, hence a high efficiency in coefficient modification.

To increase η, the gradient of group delay, $\nabla\tilde{\tau}$ of H_i is calculated according to (3). The coefficients are adjusted based on the calculation of $\nabla\tilde{\tau}$. In practice, only the few least significant digits of each CSD coefficient can be adjusted to minimize the change in group delay deviation. The number of least significant digits m that can be modified is determined by the trust region ρ in the i-th iteration. To ensure that the coefficient adjustment is smaller than ρ,

$$m = round(\log_2 \rho - l_{coef}) \tag{16}$$

```
Coeff_Adjust(H_i, m, n, G) {
    Initialize H_i_adjust; // initialize the solution
    Initialize η_best = 0;   // initialize the best trade-off value
    Initialize csd_best;  //initialize the solution
    csd_H_i = CSD(H_i) ;
    calculate ∇τ̃ ;
    for j from 1 to G
        csd_adjust[j] = Random_Adjust(csd_H_i, m) ;
        calculate η for csd_adjust[j];
        if η > η_best
            η_best = η; csd_best = csd_adjust[j];
        if no η_best updated for n rounds
            break;
    H_i_adjusted = Decimal(csd_best);
    return H_i_adjusted;
}
```

FIGURE 5. Coefficient adjustment for full adder cost minimization.

where l_coef is the word length of the coefficient and $round()$ is the nearest integer operation.

There are many possible ways to adjust a coefficient set constrained by $\nabla\tilde{\tau}$ and m. To maximize η, the adjustment must lead to a reduction in FA cost calculated by (14). All the possible CSs with length shorter than m in unadjusted digits of the coefficients are evaluated. Any adjustment in the last m digits that will introduce more possible CS is counted as a profitable adjustment, and the total number of profitable adjustments G is calculated. For each profitable adjustment, η of the new coefficient set is calculated by (15). The adjusted coefficient set with the highest η is chosen as the final solution. By maximizing η, the proposed coefficient adjustment algorithm reduces the hardware complexity of the filter with a small increase in group delay deviation. This allows more room for the reduction of group delay deviation in the generation of initial IIR filter by decreasing e_{max} in the initialization algorithm proposed in Sections B.1 and B.2. The pseudocode of the proposed coefficient adjustment algorithm is shown in Fig. 5.

The function **Coeff_Adjust** returns the adjusted coefficient H_{i_adjust}. The inputs to **Coeff_Adjust** are the optimized solution H_i in the i-th iterative optimization round, the number of digits to be modified m, the maximum number of rounds with no update n, and the number of selected adjustments G. The gradient of group delay $\nabla\tilde{\tau}$ is calculated after the coefficients of H_i has been converted into CSD representation by the function **CSD**. The function **Adjust**in the for loop adjusts the coefficient csd_H_i and then η of the adjusted coefficient set $csd_adjust$is calculated. If the adjusted coefficient set results in better η, csd_best is updated to the adjusted coefficient set. This process is repeated until no improvement in η can be made. Upon exiting the loop, the coefficient set H_{i_adjust} of the best solution is returned by converting csd_best into decimal representation.

The pseudocode of the complete IIR filter design algorithm **IIR_design** is shown in Fig. 6. It begins with searching for a good initial IIR filter coefficient set H_1 by one of the two methods presented in Section III.B depending on whether there is an imposed group delay requirement. The continuous

IIR_Design(τ, δ_p, δ_s ω_1, ω_2, ω_3, ω_4, ρ_{min}, k_p,k_s , G, n, ∇_{min}) {
  Initialize H_best; // Final solution
  Initialize λ_p, λ_s; // Convergence constants
  Initialize ρ; // Initial trust region
  Initialize m // Coefficient adjustment constants
  if τ is given
    H_1 = **Init_IIR_τ** (τ, δ_p, δ_s ω_1, ω_2, ω_3, ω_4) ;
  else
    H_1 = **Init_IIR**(δ_p, δ_s ω_1, ω_2, ω_3, ω_4) ;
  Calculate $\delta(H_1, e^{j\omega_p})$, $\delta(H_1, e^{j\omega_s})$;
  $\lambda_p = k_p\delta(H_1, e^{j\omega_p})$, $\lambda_s = k_s\delta(H_1, e^{j\omega_s})$;
  while $\rho > \rho_{min}$
    solve convex problem (11);
    if $\dfrac{\left\|dev_\tau(H_i,\omega)\right\|_\infty - \left\|dev_\tau(H_{i+1},\omega)\right\|_\infty}{\left\|\nabla_H dev_\tau(H_i,\omega)^T \Delta H_i\right\|_\infty} < \nabla_{\min}$;
      ρ=0.5ρ;
    else ρ=1.2ρ, $H_{i+1} \leftarrow H_i + \Delta_H$;
      calculate $\delta(H_{i+1}, e^{j\omega_p})$, $\delta(x_{i+1}, e^{j\omega_s})$;
      $\lambda_p = k_p\delta(H_{i+1}, e^{j\omega_p})$, $\lambda_s = k_s\delta(H_{i+1}, e^{j\omega_s})$;
      calculate m;
      H_{i+1_adjust} = **Coeff_Adjust**(H_{i+1},m, n, G) ;
      calculate $\delta(H_{i+1\_adjust}, e^{j\omega_p})$ and $\delta(H_{i+1\_adjust}, e^{j\omega_s})$
      if $\delta(H_{i+1\_adjust}, e^{j\omega_p}) < \delta_p$ and $\delta(H_{i+1\_adjust}, e^{j\omega_s}) < 1-\delta_s$
        add H_{i+1_adjust} into *Solution_list*;
      end if
    end if
  end
  choose the best result, *H_best,* from *Solution_list*;
  return *H_best*;
}

FIGURE 6. Pesudo code of the proposed algorithm.

valued coefficients of H_1 is initialized to finite word length coefficients with an accuracy equivalent to 16 decimal digits, which is more than sufficient to meet the most stringent phase linearity specifications before coefficient adjustment. The ripple of H_1 is calculated, and the convergence constants, λ_p and λ_s, are initialized to solve the iterative optimization problem. The coefficients of the solution H_i in each iteration are adjusted to maximize η and added into the *Solution_list*. The best coefficient *H_best* with the highest η in the *Solution_list*is returned as the final solution.

An overview of the proposed IIR filter design algorithm **IIR_design** is depicted in the flow chart of Fig. 7.

The time complexity of both functions **Init_IIR_**τ and **Init_IIR** increases quadratically with the input filter order N since the model reduction function involves computations with an $N \times N$ matrix that represents the filter system. Other calculations have linear time complexity with N. Hence, the overall time complexity of the proposed algorithm is $O(N^2)$.

E. DESIGN EXAMPLE

The benchmark IIR filter II from [17] is used to demonstrate the design flow of our proposed algorithm.

This is a high pass filter with normalized passband and stopband frequencies at $\omega_1 = 0.4$ and $\omega_2 = 0.6$, respectively. The design specifications call for a passband ripple

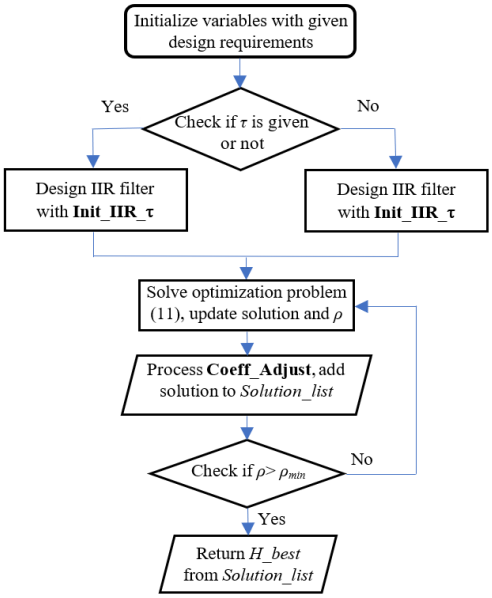

FIGURE 7. Flow chart of the proposed algorithm.

magnitude of 0.1 dB and the minimum stopband attenuation of 60 dB.

The function **IIR_Design** is called with input values, τ, δ_p, $\delta_s\omega_1$, ω_2, ω_3 and ω_4 set according to the design specifications. The algorithm starts after the following constants have been initialized as follows: r_{max} is set to 0.99 for good filter stability; the trust region ρ and the minimum trust region ρ_{min} are set to 0.01 and 0.001, respectively; G and n for coefficient adjustment are set to 10000 and 1000, respectively; k_p and k_s are both set to 10^4; ∇_{min} is set to 0.5 for linear approximation accuracy. Since no group delay is specified, **Init_IIR** is called to obtain an initial filter H_1. The initial 16 bits word length IIR filter solution H_1 has an order of 15, a group delay deviation of 0.00526, and can be implemented with 1342 equivalent FAs. From the filter coefficients of H_1, $\delta(x_1, e^{j\omega_p})$ and $\delta(x_1, e^{j\omega_s})$ are calculated to obtain λ_p and λ_s. By solving the iterative optimization problem, the IIR filter coefficient set H_2 is obtained. Its decimal coefficients, with accuracy up to six significant digits are listed below:

b = {0.00233785, 0.00496586, 0.002735830.000523338,
0.00488778, 0.00812862, −0.00131834,
−0.00696892, 0.00839353, 0.0152878, −0.0133765,
−0.0130133, 0.0611514, −0.0522488, 0.0323094}

a = {1, 3.20042, 6.57387, 10.0924, 12.4517, 12.7864,
11.1193, 8.24336, 5.20658, 2.78068, 1.23591,
0.444131, 0.122931, 0.0238834, 0.00233415}

The above filter coefficients are first quantized to a finite precision of 14 canonical signed digits. The filter after quantization has a response with passband ripple at 0.068 dB and stopband attenuation at 60.9 dB, which fulfills the design specifications. The coefficient set has a group delay

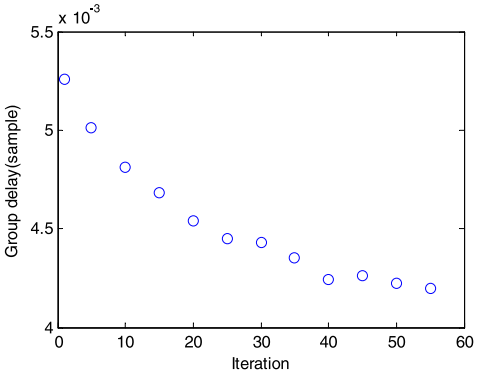

FIGURE 8. Group delay deviation of solutions obtained at different iterations.

deviation of 0.00497 and hardware cost of 1294 equivalent FAs. The improvement in group delay deviation, $\|dev_\tau(H_1+\Delta H_1,\omega)\|_\infty - \|dev_\tau(H_1,\omega)\|_\infty = 0.009$, is larger than $\nabla_{\min}\left\|\nabla_x dev_\tau(H_1,\omega)^T \Delta H_k\right\|_{\inf} = 0.005$. Hence, ρ is relaxed from 0.01 to 0.012. Based on (16) and ρ, the value of m for coefficient adjustment by **Coeff_Adjust** is calculated to be 3. The coefficient set H_2 after adjustment maintains 14 canonical signed digits accuracy and is converted back to decimal digits for the ease of demonstration:

$b = \{0.00233785, 0.00496587, 0.00273582, 0.000523339,$
$0.00488779, 0.00812860, -0.00131829,$
$-0.00696891, 0.00839353, 0.0152882, -0.0133790,$
$-0.0130128, 0.0611511, -0.0522489, 0.0323092\}$

$a = \{1, 3.20038, 6.57387, 10.0924, 12.4517, 12.7865,$
$11.1192, 8.24336, 5.20659, 2.78065, 1.23592,$
$0.444108, 0.122931, 0.0238836, 0.00233402\}$

The filter with above coefficients has passband ripple of 0.069 dB and stopband attenuation of 60.8dB. Although the group delay deviation of H_2 has increased by 6.6% to 0.0053 after **Coeff_Adjust**, the FA cost saving has also been increased from 58 to 96. The solution H_2 in this iteration, along with its complexity-linearity optimization efficiency η of 67857, is added into *Solution_list*.

The same process described above is repeated in subsequent iterations until ρ becomes smaller than ρ_{min}. For this design example, the program terminates at the 55^{th} iteration. The group delay deviations of the output filter obtained from the $1^{\text{st}}, 5^{\text{th}}, 10^{\text{th}} \cdots 55^{\text{th}}$ iterations are plotted in Fig. 8 to show the trend of improvements.

The final solution is chosen from the least cost solution in *Solution_list*. The coefficients have an accuracy of 14 canonical signed digits, with passband ripple 0.063dB and stopband attenuation of 61.90dB. Its decimal coefficient values, with accuracy up to 6 significant figures are listed below.

$b = \{0.00233783, 0.00496587, 0.00273583, 0.000523340,$
$0.00488782, 0.00812861, -0.00131833,$
$-0.00696899, 0.00839354, 0.0152882, -0.0133790,$
$-0.0130128, 0.0611511, -0.0522489, 0.0323092\}$

$a = \{1, 3.20033, 6.57387, 10.0924, 12.4517, 12.7867,$
$11.1194, 8.24336, 5.20661, 2.78062, 1.23592,$
$0.444013, 0.122931, 0.0238848, 0.00233213\}$

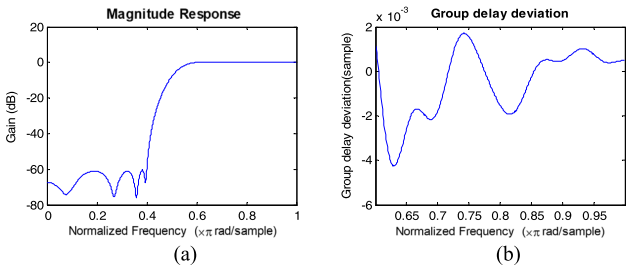

FIGURE 9. (a) Magnitude response of IIR, and (b) Group delay deviation in passband designed by the proposed algorithm.

TABLE 2. Cost saving of the IIR due to CSE for an 8-Bit input signal.

CS	f_{CS} in a	f_{CS} in b	# FAs saved
101	5	4	42
$10\bar{1}$	3	4	30
$\bar{1}01$	8	6	72
$\bar{1}0\bar{1}$	3	5	48
1001	4	2	20
$100\bar{1}$	4	3	25

The magnitude response and group delay deviation of the IIR filter solution are plotted in Fig. 9. It can be shown that the generated IIR filter satisfies the design specifications, and its group delay deviation is only 0.0042.

The frequency of occurrence (f_{CS}) of each CS of the final IIR solution is tabulated in Table 2. Subexpressions $\bar{1}001$ and $\bar{1}00\bar{1}$ are not CSs as they do not appear more than once in the coefficient set a or b. The hardware cost of the final solution of this design example is 1171 equivalent FAs, which is 11.8% lower than that of the initial solution H_1. At the same time, its group delay deviation has also been reduced by 20.2% from H_1.

IV. LOGIC SYNTHESIS RESULT AND DISCUSSION

A. COMPARISON WITH THE LATEST FIR SOLUTION

In this section, an IIR filter designed by the proposed algorithm is compared against the solution provided by the state-of-art FIR filter design algorithm [6] and MFIR filter by [34]. Practical filter 9 from [6] is used as an example. This is a low pass filter with normalized passband and stopband frequencies at 0.042 and 0.14. The specified passband ripple magnitude is 0.2 dB and the minimum stopband attenuation is 60 dB. The magnitude and phase responses of the synthesized IIR filter are shown in Fig. 10(a) and (b), respectively.

The designed IIR filter has negligible group delay deviation of 0.002, so its phase response is nearly-linear, which is showcased in Fig. 10(b). To compare the hardware cost of this design against the minimum cost linear phase FIR filter

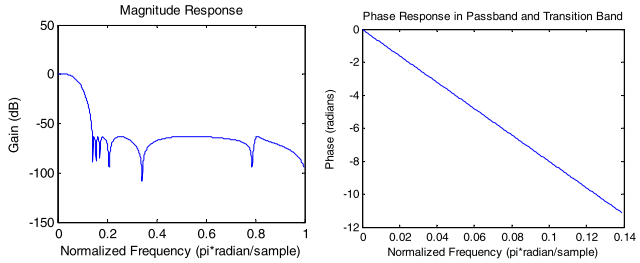

FIGURE 10. (a) Magnitude responses, and (b) phase response in passband and transition band of IIR filter designed by the proposed algorithm.

TABLE 3. Comparison between proposed IIR filter and recent fir filter solutions.

	FIR [6]	MFIR [34]	Proposed IIR
Order	51	57	13
Group delay deviation	0	0	0.0020
Area in #LUTs	2219	2535	1432
Power in mW	0.185	0.201	0.112
Passband Ripple in dB	0.196	0.173	0.192
Stopband Attenuation in dB	61	64	61

solution, both designs are synthesized using the Xilinx ISE Design Suite v14.7 and mapped to the same Xilinx Spartan6, xc6slx75t FPGA device. The power dissipations of the filters in FPGA implementation are analyzed by Xilinx Xpower Analyzer (XPA) at the same clock frequency of 70 MHz and supply voltage of 1.2V. The results are shown in Table. 3. With much lower filter order, the proposed IIR solution requires significantly less memory units and arithmetic operators. The proposed IIR solution has on average reduced the area and the power consumption of the two FIR solutions by 39.4% and 41.8%, respectively, with a negligible group delay deviation of only two thousandths. As the IIR filters designed by the proposed algorithm can achieve almost linear phase response as the FIR filters with relatively lower area and power, it can be concluded that practically most, if not all FIR filters that do not demand a perfect linear phase response can benefit from replacing them by the IIR filters designed by the proposed algorithm.

B. COMPARISON WITH RECENT IIR SOLUTIONS

The quality of the solutions generated by the proposed algorithm is also evaluated against solutions produced by the most recent IIR filter design algorithm [17] for nearly-linear phase IIR filter with minimax phase error and an argument principle based nearly-linear phase IIR filter design algorithm [18]. These two algorithms in comparison are the state-of-the-art methodologies for the design of low-complexity stable IIR filters, with performance generally exceeds previously reported works in [9], [31], and [32]. Eight practical filters from [9] are used as benchmarks, and their filter specifications are listed in Table 4. IIR1, IIR2, IIR5, IIR6 and IIR8 are low pass filters, with different bandedge frequencies

TABLE 4. Specifications of eight benchmark filters.

Filter	Passband and stopband frequency (π×rad/sample)	Passband Ripple (dB)	Stopband attenuation (dB)
IIR1	$\omega_3 = 0.4$; $\omega_4 = 0.6$	0.1	45
IIR2	$\omega_3 = 0.4$; $\omega_4 = 0.56$	0.25	44
IIR3	$\omega_1 = 0.4$; $\omega_2 = 0.6$	0.1	60
IIR4	$\omega_2 = 0.3$; $\omega_3 = 0.5$	1	41
IIR5	$\omega_3 = 0.5$; $\omega_4 = 0.6$	0.266	34
IIR6	$\omega_3 = 0.36$; $\omega_4 = 0.44$	0.2	43
IIR7	$\omega_1 = 0.475$; $\omega_2 = 0.525$	0.720	27
IIR8	$\omega_3 = 0.5$; $\omega_4 = 0.6$	0.42	30

TABLE 5. Comparison of orders and fa costs of iir filters designed by [17], [18] and the proposed algorithm.

Filter	Order			#FAs		
	[17]	[18]	Proposed	[17]	[18]	Proposed
IIR1	11	12	12	1028	1762	749
IIR2	12	14	14	1702	1926	1287
IIR3	15	15	15	1547	1796	1171
IIR4	15	18	17	1918	2541	1290
IIR5	13	14	16	1953	2328	1654
IIR6	17	20	21	3082	3275	2626
IIR7	15	18	18	2023	2350	1897
IIR8	12	12	14	1350	1542	1370

and peak ripple magnitudes. For example, IIR1 has a very small passband ripple with high stopband attenuation, and IIR6 is a low pass filter with narrow transition band. IIR3 and IIR4 are high pass filter and band pass filter, respectively. IIR7 is another high pass filter whose transition band width is $0.05\pi \times$rad/sample. Similar to the proposed algorithm, the filter coefficients in the solutions produced by [17] and [18] are quantized until their filter specifications are barely met. At this point, any further quantization level reduction will cause a violation of one or more design requirements. This is to ensure that the implementation costs for the filters designed by [17] and [18] are also fairly minimized. For the proposed algorithm, the parameters r_{max} is set to 0.99 to ensure stability after coefficient adjustment. ρ and ρ_{min} are set to 0.01 and 0.001, respectively to accelerate convergence. G and n for the coefficient adjustment algorithm are limited to 10000 and 1000, respectively to reduce the search space for lower FA cost solutions. k_p and k_s are both set to 10^4, and ∇_{min} for trust region adjustment is set to 0.5 for fast convergence.

The orders of the filters designed by [17] and [18] and the proposed algorithm are listed in Table 5. The orders of the IIR filters range from 11 to 21, representing median to long filters. Short IIR filters with order lower than 10 have no obvious advantages in hardware complexity when compared with FIR filters of absolute constant phase response, hence they are not considered. The average FA costs of the IIR filters designed by the proposed algorithm are 27.4% and 40.9% lower than those of [17] and [18], respectively. This is due to the effectiveness of our proposed coefficient adjustment in harnessing more CSs, which leads to the significantly reduced number of FAs.

TABLE 6. Sensitivity analysis and quantization errors of IIR filters designed by [17], [18] and the proposed algorithm.

(A)

Filter	Passband Ripple (dB)				Stopband Attenuation (dB)			
	before	after	Δ	Required	before	after	Δ	Required
IIR1	0.0245	0.0832	0.0587	0.1	48.18	47.37	0.81	45
IIR2	0.1198	0.1498	0.03	0.25	48.87	48.17	0.7	44
IIR3	0.0470	0.0751	0.0281	0.1	74.78	61.84	12.94	60
IIR4	0.1799	0.3272	0.1473	1	48.87	43.34	5.53	41
IIR5	0.1427	0.2138	0.0711	0.266	34.69	34.59	0.10	34
IIR6	0.0744	0.0998	0.0254	0.2	48.16	46.04	2.12	43
IIR7	0.2687	0.2696	0.0009	0.720	30.83	29.26	1.57	27
IIR8	0.2134	0.2171	0.0037	0.42	30.78	31.04	-0.26	30

(B)

Filter	Passband Ripple (dB)				Stopband Attenuation (dB)			
	before	after	Δ	Required	before	after	Δ	Required
IIR1	0.0826	0.0833	0.0007	0.1	47.08	46.96	0.12	45
IIR2	0.1625	0.1735	0.0110	0.25	44.65	44.51	0.14	44
IIR3	0.0826	0.0855	0.0029	0.1	61.00	60.82	0.18	60
IIR4	0.5625	0.6125	0.0500	1	42.08	41.83	0.25	41
IIR5	0.1988	0.2012	0.0024	0.266	37.31	37.02	0.29	34
IIR6	0.1056	0.1152	0.0096	0.2	44.23	43.51	0.72	43
IIR7	0.4854	0.5025	0.0171	0.720	28.26	28.14	0.12	27
IIR8	0.3305	0.3628	0.0323	0.42	31.25	30.96	0.29	30

(C)

Filter	Passband Ripple (dB)				Stopband Attenuation (dB)			
	before	after	Δ	Required	before	after	Δ	Required
IIR1	0.0982	0.0996	0.0014	0.1	46.30	46.24	0.06	45
IIR2	0.1790	0.1828	0.0038	0.25	44.73	44.65	0.08	44
IIR3	0.0566	0.0632	0.0066	0.1	61.90	61.32	0.58	60
IIR4	0.4996	0.5208	0.0212	1	44.93	44.62	0.31	41
IIR5	0.1799	0.1922	0.0123	0.266	35.05	34.21	0.84	34
IIR6	0.1084	0.1223	0.0139	0.2	43.22	43.01	0.21	43
IIR7	0.3718	0.3745	0.0027	0.720	27.11	27.02	0.09	27
IIR8	0.2476	0.2518	0.0042	0.42	30.69	30.41	0.28	30

The sensitivity analysis and quantization errors of the eight benchmark filters designed by [17] and [18] and the proposed algorithm are listed Table 6(A), (B) and (C), respectively. The columns labeled Δ represent the quantization errors. It can be concluded that all the required filter specifications are fulfilled after the quantization of continuous filter coefficients.

The magnitude responses and the group delay deviations of the filters designed by the proposed algorithm are shown in Fig. 11, which show that they have all met the design specifications. For IIR7 that has a very narrow transition band width specification, the proposed algorithm can still converge at the 37^{th} iteration out of the total of 39 iterations with a design fulfilling the stringent requirements of transition band width, passband ripple and stopband attenuation.

The maximum group delay deviation of all the designs produced by the three algorithms are computed and listed in Table 7. From Table 7, it can be concluded that IIR filters designed by the proposed algorithm generally have better linearity than those designed by [17] and [18]. For some benchmarks, such as IIR1 designed by [17] and IIR4 designed by [17] and [18], the group delay deviations are slightly lower than our designs. As shown in the later part of this section, IIR1 and IIR4 designed by the proposed algorithm however have significantly lower hardware complexity than

TABLE 7. Maximum group delay deviation (in samples) of IIR filters designed by [17], [18] and the proposed method.

Filter	[17]	[18]	Proposed
IIR1	0.0056	0.0106	0.0062
IIR2	0.0141	0.0359	0.0127
IIR3	0.0193	0.0284	0.0042
IIR4	0.1479	0.1316	0.1595
IIR5	0.0160	0.0243	0.0148
IIR6	0.0221	0.0665	0.0198
IIR7	0.1179	0.2335	0.1094
IIR8	0.015	0.0532	0.0258

TABLE 8. Synthesized FPGA areas in #LUTs and delays in *ns* for IIR filters designed by [17], [18] and the proposed algorithm.

Filter	Area in #LUTs			Delay in *ns*		
	[17]	[18]	Proposed	[17]	[18]	Proposed
IIR1	1020	2522	708	5.366	10.28	6.183
IIR2	2314	2475	1831	8.505	10.70	7.598
IIR3	2358	2697	1755	7.186	10.42	6.755
IIR4	2435	4122	1727	7.861	13.58	7.363
IIR5	3230	3744	2790	10.902	11.63	10.832
IIR6	4784	5175	4345	11.206	12.53	11.983
IIR7	3077	3358	2698	11.601	12.35	10.870
IIR8	1997	2620	1773	13.917	12.26	11.731

those designed by [17] and [18]. Overall, the average group delay deviation of the benchmark filters designed by the proposed algorithm is lower than [17] and [18] by 2.9% and 48.1%, respectively. For the ease of comparison, the average normalized group delay deviations are plotted in Fig. 12, where the group delay deviation for each benchmark filter designed by [17] and the proposed algorithm is normalized by the group delay deviation of the filter designed by [18].

To compare the hardware costs, all the designs are synthesized using Xilinx ISE Design Suite v14.7 and mapped to the same Xilinx Spartan6, xc6slx75t FPGA device. The synthesized areas in terms of the number of LUTs and delays in *n*s are shown in Table 8. The FPGA area of each benchmark filter designed by [17] and the proposed algorithm is normalized by the FPGA areas of the corresponding filter solution designed by [18]. The average normalized area of the filters designed by each method is plotted in Fig. 13. The area of the IIR filters designed by the proposed algorithm are on average 19.1% and 35.6% lower than those designed by [17] and [18], respectively. The area reduction results show that the savings in FAs presented in Table 5 have a good correlation with the relative physical implementation costs, with the exception that when the FA cost difference is small, it may be overwhelmed by additional contrary logic optimizations performed by the synthesis tool. The average delays of the IIR filters designed by all the algorithms in comparison are similar. This is in part due to the limited slack in the critical path of the high-speed transposed direct form II architecture of IIR filters in FPGA implementation and in part due to the total FA cost reduction is not equally distributed across all timing paths. Therefore, the savings in FA cost will not be translated into delay improvement as long as the saving

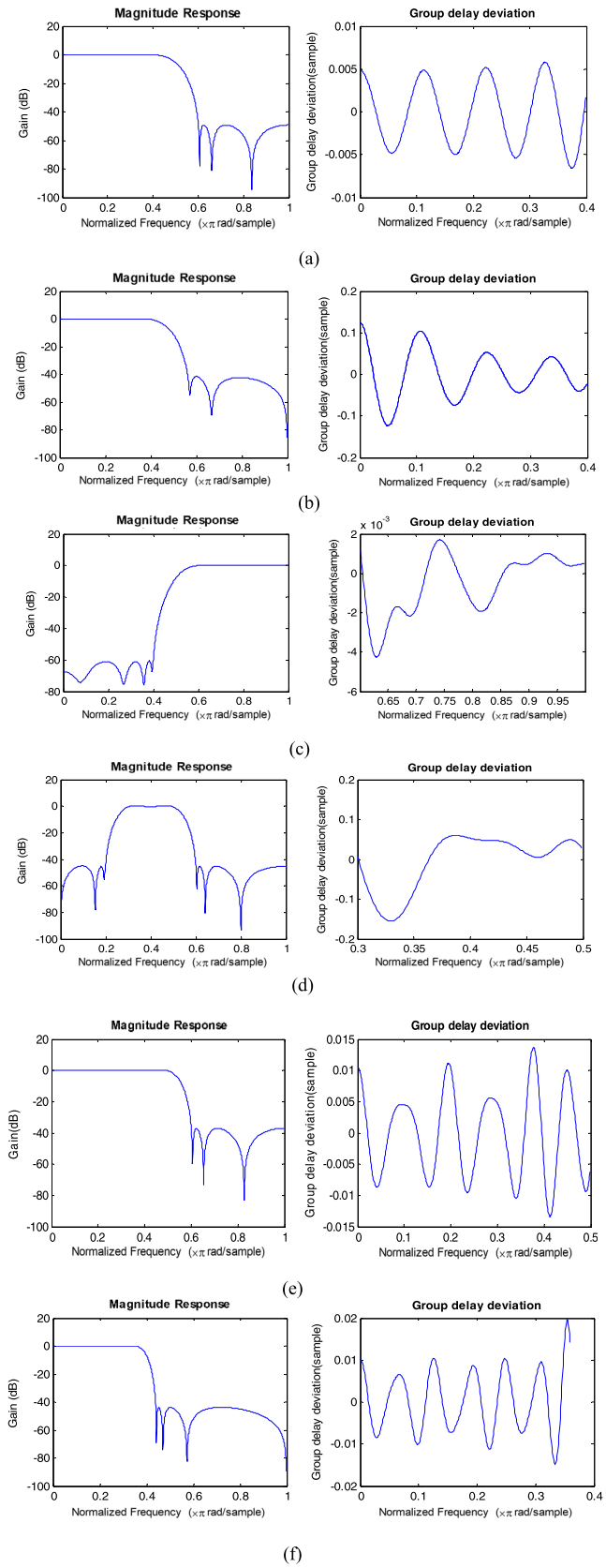

FIGURE 11. Magnitude responses and group-delay deviations in passband of benchmark filters (a) IIR1, (b) IIR2, (c) IIR3, (d) IIR, (e) IIR5, (f) IIR6, (g) IIR7, and (h) IIR8 obtained by the proposed algorithm.

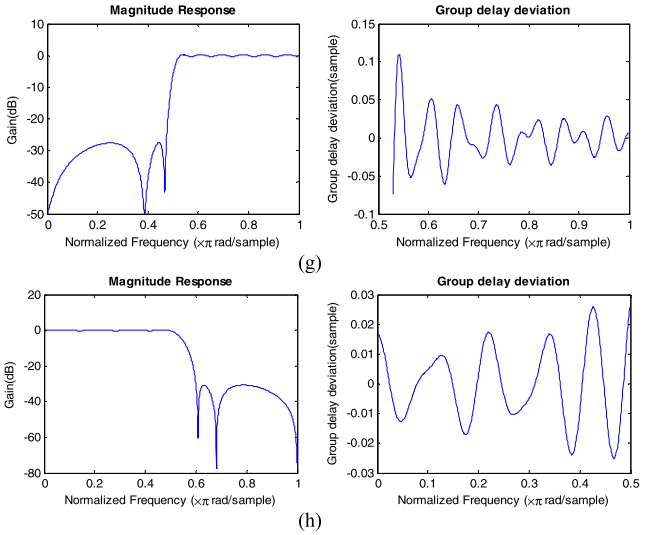

FIGURE 11. ***(Continued.)*** **Magnitude responses and group-delay deviations in passband of benchmark filters (a) IIR1, (b) IIR2, (c) IIR3, (d) IIR, (e) IIR5, (f) IIR6, (g) IIR7, and (h) IIR8 obtained by the proposed algorithm.**

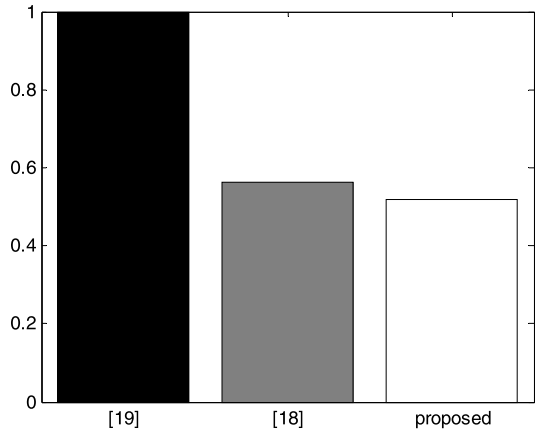

FIGURE 12. Average normalized group delay deviation of the filters designed by [17], [18] and the proposed algorithm.

in any of the critical paths is not significant enough to offset its timing increase due to FPGA routing and configurable logic block mapping constraints.

All the designs are also mapped to STM 65nm standard cell library and synthesized through Synopsys DesignCompiler^{TM}. The synthesized areas in μm^2 and delays in ns of the designs generated by the three different algorithms in comparison are shown in Table 9. Similarly, the ASIC areas of every filter are also normalized by the corresponding filter solution designed by [18]. The average normalized areas of the filters designed by the three algorithms are also plotted in Fig. 13. On average, the ASIC areas of the IIR filters designed by the proposed algorithm are 10.8% and 30.1% smaller than those designed by [17] and [18], respectively. This result is consistent with the area savings on FPGA. The average delay of the six IIR filter circuits designed

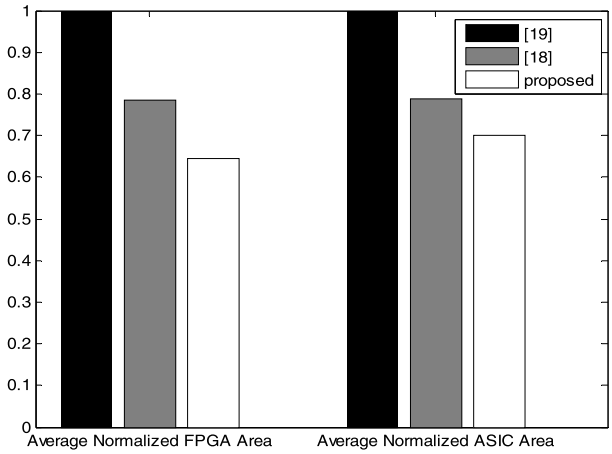

FIGURE 13. Average normalized FPGA area and ASIC area of the filters designed by [17], [18] and the proposed algorithm.

TABLE 9. Synthesized ASIC areas in μM^2 and delays in *ns* for IIR filters designed by [17], [18] and the Proposed algorithm.

	Area in μm^2			Delays in *ns*		
Filter	[17]	[18]	Proposed	[17]	[18]	Proposed
IIR1	21092	59173	20171	1.376	2.2329	1.316
IIR2	59602	64681	51304	1.767	2.0084	1.513
IIR3	62032	64645	51135	2.223	2.327	1.669
IIR4	65951	96681	50145	2.24	2.599	1.602
IIR5	82353	88356	79895	2.203	2.211	2.179
IIR6	113595	135026	100848	2.555	2.2987	2.721
IIR7	85259	97474	79188	1.958	2.010	1.989
IIR8	55333	76052	52038	1.713	2.235	1.632

by the proposed algorithm is 2.9% and 22.1% shorter than those designed by [17] and [18], respectively. Unlike FPGA, there is more freedom for optimization in logic synthesis and technology mapping in ASIC. Hence, the critical path delays of the proposed designs can still be shortened noticeably due to their logic reduction.

The power dissipations of the IIR filters in FPGA implementation are analyzed by Xilinx Xpower Analyzer (XPA). For a fair comparison, all the designs in comparison operate at the same clock frequency of 100MHz and supply voltage of 1.2V. The simulated power dissipation results in *m* W are listed in Table 10. The power consumptions of all the designs mapped to the standard cell library for ASIC implementation are also simulated by Synopsys Prime Time PX version: Z-2006.12. The supply voltage and clock frequency are set to 0.9V and 250 MHz, respectively for all the designs in comparison. As switching power is input dependent, Monte Carlo power simulation [30] was adopted by applying randomly generated input vectors epoch by epoch until the maximum error in mean power dissipation within the 95% confident interval converges to 5% or lower. This goal was reached with slightly more than 360 test vectors. Eventually, 400 random input vectors were applied, which worked out to a statistical error bound of 4% within 95% confidence interval. The ASIC power results are also presented in Table 10.

TABLE 10. Total power consumptions in *m* W for FPGA and ASIC implementations of IIR filter designed by [17], [18] and the proposed algorithm.

Filter	FPGA			ASIC		
	[17]	[18]	Proposed	[17]	[18]	Proposed
IIR1	0.032	0.105	0.027	0.014	0.036	0.017
IIR2	0.052	0.122	0.047	0.035	0.040	0.030
IIR3	0.096	0.105	0.060	0.025	0.038	0.025
IIR4	0.079	0.131	0.051	0.039	0.051	0.034
IIR5	0.085	0.122	0.047	0.043	0.049	0.037
IIR6	0.126	0.114	0.129	0.045	0.063	0.046
IIR7	0.119	0.125	0.117	0.038	0.053	0.036
IIR8	0.094	0.133	0.053	0.032	0.043	0.029

The results show that the power savings of our proposed algorithm is more significant in FPGA than in ASIC platform. On average, it reduces the FPGA power consumptions of the IIR filters designed by [17] and [18] by 23.2% and 44.3%, respectively. On the other hand, the average savings in power consumption for the ASIC implementation of IIR filters designed by the proposed algorithm over those designed by [17] and [18] are 4.0% and 32.7%, respectively. The results are indicative of the correlation between switching power and logic complexity, particular for FPGA implementation as FPGA fabrics and routings have inherently higher power dissipation than similar netlist in ASIC.

V. CONCLUSION

A new methodology for designing nearly-linear phase IIR filter with low hardware implementation cost without unduly sacrificing linearity has been presented. An iterative convex optimization problem is formulated to reduce the search complexity for minimization of group delay deviation and hardware complexity. A low order IIR filter is determined as a good initial solution to evade poor local minima in iterative convex optimization. The coefficients of the candidate IIR solution in each iteration are modified to maximize the sharing of common subexpressions for hardware cost reduction. In FPGA implementation, the nearly-linear phase IIR filter designed by the proposed algorithm saves 39.4% and 41.8% in area and power, respectively over the two most area-power FIR solutions produced by the latest FIR filter design methods. The synthesis results using 65*n*m standard cell library show that the IIR filters designed by our proposed algorithm reduces on average the group delay deviation by 25.5%, the silicon area by 20.5% and the power consumption by 18.4% comparing with the solutions generated by two recently proposed IIR filter design algorithms. More significant savings in hardware resources and power consumption are observed for their corresponding FPGA implementation.

REFERENCES

[1] R. Jia, H.-G. Yang, C. Y. Lin, R. Chen, X.-G. Wang, and Z.-H. Guo, "A computationally efficient reconfigurable FIR filter architecture based on coefficient occurrence probability," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 8, pp. 1297–1308, Aug. 2016.

[2] F. Feng, J. Chen, and C.-H. Chang, "Hypergraph based minimum arborescence algorithm for the optimization and reoptimization of multiple constant multiplications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 2, pp. 233–244, Feb. 2016.
[3] J. Ding, J. Chen, and C.-H. Chang, "A new paradigm of common subexpression elimination by unification of addition and subtraction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 10, pp. 1605–1617, Oct. 2016.
[4] Y. Li *et al.*, "A novel fully synthesizable all-digital RF transmitter for IoT applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 1, pp. 146–158, Jan. 2018.
[5] R. H. Tehrani, S. Vahid, D. Triantafyllopoulou, H. Lee, and K. Moessner, "Licensed spectrum sharing schemes for mobile operators: A survey and outlook," *IEEE Commun. Surv. Tut.*, vol. 18, no. 4, pp. 2591–2623, 4th Quart., 2016.
[6] J. Chen, C.-H. Chang, J. Ding, R. Qiao, and M. Faust, "Tap delay-and-accumulate cost aware coefficient synthesis algorithm for the design of area-power efficient FIR filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 712–722, Feb. 2018.
[7] C. Charalambous and A. Antoniou, "Equalisation of recursive digital fitters," *IEE Proc. G-Electron. Circuits Syst.*, vol. 127, no. 5, pp. 219–225, Oct. 1980.
[8] V. Sreeram and P. Agathoklis, "Design of linear-phase IIR filters via impulse-response gramians," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 389–394, Feb. 1992.
[9] R. C. Nongpiur, D. J. Shpak, and A. Antoniou, "Improved design method for nearly linear-phase IIR filters using constrained optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 895–906, Feb. 2013.
[10] A. Deczky, "Synthesis of recursive digital filters using the minimum p-error criterion," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, no. 4, pp. 257–263, Oct. 1972.
[11] M. C. Lang, "Least-squares design of IIR filters with prescribed magnitude and phase responses and a pole radius constraint," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3109–3121, Nov. 2000.
[12] R. C. Nongpiur, D. J. Shpak, and A. Antoniou, "Design of IIR digital differentiators using constrained optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1729–1739, Apr. 2014.
[13] K. Steiglitz and L. McBride, "A technique for the identification of linear systems," *IEEE Trans. Autom. Control*, vol. 10, no. 4, pp. 461–464, Oct. 1965.
[14] W.-S. Lu, "An argument-principle based stability criterion and application to the design of IIR digital filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, Island Kos, Greece, May 2006, p. 4.
[15] A. Jiang and H. K. Kwan, "IIR digital filter design with novel stability criterion based on argument principle," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 2339–02342.
[16] H. Silverman, *Complex Variable*, Boston, MA, USA: Houghton Mifflin, 1975.
[17] X. Lai and Z. Lin, "Iterative reweighted minimax phase error designs of IIR digital filters with nearly linear phases," *IEEE Trans. Signal Process.*, vol. 64, no. 9, pp. 2416–2428, May 2016.
[18] A. Jiang and H. K. Kwan, "IIR digital filter design with new stability constraint based on argument principle," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 3, pp. 583–593, Mar. 2009.
[19] X. Lai and Z. Lin, "Design and application of allpass filters with equiripple group delay errors," in *Proc. IEEE Int. Symp. Circuits Syst.*, Beijing, China, May 2013, pp. 2924–2927.
[20] M. Potkonjak, M. B. Srivastava, and A. P. Chandrakasan, "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, no. 2, pp. 151–165, Feb. 1996.
[21] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," *IEEE Trans. Circuits Syst. II. Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 689–702, Oct. 1996.
[22] A. Jiang, "IIR digital filter design using convex optimization," Ph.D. dissertations, Dept. Elect. Comput. Eng., Univ. Windsor, Windsor, ON, Canada, 2009, p. 432. [Online]. Available: https://scholar.uwindsor.ca/etd/432
[23] J. Tan and J. Chen, "Low complexity and quasi-linear phase IIR filters design based on iterative convex optimization," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, Jeju, South Korea, Oct. 2016, pp. 603–606.
[24] W.-S. Lu, "Design of stable IIR digital filters with equiripple passbands and peak-constrained least-squares stopbands," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 11, pp. 1421–1426, Nov. 1999.
[25] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L, 8 -error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, Jan. 1984.
[26] P. Wortelboer and H. van Oostveen, "Modal reduction guided by Hankel singular value intervals (with application to control design for compact disc players)," in *Proc. IEE Colloq. Integrating Control Syst. Design Anal. Flexible Struct.*, London, U.K., Feb. 1990, pp. 2/1–2/3.
[27] T. Parks and J. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, no. 2, pp. 189–194, Mar. 1972.
[28] R. M. Hewlitt and E. S. Swartzlantler, "Canonical signed digit representation for FIR digital filters," in *Proc. IEEE Workshop SiGNAL Process. Syst. SiPS Design Implement.*, Lafayette, LA, USA, Oct. 2000, pp. 416–426.
[29] K. G. Smitha and A. P. Vinod, "A new binary common subexpression elimination method for implementing low complexity FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, USA, May 2007, pp. 2327–2330.
[30] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 1, pp. 63–71, Mar. 1993.
[31] W.-S. Lu, S.-C. Pei, and C.-C. Tseng, "A weighted least-squares method for the design of stable 1-D and 2-D IIR digital filters," *IEEE Trans. Signal Process.*, vol. 46, no. 1, pp. 1–10, Jan. 1998.
[32] C.-C. Tseng, "Design of stable IIR digital filter based on least p-power error criterion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1879–1888, Sep. 2004.
[33] A. Fam, "A multiplicative realization of FIR systems that is logarithmically efficient," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Denver, CO, USA, Apr. 1981, pp. 268–270.
[34] J.-J. Vandenbussche, P. Lee, and J. Peuteman, "Multiplicative finite impulse response filters: Implementations and applications using field programmable gate arrays," *IET Signal Process.*, vol. 9, no. 5, pp. 449–456, Jul. 2015.

GELEI DENG is currently pursuing the B.Eng. degree in electrical engineering with the Singapore University of Technology and Design. He is also working as a Research Assistant with the SUTD-MIT International Design Center. His main research interests are the digital filter design and optimizations.

JIAJIA CHEN received the B.Eng. (Hons.) and Ph.D. degrees from Nanyang Technological University, Singapore, in 2004 and 2010, respectively. From 2012 to 2018, he was a Faculty Member of the Singapore University of Technology and Design, Singapore. Since 2018, he has been with the Nanjing University of Aeronautics and Astronautics, China, where he is currently a Professor. His research interests include computational transformations of low-complexity digital filters, and low power VLSI designs for digital signal processing applications. He has served as the Web Chair of the Asia-Pacific Computer Systems Architecture Conference 2005; the Technical Program Committee member of the European Signal Processing Conference 2014; and an Associate Editor for the *EURASIP Journal on Embedded Systems* (Springer), since 2016.

JIAXUAN ZHANG is currently pursuing the B.Eng. degree in computer science with the Singapore University of Technology and Design. He is working as Student Research Assistant with the SUTD-MIT International Design Center. His research interests include system optimization and data analysis.

CHIP-HONG CHANG (S'92–M'98–SM'03–F'18) received the B.Eng. degree (Hons.) from the National University of Singapore, in 1989, and the M.Eng. and Ph.D. degrees from Nanyang Technological University (NTU), in 1993 and 1998, respectively. He has served as a Technical Consultant in industry prior to joining the School of Electrical and Electronic Engineering (EEE), NTU, in 1999, where he is currently an Associate Professor. He holds joint appointments with the university as the Assistant Chair of Alumni of the School of EEE, from 2008 to 2014; the Deputy Director of the Center for High Performance Embedded Systems, from 2000 to 2011; and the Program Director of the Center for Integrated Circuits and Systems, from 2003 to 2009. He has edited four books; has published ten book chapters, around 100 international journal papers (two-thirds are IEEE), and more than 160 refereed international conference papers (mostly in IEEE); and has delivered over 30 colloquia. His current research interests include hardware security and trustable computing, low-power and fault-tolerant computing, residue number systems, and application-specific digital signal processing.

Dr. Chang has served in the organizing and technical program committee for over 60 international conferences (mostly IEEE). He is a Fellow of the IET. He serves as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, since 2011; the IEEE ACCESS, since 2013; the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and the IEEE TRANSACTIONS ON INFORMATION FORENSIC AND SECURITY, since 2016; the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I, from 2010 to 2013; *Integration*, the VLSI Journal, from 2013 to 2015; the *Journal of Hardware and System Security* (Springer), since 2016; and the *Microelectronics Journal*, since 2014. He was the editorial advisory board member of the *Open Electrical and Electronic Engineering Journal*, from 2007 to 2013, and the *Journal of Electrical and Computer Engineering*, from 2008 to 2014, as well as the Guest Editor of several special issues.