<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Area- and power-efficient nearly-linear phase response IIR filter by iterative convex optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Deng, Gelei; Chen, Jiàjìa; Zhang, Jìaxuàn; Chang, Chip-Hong</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2019</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/48873">http://hdl.handle.net/10220/48873</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2019 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See <a href="http://www.ieee.org/publications_standards/publications/rights/index.html">http://www.ieee.org/publications_standards/publications/rights/index.html</a> for more information.</td>
</tr>
</tbody>
</table>
Area- and Power-Efficient Nearly-Linear Phase Response IIR Filter by Iterative Convex Optimization

GELEI DENG, JIAJIA CHEN, JIAXUAN ZHANG, AND CHIP-HONG CHANG, (Fellow, IEEE)

1Department of Engineering Product Development, Singapore University of Technology and Design, Singapore 487372
2College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
3School 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 resurfaced 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-
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 FORMULATION

A. IIR FILTER PRELIMINARIES

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

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

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

The ripple magnitude \( \delta \) 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:

\[ \delta(e^{j\omega}) = \| H(e^{j\omega}) - \tilde{H}(e^{j\omega}) \|_\infty \]  

where \( \| \alpha \|_\infty \) denotes the infinite norm of \( \alpha \). 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

\[ \tilde{\tau}(\omega) = -\frac{d}{d\omega} \angle H(e^{j\omega}) \]  

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 \( \tau \), i.e.,

\[ dev_\tau(\omega) = \tilde{\tau}(\omega) - \tau \]  

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 \( r_{\text{max}} \) 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 \( r_{\text{max}} \) 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

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

where arg is the argument operator and \( \oint_c d \) denotes the contour integral along the circle \( c \) of radius \( r_{\text{max}} \) in counterclockwise 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 \( A_0, A_1, \ldots, A_{M-1} \) 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)
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 \( \omega_1, \omega_2, \omega_3 \) and \( \omega_M \), as shown in Table 1.

Let \( \delta_p \) and \( \delta_s \) denote the maximum allowable passband ripple and the minimum stopband attenuation, respectively. The ripple and attenuation constraints can be specified as

\[
\begin{align*}
\delta(e^{j\omega}) & \leq \delta_p \quad \text{for } \omega \in \text{passband} \\
\delta(e^{j\omega}) & \leq 1 - \delta_s \quad \text{for } \omega \in \text{passband} \\
|H(e^{j\omega})| & \leq 1 \quad \text{for } \omega \in \text{transition band}
\end{align*}
\]

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

\[
|H(e^{j\omega})| \leq 1 \quad \text{for } \omega \in \text{transition band}
\]

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

\[
\text{Minimize } dev_t(\omega), C_{FA}
\]

\[
\text{s.t. } \delta(e^{j\omega}) \leq \delta_p, \quad \delta(e^{j\omega}) \leq 1 - \delta_s, \quad |H(e^{j\omega})| \leq 1, \quad \int_{c} d[\arg A(z)] = 0
\]

where \( \omega_p, \omega_s, \omega_t \) 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

\[
H_i = \begin{bmatrix} b_0^{(i)} & b_1^{(i)} & \cdots & b_M^{(i)} \\ a_0^{(i)} & a_1^{(i)} & \cdots & a_N^{(i)} \end{bmatrix}^T,
\]

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:

\[
\text{minimize } \| dv_t(H_i, \omega) + \nabla H dv_t(H_i, \omega)^T \Delta H_i \|_{\infty}
\]

where \( \nabla \) is the gradient operator and \( dv_t(H_i, \omega) \) 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:

$$
\text{Minimize: } \left\| \text{dev}_t(H_1, \omega) + \nabla_H \text{dev}_t(H_1, \omega)^T \Delta H_1 \right\|_\infty + \lambda_p \left\| \delta(e^{|\omega|}) + \nabla_H \delta(e^{|\omega|})^T \Delta H_1 \right\|_\infty + \lambda_s \left\| \delta(e^{|s|}) + \nabla_H \delta(e^{|s|})^T \Delta H_1 \right\|_\infty
$$

Subject to:

$$
\delta(e^{|\omega|}) + \nabla_H \delta(e^{|\omega|})^T \Delta H_1 \leq \delta_p
$$

$$
\delta(e^{|s|}) + \nabla_H \delta(e^{|s|})^T \Delta H_1 \leq 1 - \delta_s
$$

$$
\frac{1}{\pi} \int_c d \arg A(z) = 0; \quad \Delta H_1 \leq \rho
$$

where $\lambda_p$ and $\lambda_s$ are convergence constants for the passband and stopband regions, respectively, and $\rho$ 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 $\epsilon_{\text{max}}$, 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 $\epsilon_{\text{max}}$ 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 $C_F$ is sought from the IIR coefficient sets bounded by the error between 0 and $\epsilon_{\text{max}}$. The pseudo code of the search for a good initial solution is shown in Fig. 2.

The function $\text{Init}_{IIR\_\tau}$ generates an initial solution $H_1$ for the convex optimization problem with a specified group delay $\tau$, 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 $2 \tau$ for a group delay of $\tau$. 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 $\text{FIR\_syn}$ uses the Parks McClellan algorithm to synthesize an FIR filter that fulfills the frequency response specifications. The function $\text{order\_estimate}$ uses Hankel-norm approximation [26] to calculate the minimum order $N_{\text{min}}$ of IIR filter that can approximate the FIR filter with an absolute error bounded by $\epsilon_{\text{max}}$. The filter order $N_{IIR}$ is incremented from $N_{\text{min}}$ to $N_{FIR}$. In each iteration, a candidate IIR filter of order $N_{IIR}$ is generated from an FIR filter by the function $\text{model\_reduction}$, and the function $\text{get\_coeff}$ extracts the coefficients of the candidate IIR filter from its corresponding state-space representation. The FA cost for the coefficient set is evaluated by $\text{FA\_count}$. At the end of the iterations, the IIR filter $H_1$ with the least FA cost and an absolute error lower than $\epsilon_{\text{max}}$ 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 $2 \tau$, 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
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 fil-  
ter is synthesized and the minimal order IIR filter is generated  
to approximate this filter with the absolute error bounded by  
$e_{\text{max}}$. 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 $\text{PM}$, which calculates the lowest FIR filter order  
$N_{\text{lowest}}$ 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, $\lambda_0$  
and $\lambda_1$ in (11) are set to be $k_p \times \delta(H_1, e^{\text{top}})$ and  
k_s \times \delta(H_1, e^{\text{low}})$, respectively, where $\delta(H_1, e^{\text{top}})$  
is the passband ripple and $\delta(H_1, e^{\text{low}})$ 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 $\rho_{\text{initial}}$.

With these parameters specified, the optimization problem  
(11) is solved iteratively. Each iteration generates a new  
solution $H_i$ with a difference $\Delta H_i$ between the new solution  
$H_i$ and its previous solution $H_{i-1}$. $\lambda_0$ and $\lambda_1$ 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 $\rho$ can be adjusted  
after the second iteration, i.e., $i = 2$, to reduce the linear  
approximation error. After obtaining $\Delta H$ from the previ- 
uous two iterations, the actual improvement on group delay  
deviation, $\|\text{dev}_t(H_i + \Delta H_i, \omega)\|_{\infty} - \|\text{dev}_t(H_i, \omega)\|_{\infty}$, is calcu- 
lated and compared with the linear approximation group  
delay deviation improvement $\|\text{dev}_t(H_i, \omega)^T \Delta H_i\|_{\infty}$. The actual percentage improvement should be larger than the  
linear approximation percentage improvement by a fraction  
of at least $\nabla_{\min}$. Otherwise, the linear approximation is not  
accurate and the trust region $\rho$ is reduced from $\rho_{\text{initial}}$  
to reduce the approximation error until the error becomes less  
than $1 - \nabla_{\min}$. If the error is already less than $1 - \nabla_{\min}$, $\rho$ is  
enlarged to expand the search region to accelerate conver- 
gence. $\rho$ 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 $\rho_{\text{th}}$. When $\rho$ is less than $\rho_{\text{th}}$, any further modifi- 
cation with trust region larger than $\rho_{\text{th}}$ will not discernibly  
decrease the group delay deviation.

D. REDUCTION OF HARDWARE COMPLEXITY  

To maximize the number of adders shared in the imple- 
mentation 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  
blocks 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 specifi- 
cations are fulfilled by the resulting filter. The quantized  
filter coefficients are then transformed into CSD represen- 
tation [28] to facilitate detection and elimination of common  
subexpressions (CSs), and the frequencies of occurrence  
of eight short horizontal CSs of the forms, 101, 101, 101, 101,  
1001, 1001, 1001 and 1001, 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:

$$S = \sum_{j=1}^{n} (f_j - 1)FA_j$$  
(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.
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 digit of a CS. Since the cost of a HA is approximately half the full-adder cost of generating a given CS can be calculated by:

$$FA_{cs} = (l_x - 1)(l_z - z_{cs} - 1) - 0.5z_{cs}$$  \hspace{1cm} (14)

where $l_x$ is the word length of input signal, $l_z$ 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 $\eta$ of FA cost reduction to group delay deviation increment defined in (15).

$$\eta = \frac{S_{pos} - S_{pre}}{\Delta \|dev_f(\omega)\|_{\infty}}$$  \hspace{1cm} (15)

where $S_{pre}$ and $S_{pos}$ are the number of FAs reduced by CS sharing before and after the adjustment of coefficients, respectively. $\Delta \|dev_f(\omega)\|_{\infty}$ represents the increment of the group delay deviation caused by the coefficient adjustment. A large $\eta$ 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 $\eta$, the gradient of group delay, $\nabla \tau$ of $H_i$ is calculated according to (3). The coefficients are adjusted based on the calculation of $\nabla \tau$. 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 $\rho$ in the $i$-th iteration. To ensure that the coefficient adjustment is smaller than $\rho$,

$$m = round(\log_2 \rho - l_{coef})$$  \hspace{1cm} (16)

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 $\nabla \tau$ and $m$. To maximize $\eta$, 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, $\eta$ of the new coefficient set is calculated by (15). The adjusted coefficient set with the highest $\eta$ is chosen as the final solution. By maximizing $\eta$, 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 $\epsilon_{max}$ 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 $H_{\text{adjust}}$. 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 $\nabla \tau$ is calculated after the coefficients of $H_i$ have been converted into CSD representation by the function CSD. The function Adjust in the for loop adjusts the coefficient $csd_{H_i}$ and then $\eta$ of the adjusted coefficient set $csd_{adjust}$ is calculated. If the adjusted coefficient set results in a better $\eta$, $csd_{best}$ is updated to the adjusted coefficient set. This process is repeated until no improvement in $\eta$ can be made. Upon exiting the loop, the coefficient set $H_{\text{adjust}}$ of the best solution is returned by converting $csd_{adjust}$ 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(\tau, \delta_p, \delta_s, \omega_1, \omega_2, \omega_3, \omega_4, \delta_m, k_p, k_s, G, n, V_{\min}) \{
  Initialize H_{\text{best}}; // Final solution
  Initialize \lambda_p, \lambda_s; // Convergence constants
  Initialize \rho; // Initial trust region
  Initialize m; // Coefficient adjustment constants
  if \tau is given
    H_1 = \text{Init\_IIR}(\tau, \delta_p, \delta_s, \omega_1, \omega_2, \omega_3, \omega_4);
  else
    H_1 = \text{Init\_IIR}(\delta_p, \delta_s, \omega_1, \omega_2, \omega_3, \omega_4);
  Calculate \delta(H_1, e^{j\omega}) , \delta(H_1, e^{j\beta});
  \lambda_p = k_p \delta(H_1, e^{j\omega}) , \lambda_s = k_s \delta(H_1, e^{j\beta});
  while \rho > \rho_{\min}
    solve convex problem (11);
    if \[ \frac{\delta_x(H_1, e^{j\omega})}{\delta_x(H_1, e^{j\beta})} < \nu_{\min} ; \]
      \rho = 0.5 \rho;
    else \rho = 1.2 \rho, H_{\text{curr}} \leftarrow H_1 + \Delta H;
    calculate \delta(H_{\text{curr}}, e^{j\omega}) , \delta(x_{\text{curr}}, e^{j\beta});
    \lambda_p = k_p \delta(H_{\text{curr}}, e^{j\omega}) , \lambda_s = k_s \delta(H_{\text{curr}}, e^{j\beta});
    calculate m;
    H_{1:1,\text{adj}} = \text{Coeff\_Adjust}(H_{1:1, m, n, G});
    calculate \delta(H_{1:1, \text{adj}}, e^{j\omega}) and \delta(H_{1:1, \text{adj}}, e^{j\beta})
    if \( \delta(H_{1:1, \text{adj}}, e^{j\omega}) < \delta_p \) and \( \delta(H_{1:1, \text{adj}}, e^{j\beta}) < \delta_s \)
      add H_{1:1, \text{adj}} into Solution\_list;
  end if
  end if
  choose the best result, H_{\text{best}}, from Solution\_list;
  return H_{\text{best}};
\}

FIGURE 6. Pseudo 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, \( \lambda_p \) and \( \lambda_s \), are initialized to solve the iterative optimization problem. The coefficients of the solution H_1 in each iteration are adjusted to maximize \eta and added into the Solution\_list. The best coefficient H_{\text{best}} with the highest \eta 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 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 magnitude of 0.1 dB and the minimum stopband attenuation of 60 dB.

The function IIR\_Design is called with input values, \( \tau, \delta_p, \delta_s, \omega_1, \omega_2, \omega_3 \) and \( \omega_4 \) set according to the design specifications. The algorithm starts after the following constants have been initialized as follows: \( r_{\text{max}} \) is set to 0.99 for good filter stability; the trust region \( \rho \) and the minimum trust region \( \rho_{\text{min}} \) 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; \( \nu_{\min} \) 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, \( \delta(x_1, e^{j\omega}) \) and \( \delta(x_1, e^{j\beta}) \) are calculated to obtain \( \lambda_p \) and \( \lambda_s \). 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:

\[
\begin{align*}
  &b = \{0.00233785, 0.00496586, 0.00273583, 0.00523338, \\
  &0.00488778, 0.00812862, -0.00131834, \\
  &-0.00696892, 0.0061514, -0.0522488, 0.0323094\} \\
  &a = \{1, 3.20042, 6.57387, 10.0924, 12.4517, 12.7864, \\
  &11.193, 8.2436, 5.20658, 2.78068, 1.23591, \\
  &0.444131, 0.122931, 0.023883, 0.00233415\}
\end{align*}
\]

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...
deviation of 0.00497 and hardware cost of 1294 equivalent FAs. The improvement in group delay deviation, \[ \| \text{dev}_\tau (H_1 + \Delta H_1, \omega) \|_\infty - \| \text{dev}_\tau (H_1, \omega) \|_\infty = 0.009, \]

is larger than \( \nu_{\text{min}} \| \nabla \text{dev}_\tau (H_1, \omega)^T \Delta H_k \|_\infty = 0.005. \)
Hence, \( \rho \) is relaxed from 0.01 to 0.012. Based on (16) and \( \rho \), the value of \( m \) for coefficient adjustment by \( \text{Coeff\_Adjust} \) 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:

\[
b = \{0.00233785, 0.00496587, 0.00273582, 0.000523339, 0.00488779, 0.00812860, -0.000131829, -0.00696891, 0.00839353, 0.0152882, -0.0133790, -0.0130128, 0.0611511, -0.0522489, 0.0323092\}
\]

\[
a = \{1, 3.20033, 6.57387, 10.0924, 12.4517, 12.7865, 11.1194, 8.24336, 5.20661, 2.78062, 1.23592, 0.444108, 0.122931, 0.0238836, 0.00233402\}
\]

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 \( \text{Coeff\_Adjust} \), 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 \( \eta \) of 67857, is added into \( \text{Solution\_list} \).

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

The final solution is chosen from the least cost solution in \( \text{Solution\_list} \). 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.

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

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

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 1001 and 100 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...
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 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 $0.05 \pi \times \text{rad/sample}$. 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 $r_{\text{max}}$ is set to 0.99 to ensure stability after coefficient adjustment. $\rho$ and $\rho_{\text{min}}$ 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 costs. $k_p$ and $k_s$ are both set to $10^{4}$, and $\nabla_{\text{min}}$ 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.
The sensitivity analysis and quantization errors of the eight benchmark filters designed by [17] and [18] and the proposed algorithm are listed in Table 6(A), (B) and (C), respectively. The columns labeled $\Delta$ 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 37th 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 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 $\mu$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) IIRS, (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™. The synthesized areas in µm² 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
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 mW 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.

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 65nm 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


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.

...