Title	Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common subexpressions(Published version)
Author(s)	Xu, Fei; Chang, Chip Hong; Jong, Ching Chuen
Citation	Xu, F., Chang, C. H., & Jong, C.C. (2007). Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common subexpressions. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 26(10), 1898-1907.
Date	2007
URL	http://hdl.handle.net/10220/4641
Rights	

4) In [9, Sec. 3.4.3], the authors have tried to extract all the sentences into an equivalent sentence over $GF(2)$, which is a very cumbersome step. Instead we apply a *hierarchical generation* of the ROFDD, which *reduces the overall time* that is required. This point on time complexity is reflected in the discussion presented in Section IV of our paper.

VI. CONCLUSION

This paper shows a hierarchical technique of generation of ROFDDs for Galois field circuits using the isomorphism between $GF(2^m)$ and $GF(2^n)^p$, where $m = np$. Theoretically, it has been explained how the algorithms lead to savings in terms of time and space according to the resources of the environment. Experimental results have been provided to support the claim. The approach for the hierarchical construction of the DDs may be applied recursively in order to magnify the gains of a single iteration.

REFERENCES

[1] J. Rotman, *Galois Theory*, 2nd ed. New York: Springer-Verlag, 2005.

[2] T. Kropf, *Introduction to Formal Hardware Verification: Methods and Tools for Designing Correct Circuits and Systems*. New York: Springer-Verlag, 1999.

[3] U. Kebschull and W. Rosenstiel, "Efficient graph-based computation and manipulation of functional decision diagrams," in *Proc. Eur. Conf. Des. Autom.*, 1993, pp. 278–282.

[4] U. Kebschull, E. Schubert, and W. Rosenstiel, "Multi-level logic synthesis based on functional decision diagrams," in *Proc. Eur. Conf. Design Autom.*, 1992, pp. 43–47.

[5] B. Becker, R. Drechsler, and M. Theobald, "On the implementation of a package for efficient representation and manipulation of functional decision diagrams," in *Proc. IFIP WG 10.5 Workshop Appl. Reed-Muller Expansion Circuit Des.*, 1993, pp. 162–169.

[6] R. Stankovic and R. Drechsler, "Circuit design from Kronecker Galois field DDs for MV functions," in *Proc. IEEE Int. Symp. Multiple Valued Logic*, 1997, pp. 275–280.

[7] R. Drechsler, A. Sarabi, B. Becker, M. Theobald, and M. K. Perkowski, "Efficient representation and manipulation of switching function based on ordered Kronecker functional decision diagrams," in *Proc. Des. Autom. Conf.*, 1994, pp. 415–419.

[8] H. Hengster, R. Drechsler, S. Eckrich, T. Pfeiffer, and B. Becker, "AND/EXOR based synthesis of testable KFDD-circuit with small depth," in *Proc. ATS*, 1996, pp. 148–154.

[9] S. Morioka, Y. Katayama, and T. Yamane, "Towards efficient verification of arithmetic algorithms over Galois fields $GF(2^m)$," in *Proc. CAV*, 2001, vol. 2102, pp. 465–477.

[10] C. Paar, P. Fleischmann, and P. Roelse, "Efficient multiplier architectures for Galois fields $GF(2^{4n})$," *IEEE Trans. Comput.*, vol. 47, no. 2, pp. 162–170, Feb. 1998.

[11] F. Somenzi, *CUDD: Colorado University Decision Diagram Package*, 1996. [Online]. Available: ftp://vlsi.colorado.edu/pub/

[12] J. Daemen and V. Rijmen, *The Design of Rijndael*. New York: Springer-Verlag, 2002.

[13] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[14] A. Karatsuba and Y. Ofman, "Multiplication of multidigit numbers on automata," *Sov. Phys. Dokl (Engl. transl.)*, vol. 7, no. 7, pp. 595–596, 1963.

[15] V. Rijmen, *Efficient Implementation of the Rijndael Sbox*. [Online]. Available: http://www.esat.kuleuven.ac.be/rijmen/rijndael

[16] S. Morioka and Y. Katayama, "Design methodology for a one-shot Reed-Solomon encoder and decoder," in *Proc. ICCD*, 1999, pp. 60–67.

[17] A. Halbutogullar and C. K. Koc, "Mastrovito multiplier for general irreducible polynomials," *IEEE Trans. Comput.*, vol. 45, no. 5, pp. 503–518, May 2000.

Design of Low-Complexity FIR Filters Based on Signed-Powers-of-Two Coefficients With Reusable Common Subexpressions

Fei Xu, Chip Hong Chang, and Ching Chuen Jong

***Abstract*—In this paper, a new efficient algorithm is proposed for the synthesis of low-complexity finite-impulse response (FIR) filters with resource sharing. The original problem statement based on the minimization of signed-power-of-two (SPT) terms has been reformulated to account for the sharable adders. The minimization of common SPT (CSPT) terms that were considered in our proposed algorithm addresses the optimization of the reusability of adders for two major types of common subexpressions, together with the minimization of adders that are needed for the spare SPT terms. The coefficient set is synthesized in two stages. In the first stage, CSPT terms in the vicinity of the scaled and rounded canonical signed digit (CSD) coefficients are allocated to obtain a CSD coefficient set, with the total number of CSPT terms not exceeding the initial coefficient set. The balanced normalized peak ripple magnitude due to the quantization error is fulfilled in the second stage by a local search method. The algorithm uses a common-subexpression-based hamming weight pyramid to seek for low-cost candidate coefficients with preferential consideration of shared common subexpressions. Experimental results demonstrate that our algorithm is capable of synthesizing FIR filters with the least CSPT terms compared with existing filter synthesis algorithms.**

***Index Terms*—Canonical signed digit (CSD), common subexpression, filter synthesis.**

I. INTRODUCTION

Finite-impulse response (FIR) filters play a vital role in modern communication system because of its versatility, stability, and simplicity [1]–[9]. As multipliers are generally agreed to be a power-hungry device and occupy a large silicon area, the trend toward the design of fixed-point FIR filters is to replace the expensive multiplication operations by simpler additions and hardwired shifters [7]–[13]. The basic principle behind the design of FIR filter for multiplierless implementation is to approximate each filter coefficient with a minimal number of signed-power-of-two (SPT) terms. The process of finding the SPT terms to represent the real coefficient set is called the filter coefficient synthesis. In this way, the filter complexity is determined by the number of additions/subtractions required to implement the multiplications, which, in turn, is directly related to the number of SPT terms used to synthesize the filter coefficients. Thus, the constrained optimization problem becomes one of finding a set of filter coefficients with a minimal number of SPT terms that satisfy a given magnitude response specification [1]–[5], [14]–[24]. A minimum representation refers to a representation of a numeral that has the minimum number of SPT terms. The canonical signed digit (CSD) representation is one of the most commonly used minimum representations in digital filter coefficient synthesis [25], [26]. There exist a number of algorithms for synthesizing CSD coefficients to minimize the number of SPT terms that are required for the implementation of a low-complexity FIR filter [1]–[5], [14]–[24].

The filter transfer function that is to be met by digital implementation is always constrained by the finite wordlength used to represent

Manuscript received September 26, 2005; revised June 30, 2006. This paper was recommended by Associate Editor A. Raghunatan.

F. Xu is with Xilinx Asia Pacific Pte. Ltd., Singapore 486051, Singapore (e-mail: fei.xu@xilinx.com).

C. H. Chang and C. C. Jong are with the Nanyang Technological University, Singapore 639798, Singapore (e-mail: echchang@ntu.edu.sg; eccjong@ntu.edu.sg).

Digital Object Identifier 10.1109/TCAD.2007.895615

the real-valued filter coefficients. In this respect, quantization errors have to be addressed to ensure that the quantized coefficient set still meets the magnitude response specification. Filter coefficient synthesis methodologies can generally be categorized into two types, namely, the optimal approaches and the suboptimal approaches. The mixed integer linear programming (MILP) is frequently employed in the optimal approach [7], [14], [18], [24]. The main drawback of the optimal approach is its formidable computation time, especially for filters with a large number of taps. On the other hand, the suboptimal approaches do not guarantee optimal results, but quasi-optimal results are obtained in reasonable time. The most popular suboptimal approach is the local search approach. It is based on the observation that the desired solution usually lies in the vicinity of the real-valued coefficient set. Hence, the heuristic search starts with the real-valued coefficient set and search in its neighborhood to obtain the minimal solution. The computation time for the local search method depends very much on the neighborhood size. Kodek and Steiglitz [14] mentioned in their paper that the bivariate search, where the rounded coefficients are varied by ± 1 quantization step size, yields better results than a univariate local search. Thereafter, the bivariate method has become a common method for constructing the neighborhood around the scaled and rounded coefficients in a local search, although the amount of quantizations enumerated varies in different methods. Samueli [21] proposed a bivariate local search method whereby the filter response is simulated for every change in a coefficient. Chen and Willson [5] developed a two-stage local search scheme by applying the trellis search algorithm. Instead of rigidly limiting the number of SPT terms per coefficient, this algorithm allocates the SPT terms dynamically to locations that best improve the filter's frequency response. The computation time of the local search algorithm does not grow exponentially as the filter length increases. However, there is a risk of stuck at the local minimum. This problem is very difficult to tackle, and there is no solution to it, according to the authors, other than adjusting the initial set of coefficients. Other suboptimal approaches are the tree search with least squares optimization [15], [17], stochastic optimization [2], [4], and quantization by coefficient sensitivity [1], [22] among many others.

After the filter synthesis step, the coefficient set can be implemented in different filter structures. The transposed direct form structure has been the preferred architecture over direct form structure for high-speed high-order fixed filter implementation since its critical path delay is independent of the number of taps. The transposition theorem recasts the inner-product-based transfer function into a single-input–multiple-output combinational circuit and an accumulator-delay line. The system of constant mutliplier circuits can be modeled as a multiple constant multiplications (MCM) block [10] in high-level synthesis where redundancy is sought for hardware reduction [6], [9]–[11]. The common subexpression elimination (CSE) algorithms [9]–[11] search for common subexpressions to maximize the reuse of the products of the input and common subexpressions. The SPT terms in the coefficients are merged to form the common subexpressions, and the hardware cost is no longer proportional to the number of SPT terms. The hardware cost, which is simplified to the number of adders, is determined by the number of occurrences of common subexpressions and the number of spare SPT terms. This poses the skepticism to earlier postulation that a coefficient set with less SPT terms will also yield a solution with a less adder cost. The excess SPT terms may compose more common subexpressions that can be shared across coefficients and, hence, reduce the adder cost for the implementation of the FIR filter. A few researchers have tried the integration of the CSE step with the synthesis step. Chen and Willson [5] mentioned additionally that they had merged the common subexpressions 101 and $10\bar{1}$ before trellis search, and the results showed that the merge–trellis search realized a saving of 18%–23% of the carry–save adders (CSAs) for the CSA-tree-based transposed direct form architecture. Yli-Kaakinen and Saramaki's algorithm [24] involved the sharing of a higher weight common subexpression. Yli-Kaakinen and Saramaki's algorithm first generates a lookup table containing all possible power-of-two numbers for a given wordlength and a given maximum number of SPT terms per coefficient, and then a branch-and-bound procedure is applied to evaluate the coefficient sets. Existing CSE algorithms are applied in Yli-Kaakinen and Saramaki's algorithm. A MILP problem for designing FIR filters that incorporate subexpression sharing was formulated by Gustafsson and Wanhammar [7]. In [19], Rosa *et al.* also proposed an algorithm that performs CSE on selected coefficient sets of weight n based on the filter specifications. In [3], the number of adders from CSE is used as a criterion in their design of peak-constrained least squares FIR filters. However, these algorithms either restrict the occurrences of common subexpressions by merging SPT terms before the minimization process based on an initial coefficient set or relying on an exhaustive search for the identification of common subexpressions, which become inefficient for large coefficient sets.

This paper presents a new algorithm for the design of a low-complexity FIR filter that blends common subexpression elimination with the filter coefficient synthesis algorithm right from the magnitude response specification [12]. We treat the synthesis problem with the consideration of common subexpressions and solve it heuristically in two stages. Two most frequently used common subexpressions 101 and $10\bar{1}$ and spare SPT terms in each coefficient are treated as common SPT (CSPT) terms to be optimized simultaneously. By limiting the number of CSPT terms to no more than that of the rounded CSD coefficient set, the first stage of our algorithm has the freedom to allocate any CSD coefficient in the vicinity of the rounded coefficient set to increase the occurrences of the two chosen common subexpressions while reducing the total number of spare SPT terms in the minimal CSD coefficient set. The unfulfilled normalized peak ripple magnitude (NPRM) in the first stage due to the quantization error is compensated in the second stage by an efficient wordlength-dependent adaptive neighborhood search method. The solutions generated by our algorithm can also be further optimized to lower the adder cost by applying higher weight common subexpressions sharing as in the work of Yli-Kaakinen and Saramaki's [19].

This paper is organized as follows. Section II states the problem formulation of our filter synthesis algorithm. The notion of hamming weight pyramid (HWP) and its modified version is introduced in Section III, where our proposed two-stage integrated algorithm is also presented. We compare our algorithm with other algorithms using two well-known design examples in Section IV. The paper is concluded in Section V.

II. Coefficient Synthesis Problem Formulation

For fixed-point FIR filter implementation, the frequency response error due to truncation or rounding of real-valued filter coefficients is to be minimized. In most practical applications, the magnitude response of an FIR filter needs not be zero at stopband nor be constant over the entire passband. To synthesize a set of finite wordlength coefficients, the absolute value of the passband gain is immaterial. What matters most is the relative attenuation between the passband and stopband. Therefore, the NPRM, which is defined as the peak ripple magnitude divided by passband gain [18], is often used as a parameter to guide the synthesis of filter coefficients. That is,

$$\mathrm{NPRM} = \max\left\{\frac{\delta_p\omega_p}{g}, \frac{\delta_s\omega_s}{g}\right\} \tag{1}$$

where g is the average passband gain, δ_p and δ_s are the passband and stopband ripples, respectively, and ω_p and ω_s are the bandedge frequencies at the end of passband and the beginning of stopband, respectively.

The coefficient multiplications involved in the convolution determine the complexity of digital filters. The hardware cost can be reduced by representing the coefficients as sums of SPT terms to replace the multiplications by shift-and-add circuits. A filter coefficient $h_{\mathrm{spt}}(i)$ can be expressed as a sum of SPT terms [5], i.e.,

$$h_{\mathrm{spt}}(i) = \sum_{k=1}^{L_i} s_{i,k} 2^{-p_{i,k}} \tag{2}$$

where $s_{i,k} \in \{-1, 1\}$ is the kth SPT term at bit position $p_{i,k} \in \{1, 2, \ldots, B\}$. It is assumed without loss of generality that $0 \leq |h_{\mathrm{spt}}(i)| < 1$, and $h_{\mathrm{spt}}(i)$ has a wordlength of B bits and comprises L_i SPT terms.

Thus, the fundamental optimization problem becomes one of synthesizing a set of coefficients for an N-tap filter that minimizes the total number of SPT terms J, i.e.,

$$J = \sum_{i=1}^{N} \sum_{k=1}^{L_i} |s_{i,k}| \tag{3}$$

subject to the objective function of the minimal NPRM constraint depicted in (1).

The minimization criterion J of (3), which is adopted by most filter coefficient synthesis algorithms, does not consider the possibility of sharing common subexpressions to reduce the adder cost. The detection of good common subexpressions to eliminate redundant computations can further reduce the implementation complexity.

CSD representation has the least SPT terms. Asymptotically, an n-bit CSD number can be broken down into $n/18 + O(1)$ pairs of $10\bar{1}$ subexpression, $n/18 + O(1)$ pairs of 101 subexpression, and $n/9 + O(1)$ isolated 1 or $\bar{1}$. Referring to a $10\bar{1}$ pair, a 101 pair, or an otherwise isolated nonzero digit as a term, it can be proved that the asymptotic expected number of terms in an n-bit CSD number is $2(n+1)/9$. This represents a 33% saving compared with the total number of nonzero digits, which is $(3n+1)/9$ [9]. By considering the shortest two nonzero digit common subexpressions 101 and $10\bar{1}$ in this paper, the operand width and, hence, the area and time complexity of the adder used for their generation, is reduced. In addition, neglecting other common subexpressions also reduces the search space of the algorithm and simplifies layout. Common subexpressions with a low frequency of reuse increase the difficulty to maintain a well-structured filter layout style. Hartley [9] showed that substantial hardware reduction (about 30%) can be achieved by finding the occurrences of only the two common subexpressions 101 and $10\bar{1}$, with little adverse effect on routability.

By extracting the common subexpressions $x_{101} = x \ll 2 + x$ and $x_{10\bar{1}} = x \ll 2 - x$ from (2), the general expression for $h_{\mathrm{spt}}(i)$ can be rewritten as

$$\begin{aligned} h_{\mathrm{cspt}}(i) = & \sum_{k=1}^{L_{101,i}} cs_{i,k}^{101}\left(2^{-p_{i,k}^{101}+2} + 2^{-p_{i,k}^{101}}\right) \\ & + \sum_{k=1}^{L_{10\bar{1},i}} cs_{i,k}^{10\bar{1}}\left(2^{-p_{i,k}^{10\bar{1}}+2} - 2^{-p_{i,k}^{10\bar{1}}}\right) + \sum_{k=0}^{L'_i} s'_{i,k} 2^{-p'_{i,k}} \end{aligned} \tag{4}$$

where $cs_{i,k}^{101}$, $cs_{i,k}^{10\bar{1}} \in \{1, \bar{1}\}$ are the signs of the common subexpressions 101 and $10\bar{1}$ that are present in the ith coefficient with its least significant digit located at positions $p_{i,k}^{101}$ and $p_{i,k}^{10\bar{1}}$, respectively. $s'_{i,k} \in \{1, \bar{1}\}$ are isolated SPT terms (not belonging to any common subexpression) in the ith coefficient located at digit position $p'_{i,k}$. $p_{i,k}, p'_{i,k} \in \{1, 2, \ldots, B\}$ for a B-bit coefficient. $L_{101,i}$ and $L_{10\bar{1},i}$ are the number of common subexpressions of types 101 and $10\bar{1}$, respectively. $L'_i = L_i - 2(L_{101,i} + L_{10\bar{1},i})$ is the number of SPT terms of the ith coefficient that are not present in any common subexpression, whereas L_i is the total number of SPT terms.

An adder is required to generate each of the common subexpressions 101 and $10\bar{1}$ once before these common subexpressions can be shared among the coefficients that embrace them. Each isolated SPT term that is not part of a common subexpression requires one adder. Therefore, the total number of adders required to implement the multiplication block of a transposed direct form filter is bounded by

$$\mathrm{Cost} \leq 2 + \sum_{i=1}^{N} \left(L'_i + L_{101,i} + L_{10\bar{1}}\right). \tag{5}$$

The equality of (5) happens when each of 101 and $10\bar{1}$ occurs at least twice. The adder cost is minimized by maximizing the values of $L_{101,i}$ and $L_{10\bar{1},i}$ bounded by the allowable number of SPT terms $\sum_{i=1}^{N} L_i$ of the coefficient set. Therefore, the objective of our proposed SPT coefficient synthesis algorithm for low-complexity FIR filters will be recast to that of reducing the combined number of common subexpressions and nonsharable SPT terms L' in each coefficient, i.e.,

$$J' = \sum_{i=1}^{N} L'_i + L_{101,i} + L_{10\bar{1},i} \tag{6}$$

subject to the minimal NPRM constraint of the desired amplitude response specification. Therefore, the number of CSPT terms $L'_i + L_{101,i} + L_{10\bar{1},i}$ instead of L_i becomes the new minimization criterion.

The advantage of having filter coefficients with many shared common subexpressions is that under the same limit of adders allowed for each tap, effectively more SPT terms can be assigned to a coefficient to improve the coefficient's precision, thereby leading to a better approximation to the desired frequency response characteristics.

III. The Proposed Algorithm

As the desired objective function of minimizing the NPRM is nonlinear [18], heuristic optimization of multiple constraints in a single run is susceptible to the local minimum problem due to poor initial condition. If a reasonable upper bound on the CSPT terms based on the rounded CSD coefficients is imposed, a quality initial CSD coefficient set can be obtained. This preliminary coefficient set may not meet the desired amplitude response specification. Its NPRM can be further optimized until the specification is met by slightly relaxing the allowable number of CSPT terms. These two processes require the location of CSD numbers with a specific number of CSPT terms or the determination of the number of CSPT terms of a specific CSD number. Therefore, we devise a search tool for the CSD numbers that specially treats subexpressions 101 and $10\bar{1}$.

In [26], we proposed an HWP to succinctly compress the information about the distribution of the hamming weights of CSD numbers, and it can serve as a useful search tool for the nearest neighbors to a CSD number with a similar or lower hamming weight. Fig. 1 shows the pyramid of the hamming weights of CSD numbers $2^r + c$, where r and c correspond to the row number and the offset from the center column of the pyramid, respectively. By reading the weights in a top-down and left-right order, the integers in the pyramid correspond to the CSD representations of natural number sequence 1, 2, 3, 4,. . ..

Row r \ Column c	…	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	…
0												1											
1												1											
2											2	1	2										
3										2	2	1	2	2									
4							3	2	3	2	2	1	2	2	3	2	3						
5		3	3	2	3	3	3	2	3	2	2	1	2	2	3	2	3	3	3	2	3	3	
…												…											

Fig. 1. HWP.

Row r \ Column c	…	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	…
0												1											
1												1											
2											1	1	1										
3										2	2	1	2	2									
4							2	2	2	2	2	1	2	2	2	2	2						
5		3	3	2	3	3	2	2	2	2	2	1	2	2	2	2	2	3	3	2	3	3	
…												…											

Fig. 2. CS-HWP.

In order to account for the saving of one adder by considering 101 and $10\bar{1}$ as common subexpressions, the HWP of Fig. 1 needs to be revamped so that it can be useful for seeking nearest CSD numbers with an equal or lower adder cost. Based on the adder cost of (5), it would be logical to reduce the hamming weight by one for each nonoverlapping string of either 101 or $10\bar{1}$ encountered in a CSD number. The resultant HWP is shown in Fig. 2, and it is called the common-subexpression-based HWP (CS-HWP). For example, the CSD number of decimal 19 is $1010\bar{1}$. It corresponds to the entries in row 4 column 3 of HWP and CS-HWP. Its hamming weight is 3 in Fig. 1 but is reduced to 2 in Fig. 2 due to the presence of subexpression $10\bar{1}$. In short, each nonoverlapping string of 101 and $10\bar{1}$ is treated as a single CSPT term, and the number of CSPT terms can be obtained from the CS-HWP.

In what follows, a two-stage algorithm for the design of a low-complexity FIR filter is presented. In the first stage, a set of CSD coefficients in the continuum of the rounded CSD coefficients with a minimal filter response error and yet contains no more CSPT terms than the rounded CSD coefficient set is generated. The balanced NPRM budget to the desired response due to the quantization error of the obtained CSD coefficient set is compensated in the second stage. The first stage is called the *CSPT coefficient allocation stage*, and the second stage is called the *quantization compensation stage*.

A. CSPT Coefficient Allocation

In [17], Lim and Parker mentioned that, for a minimax design, the optimal coefficient set is usually not the one that is obtained by rounding but the optimal CSD coefficients do lie near the rounded ones. During our experiments, we found that the rounding process has a strong influence on the filter frequency response. Rounding a real-valued coefficient set fulfilling an amplitude response specification to its finite wordlength CSD coefficients are quite likely to fail under the same specification. The aim of this allocation stage is to search for a good initial CSD coefficient set with a reduced number of CSPT terms by considering common subexpressions.

First, the real-valued coefficient set $\{h\}$ fulfilling the frequency response specification is quantized to the desired wordlength in CSD representation. From the rounded CSD coefficient set $\{h_{\mathrm{csd}}\}$, new CSD coefficients are added to expand the candidate pool (CP) by finding the nearest CSD coefficients to each $h_{\mathrm{csd}}(i)$ for $i = 1, 2, \ldots, N$, with the number of CSPT terms equal to or smaller than those of $h_{\mathrm{csd}}(i)$. With the help of CS-HWP, the required nearest CSD coefficients can easily be found by searching from two sides of the original coefficient location in the CS-HWT for entries with an equal or lower number of CSPT terms. Let the coefficients in the CP be designated as $h_{\mathrm{cspt}}(i)(j)$ for $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$. $M = 2\mathrm{ncspt}(i)$, where $\mathrm{ncspt}(i)$ is the number of CSPT terms of $h_{\mathrm{csd}}(i)$. In other words, one nearest CSD number with an equal number of CSPT terms and two nearest CSD numbers with a smaller number of CSPT terms are sought for each coefficient $h_{\mathrm{csd}}(i)$. We found from experimentation that this number of nearest CSD coefficients is adequate for most amplitude response specifications. By having a larger value of M, more coefficients with larger quantization errors will be included, but these additional CSD coefficients are rarely selected eventually as the achievable NPRM error has already reached its margin of diminishing return. Even if some of them have been included in the final solution of this stage, the improvement that they made to the NPRM of the solution is insignificant to warrant the extra computation.

Next, a preliminary minimal filter coefficient set $\{h_p\}$ is established by setting all its coefficient values to zero. The initial coefficients of $\{h_p\}$ either remain as zero or are gradually substituted by the coefficients of the CP $\{h_{\mathrm{cspt}}\}$ in an iterative process to reduce the NPRM of $\{h_p\}$. Each iteration increments the total number of CSPT terms of $\{h_p\}$ by one. At any iteration r, a number of coefficient sets can be formed by replacing an entry in the current $\{h_p\}$ by one from the CP such that the total number of CSPT terms of the coefficient set so formed is equal to r. The best coefficient set with the lowest NPRM error from among the eligible sets is selected to replace $\{h_p\}$. At the end of each iteration, all candidate coefficients with the same number of CSPT terms from the same tap as the substituting coefficient, including the substituting coefficient itself, are deleted from the CP. The process continues until the CP is exhausted.

This search for better quality CSPT terms for each coefficient is in congruence with the practice of many popular search methods in filter synthesis [1], [5], [22], except that common subexpressions have been considered. We keep a record of the minimum NPRM, which is denoted by $\mathrm{NPRM}_{\min}$, and its corresponding coefficient set $\{h_{p\min}\}$. Upon the termination of CSPT coefficient allocation stage, the coefficient set $\{h_{p\min}\}$ with the minimum NPRM throughout the stage will be used for the second stage. This stage is highly efficient because $\{h_p\}$ is small, and the coefficients in the CP have been prudently chosen with the aid of CS-HWT. The computation time is $O(MN/2)$, which is well bounded by the cardinality of $|h_{\mathrm{cspt}}|$. Although the search is, by no means, exhaustive, the conditional backtracking helps to avoid a poor local minimum.

The size of the CP marks the frequency characteristics of initial coefficient set for the second stage. Hence, it should be defined to leave an error margin that is small enough to ensure that the second stage could converge to the desired NPRM, but sufficiently large that the first stage does not incur excessive adder cost. Usually, there is no deterministic answer to what constitutes the best starting coefficient set for the local search. Chen and Willson [5] allocated up to about 75% of the SPT terms in the first pass of local search without having an adverse effect on the final result. In contrast to his heuristic local search methods, our criterion for the initial coefficient set is defined based on the mediation of amplitude response specification and the number of CSPT terms. It is normal that the $\{h_{p\min}\}$ obtained in the first stage based on the rounded CSD coefficients will not meet the NPRM requirement. Therefore, adaptive quantization compensation is applied in the second stage based on the balanced NPRM budget while suppressing the growth of the number of CSPT terms. The amount of compensation needed and, hence, the effort required by the optimization method, depends on the quality of its preliminary coefficient set. A sensible NPRM budget can be empirically determined in the first stage. Given the tradeoff between the number

TABLE I
NEIGHBORHOOD FOR QUANTIZATION ERROR COMPENSATION

Wordlength, B	Discrete SPT terms, Δq	Δq_{max}
8	$\pm 2^{-7}, \pm 2^{-8}$	2^{-7}
9	$\pm 2^{-8}, \pm 2^{-9}$	2^{-8}
10	$\pm 2^{-8}, \pm 2^{-9}, \pm 2^{-10}$	2^{-8}
11	$\pm 2^{-9}, \pm 2^{-10}, \pm 2^{-11}$	2^{-9}
12	$\pm 2^{-9}, \pm 2^{-10}, \pm 2^{-11}, \pm 2^{-12}$	2^{-9}
13	$\pm 2^{-10}, \pm 2^{-11}, \pm 2^{-12}, \pm 2^{-13}$	2^{-10}
14	$\pm 2^{-11}, \pm 2^{-12}, \pm 2^{-13}, \pm 2^{-14}$	2^{-11}

of CSPT terms and the acceptable quantization error, we avoid a high adder cost at this stage by setting an upper bound on the maximum number of CSPT terms to that of the rounded coefficient set, and we further limit the size of the CP to MN to reduce inefficient computation for an insignificant improvement in NPRM.

B. Quantization Compensation

In the second stage, the frequency response characteristic of the optimal coefficient set $\{h_{p\,\min}\}$ with $\text{NPRM}_{\min}$ is further improved by modifying the least significant digits of the coefficients until the desired NPRM has been reached. The optimization of frequency response shall be accomplished with little or no penalty to the adder cost. In fact, with the consideration of CSPT terms, the number of adders may even be reduced from the primitive set $\{h_{p\,\min}\}$ upon achieving the desired amplitude response specification. There are two main concerns in this stage: the definition of neighborhood and the search strategy.

The neighborhood defines a boundary within which small discrete SPT terms are introduced to adjust the coefficient values to compensate for the frequency response error. It determines the degree of freedom of the search procedure. A larger neighborhood provides a better chance of finding the optimal coefficient set, but it also leads to a longer computation time. For short wordlength coefficients, allowing a large number of LSDs may defeat the purpose of having local search, as opposed to exhaustive search. However, there is no reason to constrain the search space to merely the last two LSDs for long wordlength coefficients as in the work of Chen and Willson [5]. For example, for wordlength of 10 bits, the neighborhood defined by [5] is a circumference of radius of 2^{-9} around the preliminary coefficient set. This is sufficient for many coefficient sets but deficient for some other situations. If the desired wordlength is 14 bits, the neighborhood will be a circumference of radius of 2^{-13}. This search space is trivial to improve the quality of the solution. In our proposed method, the number of LSDs for the neighborhood is wordlength dependent. More LSDs are chosen for a longer wordlength to ensure that the search space is sufficiently large for optimization, but small enough to maintain efficiency of the algorithm. The number of LSDs is determined empirically. To find the suitable neighborhood size, we tested filters of different coefficient wordlengths, each optimized with different neighborhood sizes, from two to five LSDs. We found that, for some filters with longer coefficient wordlength, the search process converges faster when the neighborhood size reaches a critical number of LSDs than when a smaller search space is used. There are also cases where no solution can be found or the algorithm fails to converge when the size of the neighborhood is very small. Table I lists all LSDs from $\pm 2^{-m}$ to $\pm 2^{-B}$ in the circumference of critical radius m, which is determined empirically for coefficients of wordlength B for $B = 8$–14. The neighborhoods in Table I are presented in discrete SPT terms Δq and the maximum absolute quantization step $\Delta q_{\max}$.

The frequency response of the compensated coefficient set $\{h_{qc}\}$ is iteratively improved as discrete SPT terms in Table I are added

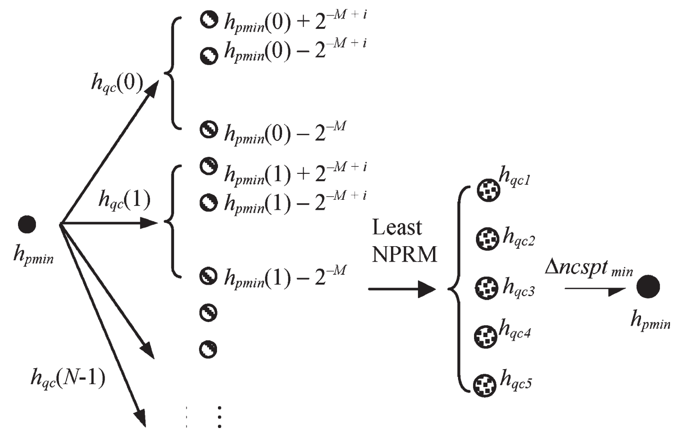

Fig. 3. One iteration of quantization compensation stage.

to compensate for the quantization error. As the coefficient values dynamically change in this process, unrestrictive compensation can lead to the overflow or underflow of the coefficient values beyond the maximum compensation that is defined by the neighborhood of the initial coefficient set $\{h_{p\,\min}\}$. Therefore, the range of the ith compensated coefficient is limited to the following upper and lower bounds at all time:

$$h_{p\,\min}(i) - 2\Delta q_{\max} \leq h_{qc}(i) \leq h_{p\,\min}(i) + 2\Delta q_{\max}. \quad (7)$$

The upper and lower bounds in (7) are calculated once from $\{h_{p\,\min}\}$ of the first stage, and out-of-range coefficients are pruned in each iteration. The restrictive search narrows down the search space substantially. Therefore, the computational time for each iteration reduces until convergence.

The procedure for quantization compensation is detailed as follows. To each coefficient $h_{p\,\min}(i)$, a number of compensated coefficients are created by adding the discrete SPT terms that are defined by the adaptive neighborhood of Table I. Compensated coefficient values falling outside the range of (7) are discarded. The NPRM of every combination of coefficient set that can be generated from the pool of compensated coefficients is evaluated, and the five best NPRM coefficient sets are stored as $\{h_{qc1}\}, \{h_{qc2}\}, \ldots, \{h_{qc5}\}$. The numbers of CSPT terms for these coefficient sets can be extracted from the CS-HWT and stored in a list $\{\text{ncspt}_1, \text{ncspt}_2, \ldots, \text{ncspt}_5\}$. The increment (or decrement) in the number of CSPT terms of these five best compensated coefficient sets from that of $\{h_{p\,\min}\}$ is recorded as $\{\Delta\text{ncspt}_1, \Delta\text{ncspt}_2, \ldots, \Delta\text{ncspt}_5\}$. Δncspt is positive for an increment and negative for a decrement in the number of CSPT terms. The five compensated coefficient sets are sorted in ascending order of Δncspt. If there is a tie, the sets are further sorted in ascending order of NPRM. The coefficient set with the minimum Δncspt is selected to replace $\{h_{p\,\min}\}$ as the starting coefficient set for the next iteration until the desired NPRM is fulfilled by the selected $\{h_{p\,\min}\}$. Fig. 3 shows the trace of an iteration.

C. Pseudocode and Example

The pseudocode of the two-stage algorithm is given in Fig. 4. In Fig. 4, $\mathbf{CSD}(\{h\})$ rounds the coefficients of $\{h\}$ from real values to the nearest CSD numbers. $\mathbf{CSHWP}(\{h_{\text{csd}}\})$ searches in the CS-HWT for the nearest CSD numbers with CSPT terms that are less than or equal to those of $\{h_{\text{csd}}\}$. The NPRM specification is given by the constant $desired\_NPRM$. $\mathbf{NPRM}(\{h\})$ returns the NPRM of the coefficient set $\{h\}$, and $\mathbf{NCSPT}(\{h\})$ returns the total number of CSPT terms of $\{h\}$. The statement $\{h_p\} + h_{\text{cspt}}(i)(j) - h_p(i)$ means

```
two-stage search({h}, desired_NPRM)
{
    {h_csd} = CSD({h}); {h_cspt} = CSHWP({h_csd});
    initialize {h_p} to a list of all zero coefficients;
    {h_pmin} = {h_p}; min_NPRM = 0;
    % Coefficient Allocation Stage
    for (r = 1 to NCSPT({h_cspt })) {
        for all h_cspt(i)(j) with NCSPT({h_cspt(i)(j)}) = NCSPT({h_p(i)}) + 1
            {h_p}_i,j = {h_p} + h_cspt(i)(j) – h_p(i);
        {h_p} = {{h_p}_i,j | (i, j) = arg[min(NPRM({h_p}_i,j))]};
        if (NPRM({h_p}) < min_NPRM) {
            {h_pmin} = {h_p}; min_NPRM = NPRM({h_p}); } }
    % Quantization Compensation Stage
    for (i = 1 to N)
        l_limit(i ) = h_pmin(i) – 2Δq_max; u_limit(i) = h_pmin(i) + 2Δq_max;
    while (NPRM({h_pmin}) > desired_NPRM)  {
        for (i = 1 to N)
            for (Δq from 2^-B to Δq_max)  % Δq from Table 1.
                if(h_pmin(i)+Δq>l_limit(i) and h_pmin(i)+Δq<u_limit(i))
                    {h_pmin(i)} = {h_pmin(i)} + {h_pmin(i)+Δq} – h_pmin(i);
        list all possible coefficients sets, {h_qc} with h_qc(i) from {h_pmin(i)};
        Find five best NPRM coefficient sets, {{h_qc1}, {h_qc2}, …, {h_qc5}};
        for (i = 1 to 5) Δncspt_i = NCSPT({h_qci}) – NCSPT({h_pmin});
        Sort the five sets in ascending order of Δncspt then NPRM;
        {h_pmin} = first coefficient set in the sorted list;   }
    return {h_pmin};
}
```

Fig. 4. Pseudocode of the proposed algorithm.

replacing the coefficient $h_p(i)$ in $\{h_p\}$ with $h_{\text{cspt}}(i)(j)$. Compared with the exhaustive search or optimal methods, our algorithm avoids the search space from growing exponentially. The shortcoming of any heuristic algorithm is that, for a coefficient set with a large quantization error in the conversion from real-valued coefficients to finite wordlength coefficients, particularly for small wordlength, the algorithm may converge to an NPRM that misses the desired amplitude response specification by a small margin. Our algorithm relays on the optimality of each choice greedily and is not absolutely free from local minimum problem. One possible way to break away from the occasional local minimum problem is to expand the search scope in each iteration, but this will inevitably introduce more computations. Many of these additional computations are fruitless for most coefficient sets that can be easily managed by the present approach. Conditional backtracking with expansion of search scope based on the assessment of the NPRM margin of first stage and the descending rate of NPRM in the second stage may be feasible.

A simple three-tap filter with coefficients $h(1) = 0.0051$, $h(2) = 0.0341$, and $h(3) = -0.1419$ is used to illustrate the proposed design procedure. For a wordlength of 10 bits, its rounded CSD coefficients are $h_{\text{csd}}(1) = 2^{-8} + 2^{-10}$, $h_{\text{csd}}(2) = 2^{-5} + 2^{-8} - 2^{-10}$, and $h_{\text{csd}}(3) = -2^{-3} - 2^{-6} - 2^{-10}$. The normalized CSD coefficient array is given by

$$h_{\text{csd}} = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \bar{1} \\ 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 & 0 & 0 & \bar{1} \end{Bmatrix}. \quad (8)$$

Since $\text{ncspt} = \{1, 2, 3\}$, new CSD coefficients are sought from the CS-HWP to form the following CP in the *CSPT coefficient allocation stage*:

$$h_{\text{cspt}}(1) = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{Bmatrix}$$

$$h_{\text{cspt}}(2) = \begin{Bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \bar{1} \end{Bmatrix}$$

$$h_{\text{cspt}}(3) = \begin{Bmatrix} 0 & 0 & \bar{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{1} & 0 & \bar{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 & \bar{1} & 0 & 0 \\ 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 \\ 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 & 0 & 0 & \bar{1} \end{Bmatrix}. \quad (9)$$

The cardinality of $h_{\text{cspt}}(i)$ for the ith coefficient in the CP is twice that of $\text{ncspt}(i)$. The preliminary coefficient set $\{h_p\}$ is initialized to $\{0, 0, 0\}$. For the first iteration, six candidate coefficient sets with one CSPT term are formed by substituting a coefficient in $\{h_p\}$ by a coefficient from the CP with one CSPT term. The qualified coefficients from the CP include two candidates in $\{h_{\text{cspt}}(1)\}$ to substitute for $h_p(1)$, the first two candidates in $\{h_{\text{cspt}}(2)\}$ for $h_p(3)$, and $\{h_{\text{cspt}}(3)\}$ for $h_p(3)$. As $\{h_{\text{cspt}}(i)\}$ is sorted in ascending order of the number of CSPT terms, the first two terms of $\{h_{\text{cspt}}(i)\}$ are evaluated for the prospective $\{h_p\}_{i,j}$ in each iteration. The coefficient set containing $h_{\text{cspt}}(3)(2)$ has the least NPRM, and it is selected to replace $\{h_p\}$ for the next iteration. $\{h_p\} = \{0, 0, -2^{-3} - 2^{-5}\} = \{0, 0, -0.1536\}$. The coefficient $h_p(3) = -2^{-3} - 2^{-5}$ is a common subexpression $\bar{1}0\bar{1} \equiv 101$, and it is deemed to have only one CSPT term. The two coefficients $h_{\text{cspt}}(3)(1)$ and $h_{\text{cspt}}(3)(2)$ of one CSPT term are then deleted from $\{h_{\text{cspt}}(3)\}$. In the second iteration, the number of CSPT terms of $\{h_p\}$ is incremented to two. Two possibilities exist: either $h_p(1)$ or $h_p(2)$ is replaced with a candidate coefficient of one CSPT term from $\{h_{\text{cspt}}(1)\}$ or $\{h_{\text{cspt}}(2)\}$, or $h_p(3)$ is replaced with a candidate coefficient of two CSPT terms from $\{h_{\text{cspt}}(3)\}$. Altogether, six candidate coefficient sets can be generated from the first two candidates of $\{h_{\text{cspt}}(1)\}$, $\{h_{\text{cspt}}(2)\}$, and $\{h_{\text{cspt}}(3)\}$. The candidate coefficient set containing the second candidate of $\{h_{\text{cspt}}(3)\}$ is found to have the minimum NPRM. Consequently, $\{h_p\} = \{0, 0, -2^{-3} - 2^{-6} - 2^{-8}\} = \{0, 0, -0.1445\}$. Two more coefficients in $\{h_{\text{cspt}}(3)\}$ of two CSPT terms are deleted, and the updated CP is given by

$$h_{\text{cspt}}(1) = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{Bmatrix}$$

$$h_{\text{cspt}}(2) = \begin{Bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \bar{1} \end{Bmatrix}$$

$$h_{\text{cspt}}(3) = \begin{Bmatrix} 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 \\ 0 & 0 & \bar{1} & 0 & 0 & \bar{1} & 0 & 0 & 0 & \bar{1} \end{Bmatrix}. \quad (10)$$

The process continues until $\{h_{\text{cspt}}\}$ is exhausted. The *CSPT coefficient allocation stage* terminates with $\{h_{p\,\min}\} = \{2^{-8}, 2^{-5} + 2^{-8} - 2^{-10}, -2^{-3} - 2^{-6} - 2^{-8}\} = \{0.0039, 0.0341, -0.1445\}$. The value of the ith coefficient in the *quantization compensation stage* is restricted to the range $(h_{p\,\min}(i) - 2^{-7}, h_{p\,\min}(i) + 2^{-7})$, as given in (7). The lower and upper limits for each coefficient are given by $l\_\text{limit} = \{-0.0039, 0.0263, -0.1523\}$ and $u\_\text{limit} = \{0.0117, 0.0419, -0.1367\}$, respectively. The {ncspt} for $\{h_{p\,\min}\}$ is $\{1, 2, 2\}$. Each coefficient is allowed to be compensated by six different discrete SPT terms $\{\pm 2^{-8}, \pm 2^{-9}, \pm 2^{-10}\}$ from Table I. The NPRMs of 18 compensated coefficient sets are evaluated from which the five least NPRM coefficient sets are extracted.These coefficient sets, which are sorted in ascending order of NPRM, are: $\{h_{p\,\min}(1),$

$h_{p\min}(2), h_{p\min}(3) - 2^{-10}\}$, $\{h_{p\min}(1), h_{p\min}(2) - 2^{-10}, h_{p\min}(3)\}$, $\{h_{p\min}(1), h_{p\min}(2) - 2^{-9}, h_{p\min}(3)\}$, $\{h_{p\min}(1), h_{p\min}(2), h_{p\min}(3) + 2^{-10}\}$, and $\{h_{p\min}(1) + 2^{-10}, h_{p\min}(2), h_{p\min}(3)\}$. Their numbers of CSPT terms are, respectively, $\{1, 2, 2\}$, $\{1, 2, 2\}$, $\{1, 2, 2\}$, $\{1, 2, 3\}$, and $\{1, 2, 3\}$. Therefore, $\{\Delta\text{ncspt}\} = \{1, 0, 0, 1, 0\}$. The second, third, and fifth sets have maintained the same number of CSPT terms because the compensations have either generated new common subexpressions or relocated some existing common subexpressions. Among these three least Δncspt sets, the second set with the smallest NPRM is selected. $\{h_{p\min}\}$ is updated to $\{2^{-8}, 2^{-5} + 2^{-9}, -2^{-3} - 2^{-6} - 2^{-8}\} = \{0.0039, 0.0332, -0.1445\}$ and $\{\text{ncspt}\} = \{1, 2, 2\}$. The process iterates until the required NPRM is met.

IV. Experimental Results

In this section, we demonstrate the capability of our proposed algorithm through the implementation of two practical design examples. The specifications of these two filters are frequently used as benchmarks for evaluating the optimality of different digital filter coefficient synthesis algorithms. The comparisons are based on the results taken from [5] and [8]. The metric used for benchmarking the performances of different solutions are the number of CSPT terms, which provides an indication of the adder cost, and the NPRM, which provides an indication of the quality of fitness to the actual frequency responses. For fair comparison, we explicitly account for the sharing of common subexpressions 101 and $10\bar{1}$ for the number of CSPT terms of the coefficient sets generated by those algorithms that have not considered common subexpressions.

Example 1: The halfband filter is part of a programmable digital decimation filter using the sharpened cascade integrator comb architecture with the frequency response specification that is designed for wideband satellite communication systems [27]. The normalized passband and stopband edge frequencies are 0.2π and 0.8π, respectively. The coefficient wordlength is 14 bits, and the filter length is 15. The stopband attenuation is -80 dB. We first design an equiripple low-pass filter using Matlab Simulink based on the Parks–McClellan algorithm [28] and use the obtained real-valued filter coefficient set as the input $\{h\}$ to our algorithm. Our final coefficient set is given as follows: $h(0) = h(14) = -2^{-9} - 2^{-11}$; $h(1) = h(13) = 0$; $h(2) = h(12) = 2^{-6} + 2^{-10} + 2^{-12}$; $h(3) = h(11) = 0$; $h(4) = h(10) = -2^{-4} - 2^{-8} - 2^{-10} - 2^{-13}$; $h(5) = h(9) = 0$; $h(6) = h(8) = 2^{-2} + 2^{-4} - 2^{-7} - 2^{-9} + 2^{-11} - 2^{-13}$; $h(7) = 2^{-1}$.

The NPRM and the number of CSPT terms are -72.28 dB and 28 for the rounded CSD coefficient set $\{h_{\text{csd}}\}$. The NPRM and the number of CSPT terms reduce to -78.45 dB and 21 after the CSPT coefficient allocation stage. The NPRM reaches -83.63 dB after the quantization compensation stage, and the number of CSPT terms of the final solution is 19. The NPRM per CSPT term from the initial rounded coefficient set to the final solution is progressively reduced from -2.58 dB to -3.74 dB, to -4.40 dB. In fact, the quantization compensation stage takes only two iterations to meet the amplitude response specification.

Table II gives the filter characteristics designed by different algorithms to attain -80 dB stopband attenuation with a minimal possible number of CSPT or SPT terms.

The original coefficient set provided by Kwentus *et al.* [27] aims at only -75 dB NPRM, and its results are used as a reference in the first row of Table II. In Table II, N_{SPT}, N_{CSPT}, N_{101}, and $N_{10\bar{1}}$ denote the number of SPT terms, the number of CSPT terms, and the numbers of common subexpressions 101 and $10\bar{1}$, respectively. We have searched for the common subexpressions 101 and $10\bar{1}$ in the solution sets listed in Kwentus *et al.*'s [27] and Chen and Willson's [5] papers to account for the numbers N_{CSPT}, N_{101}, and $N_{10\bar{1}}$. For Samueli's [20] and Li *et al.*'s [16] algorithms, we are unable to obtain the implementation from their papers, and only the published results of N_{SPT} and NPRM are listed. Based on the NPRM measurement, our algorithm converges to a solution that has a better amplitude response than Kwentus *et al.*'s, Samueli's and Li *et al.*'s solutions. Chen and Willson's trellis search algorithm produces the least number of SPT terms, but the actual implementation cost is reflected in the number of CSPT terms as adders saved by operator sharing have to be accounted. The high N_{SPT} of our algorithm is expected as we encourage operator sharing and eliminate redundant computations by maximizing the number of common subexpressions, which is evincible in the highest aggregate of N_{101} and $N_{10\bar{1}}$. Our solution has the least N_{CSPT} among the algorithms that have the N_{CSPT} listed. We save three CSPT terms over the trellis search algorithm due to the extraction of five more common subexpressions than Chen and Willson's algorithm. In Chen and Willson's solution, the maximum number of SPT terms per coefficient was three; the solution obtained by our algorithm also employs no more than three CSPT terms in any coefficient. The saving in adder cost is achieved at no performance penalty to the logic depth. Table III lists the actual total number of carry-propagate adders (CPAs) that are needed to implement the published coefficient sets that are generated by the three algorithms in the transposed direct form structure. Since this is an even symmetric filter, the first half of the filter taps is duplicated in the second half. The adders that were used to generate the common subexpressions are treated as being equally complex as other adders, although the wordlength of the adders is generally much shorter (a 3-bit adder is sufficient to implement the 101 or $10\bar{1}$ subexpression). As only one subexpression of 101 appears in the solution of the trellis search algorithm, the actual adder cost is the same for its solution with or without the consideration of common subexpressions. With the sharing of common subexpression 101, Kwentus *et al.*'s solution can be realized with one less adder than that of the trellis search algorithm. As there is only one occurrence of the subexpression $10\bar{1}$, our implementation cost is the same as that of Kwentus *et al.*'s but our amplitude response error is much smaller. The amplitude response of Chen and Willson's, Kwentus *et al.*'s, and our solutions are shown in Fig. 5.

TABLE II
Summary of Different Filter Designs for Example 1

Algorithms	NPRM (dB)	N_{SPT}	N_{CSPT}	N_{101}	$N_{10\bar{1}}$
Kwentus [27]	–74.99	31	21	5	0
Samueli [20]	–83.22	28	-	-	-
Li [16]	–82.35	27	-	-	-
Chen [5]	–85.62	23	22	1	0
Proposed	–83.63	31	19	5	1

TABLE III
Comparison of Adder Costs for Example 1

Algorithms	Kwentus [27]	Chen [5]	Proposed
Adder Cost	15	16	15

Example 2: This popular filter specification is taken from [5], [8], [23], and [24]. The normalized passband and stopband edge frequencies are 0.3π and 0.5π, respectively, with equal weighting on the passband and stopband ripples. The coefficient wordlength is 12 bits, and the filter length is 28. The minimum number of SPT terms required by the MILP algorithm [17], Samueli's local search algorithm [20], and Li *et al.*'s [16], Chen and Willson's [5], Yao's [23], Yli-Kaakinen and Saramaki's [24], and Gustafsson and Wanhammar's [7] algorithms to achieve -50 dB normalized peak ripple values is summarized in Table IV. As no original coefficient set of this filter is published in any paper, we use the infinite precision

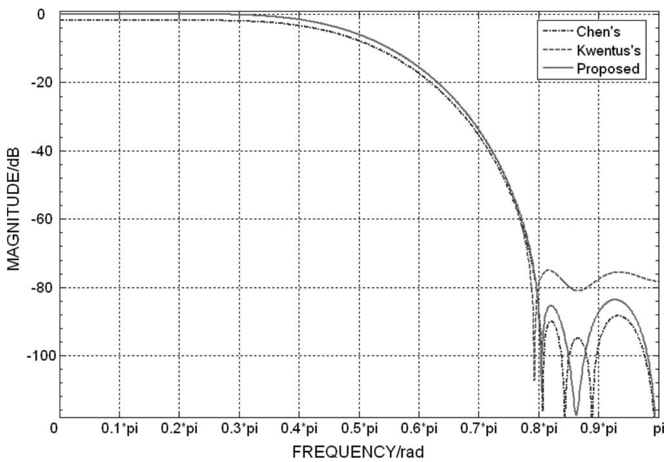

Fig. 5. Comparison of magnitude responses for Example 1.

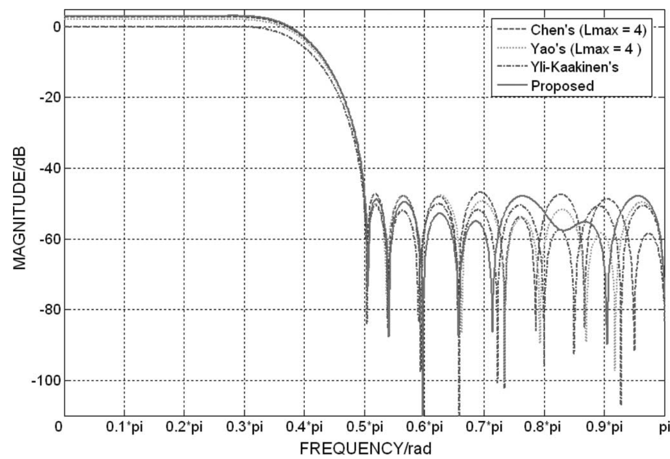

Fig. 6. Comparison of magnitude responses for Example 2.

TABLE IV
SUMMARY OF DIFFERENT FILTER DESIGNS FOR EXAMPLE 2

S/N	Algorithm	N	B	NPRM (dB)	L_{max}	N_{SPT}	N_{CSPT}	N_{101}	$N_{10\bar{1}}$
1	Matlab	28	12	–51.12	-	80	56	5	7
2	MILP [17]	28	12	–50.18	4	68	-	-	-
3	MILP [17]	28	12	–50.18	3	68	-	-	-
4	Li [16]	28	12	–50.35	4	60	-	-	-
5	Gustafsson [7]	28	12	-	-	54	-	-	-
6	Gustafsson [7]	28	12	-	3	56	-	-	-
7	Yli-Kaakinen [24]	28	12	–50.01	-	56	-	-	-
8	Yli-Kaakinen [24]	30	10	–50.71	-	52	42	1	4
9	Yao [23]	28	12	–50.08	4	54	42	2	4
10	Yao [23]	28	12	–50.02	3	56	42	3	4
11	Chen [5]	28	12	–50.17	4	58	42	5	3
12	Chen [5]	28	11	–50.12	3	60	46	4	3
13	Proposed	28	12	–50.05	-	62	40	3	8

coefficient set that was generated using Matlab Simulink based on the Parks–McClellan algorithm [28] as a reference. The final coefficient set generated by our proposed algorithm is given as follows: $h(0) = h(27) = -2^{-8} + 2^{-10} + 2^{-12}$; $h(1) = h(26) = 0$; $h(2) = h(25) = 2^{-7} - 2^{-9} + 2^{-12}$; $h(3) = h(24) = 2^{-7} - 2^{-9} - 2^{-12}$; $h(4) = h(23) = -2^{-7}$; $h(5) = h(22) = -2^{-6} - 2^{-9} + 2^{-11}$; $h(6) = h(21) = 0$; $h(7) = h(20) = 2^{-5}$; $h(8) = h(19) = 2^{-5} - 2^{-7} + 2^{-9} + 2^{-11}$; $h(9) = h(18) = -2^{-5} - 2^{-8} + 2^{-10}$; $h(10) = h(17) = -2^{-4} - 2^{-6} + 2^{-10}$; $h(11) = h(16) = 0$; $h(12) = h(15) = 2^{-2} - 2^{-4} + 2^{-6} - 2^{-8} - 2^{-11}$; $h(13) = h(14) = 2^{-1} - 2^{-3}$.

The desired NPRM has been achieved directly with 56 CSPT terms for the rounded CSD coefficient set $\{h_{\mathrm{csd}}\}$. The number of CSPT terms has been brought down to 44 after the CSPT coefficient allocation stage with a slight increase in NPRM, and the quantization compensation stage further reduces the number of CSPT terms to 40 in 93 iterations before it converges to an NPRM of -50.05 dB. The NPRMs per CSPT term from the initial rounded coefficient set to the final solution are progressively reduced from -0.89 dB to -1.10 dB, to -1.25 dB. The frequency responses of the coefficient sets that were generated by different algorithms, including ours, are shown in Fig. 6.

Table IV gives the comparisons of the NPRM, N_{SPT}, N_{CSPT}, N_{101}, and $N_{10\bar{1}}$ achieved by the solutions of various algorithms. $L_{\max}$ is the maximum allowable number of SPT terms per coefficient. It should be noted that we have also included two published solutions in Table IV that use different numbers of taps N and wordlengths B for the same amplitude response. The results used for the comparison are taken from [5] and [8]. For those algorithms whose final coefficient sets were not given in the papers, the unavailable information is marked with "–" in Table IV. The results show that our proposed algorithm produces a solution with the least CSPT terms. Altogether, 11 common subexpressions of 101 and $10\bar{1}$ have been generated by our algorithm, which is the second highest number. The original decimal coefficient set that was generated using Matlab has the most common subexpressions, but it has also a high number of nonsharable SPT terms. Our algorithm achieves the least number of CSPT terms at the expense of a comparatively higher number of SPT terms due to our physical adder cost oriented objective of optimization.

TABLE V
AREA COMPARISON OF SYNTHESIZED CIRCUITS

Filters	Proposed	Filter 1	Filter 8	Filter 9	Filter 10	Filter 11	Filter 12
Area(μm^2)	65216	93855	66453	67894	69018	71803	78243
%Saving	-	43.92	1.90	4.11	5.83	10.10	19.98

To have an insight into the magnitude of gate area improvement translated from the saving in CSPT terms, we have written structural VHDL codes to implement filter numbers 1 and 8–13 of Table IV using 12-bit input data. These filters are selected to compare the solutions with different numbers of CSPT terms and coefficient wordlengths. The RTL designs were functionally verified with Mentor Graphic ModelSim before they are synthesized into gate level netlist using Synopsys Design Compiler version 2004.06. The synthesis was performed with a standard application-specified integrated circuit design flow using the TSMC 0.18-μm standard cell library. The circuits were synthesized with a consistent timing constraint of 10 ns. The prelayout combinational circuit areas were given in Table V.

The synthesis results showed that our proposed algorithm for maximizing the sharing of common subexpressions 101 and $10\bar{1}$ led to a 2%–44% reduction in the combinational area. The savings in the combinational area is attributable to the reduction in the number of adders used, as well as the lengths of the adders. For the same number of nonzero digits in a coefficient, extraction of the two shortest common subexpressions tends to produce shorter length adders.

If the coefficient multipliers in each tap are implemented with CSAs, the number of CPAs can be reduced to only one. The performance of the filter is improved, particularly when the partial sums in the critical tap is high at the expense of using two sets of registers, as shown in Fig. 7. Doubling the number of registers increases the noncombinational circuit areas and complicates the clock distribution network. Common subexpression sharing can also be integrated into the CSA trees. The sum and carry outputs of the products of input and 101, and input and $10\bar{1}$, can be precomputed.

In [5], Chen and Willson proposed a merge–search method to reduce the number of CSAs for a CSA-tree-based transposed direct form

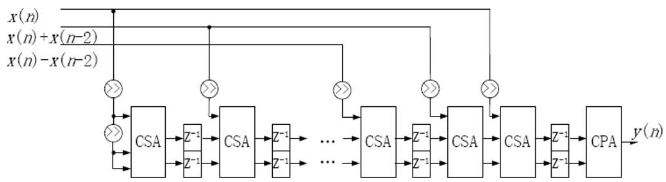

Fig. 7. CSA implementation of a transposed direct form filter.

TABLE VI
COMPARISON WITH MERGE–SEARCH ALGORITHMS FOR CSA-BASED FILTER

Algorithms	CSA depth limit per coefficient	N	N_{CSA}
Trellis-merge	≤ 3	28	40
	≤ 2	28	40
MILP-merge	≤ 3	28	49
	≤ 2	29	52
Proposed	-	28	38

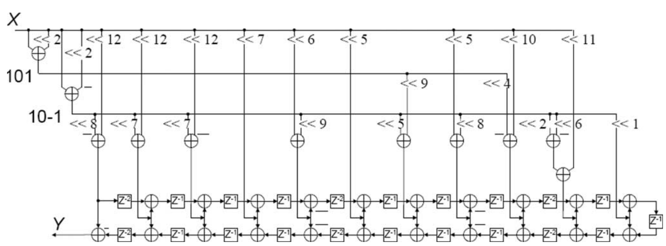

Fig. 8. Symmetric filter with common subexpressions 101 and $10\overline{1}$.

FIR filter architecture. An additional precomputation cycle involving two fast 3-bit CPAs for the common subexpressions 101 and $10\overline{1}$ is incurred in this architecture. A constraint on $L_{\max}$ is imposed to limit the maximum number of CSAs for a high-throughput rate. The solutions of merging 101 and $10\overline{1}$ within the CSD coefficients before the optimization by MILP, which is called the MILP–merge, are also provided in [5]. We have implemented our solution based on the transposed direct form architecture with CSAs, and the results are tabulated in Table VI, along with the trellis–merge and MILP–merge algorithms from [5] for comparison.

Although we do not restrict the number of CSPT terms per coefficient, the solution generated by our algorithm has no more than five SPT terms and three CSPT terms in any coefficient. Therefore, no more than two CSAs are needed for every coefficient. Without degrading the throughput rate, our solution requires a slightly smaller number of CSAs.

It should be noted, however, that the CSA-based transposed direct form architecture cannot benefit from any duplicated coefficient as there are at least two CSAs needed per tap to accumulate the stored sum and carry from the previous tap to those of the current tap, in addition to doubling the number of registers needed to store the sum and carry signals. If the conventional CPAs are used for the transposed direct form filter, the saving in the adder cost is prominent for a symmetric filter, as half of the coefficients are duplicated and incur no additional adder cost. The implementation of our solution in the symmetrical transposed direct form filter structure is shown in Fig. 8. Two 3-bit adders are used for the generation of subexpressions 101 and $10\overline{1}$, 11 adders are used to generate the coefficients, and 21 adders are used in the accumulation line.

The solution of our proposed algorithm is further compared with the solution of Yli-Kaakinen and Saramaki's algorithm [24], which is one that involves sharing of higher weight common subexpressions. Yli-Kaakinen and Saramaki have provided two circuits with higher weight common subexpressions for this filter specification using different numbers of taps and wordlengths. The best implementation is obtained with $L = 30$ and $B = 10$. We also extracted the higher weight common subexpressions from our solution, and the final circuit implementation of our design is shown in Fig. 9. Two weight-3 common subexpressions have been identified, which are $10\overline{1}00\overline{1}$ and $\overline{1}00\overline{1}01$. Since both weight-3 common subexpressions appear twice, two more adders are saved. This solution needs only 30 adders, which is the same as Yli-Kaakinen and Saramaki's most optimal solution.

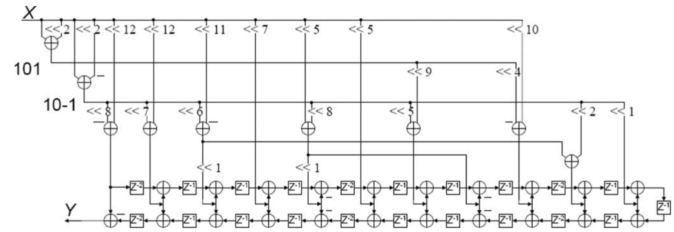

Fig. 9. Symmetric filter with higher weight common subexpressions.

TABLE VII
EXPERIMENTAL RESULTS FOR OTHER FILTER SPECIFICATIONS

	ω_p	ω_s	$\frac{\delta_p}{\delta_s}$	N	B	Desired NPRM	NPRM achieved	$NCSPT$	$\frac{NCSPT}{N \times B}$
FIR1	0.3π	0.5π	1	25	12	-46 dB	-46.16 dB	39	0.13
FIR2	0.3π	0.5π	3	35	11	-50 dB	-50.06 dB	44	0.11
FIR3	0.26π	0.41π	32/23	28	10	-40 dB	-40.18 dB	44	0.16
FIR4	0.3π	0.44π	1	53	9	-40 dB	-40.27 dB	46	0.10
FIR5	0.6π	0.69π	1	64	13	-50 dB	-50.08 dB	98	0.12
FIR6	0.47π	0.5π	1	131	12	-40 dB	-40.00 dB	195	0.12

To show that our algorithm is also capable of synthesizing desirable solutions to a wide variety of FIR filter response specifications, we also evaluated our proposed algorithm using a number of other frequently used specifications for coefficient synthesis algorithms. The results are shown in Table VII.

In Table VII, ω_p and ω_s are the normalized passband and stopband edge frequencies, respectively, δ_p and δ_s are the passband and stopband ripple magnitudes, respectively, N is the filter length, and B is the wordlength of the coefficients. The last column of Table VI provides the ratios of NCSPT to the product of filter length and coefficient wordlength. This normalized number of CSPT terms serves as a useful figure-of-merit to evaluate the overall performance of the algorithm in terms of the average adder cost per tap per bit. As mentioned before, the maximum number of nonzero digits for an n-bit binary number that is expressed in CSD format is bounded by $(n + 1)/2$ and the expected number of nonzero digits approaches $n/3 + 1/9$ asymptotically [9]. The expected number of nonzero digits varies from one filter to another, and the nonzero digits of the FIR filter are not uniformly distributed among different CSD coefficients. The figure-of-merit shows that, on the average, for every n-bit binary coefficient, our algorithm can reduce the number of CSPT terms to less than $n/8$. This is a noteworthy reduction. The NCSPT is highly correlated to the actual adder cost incurred than the number of SPT terms, with an overhead of only two (and sometimes one) adders to generate the two common subexpressions 101 and $10\overline{1}$.

V. CONCLUSION

We have proposed an efficient algorithm for the design of low-complexity FIR filter by maximizing the sharing of adders in the synthesis of filter coefficients to meet the desired magnitude response specification. The proposed algorithm considers the reuse of the two most frequently used subexpressions 101 and $10\overline{1}$. By dividing the multiple-constraint problem of minimizing the CSPT terms into two stages, the problem is tackled heuristically in an efficient and elegant approach. With the help of CS-HWP, a minimal CSD coefficient set with the minimal NPRM in the continuum of the rounded CSD coefficient set is found in the first stage. Depending on the wordlength of the coefficients, and the mismatch between the desired response and that obtained in the first stage, bivariate quantization steps are iteratively added in the second stage to compensate for the quantization errors until the desired NPRM is met. We have shown, by means of benchmark design examples, that our proposed algorithm is capable of synthesizing transposed direct form FIR filters with the least CSPT terms in both the CSA-tree-based and conventional adder-based architectures among the existing filter synthesis algorithms. As it is a local search algorithm and the search range is well defined, the proposed algorithm finds a set of filter coefficients that meets the filter specification in a reasonable time.

REFERENCES

[1] D. Ait-Boudaoud and R. Cemes, "Modified sensitivity criterion for the design of powers-of-two FIR filters," *Electron. Lett.*, vol. 29, no. 16, pp. 1467–1469, Aug. 1993.

[2] N. Benvenuto, M. Marchesi, and A. Uncini, "Applications of simulated annealing for the design of special digital filters," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 323–332, Feb. 1992.

[3] T. W. Fox and L. E. Turner, "The design of peak-constrained least square FIR filters with low-complexity finite-precision coefficient," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 2, pp. 151–154, Feb. 2002.

[4] R. Cemes and D. Ait-Boudaoud, "Genetic approach to design of multiplierless FIR filters," *Electron. Lett.*, vol. 29, no. 24, pp. 2090–2091, Nov. 1993.

[5] C. L. Chen and A. N. Willson, Jr., "A trellis search algorithm for the design of FIR filters with signed-powers-of-two coefficients," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 29–39, Jan. 1999.

[6] H. Choo, K. Muhammad, and K. Roy, "Complexity reduction of digital filters using shift inclusive differential coefficients," *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1760–1772, Jun. 2004.

[7] O. Gustafsson and L. Wanhammar, "Design of linear-phase FIR filters combining subexpression sharing with MILP," in *Proc. IEEE Midwest Symp. Circuits and Syst.*, Tulsa, OK, Aug. 2002, vol. 3, pp. 9–12.

[8] O. Gustafsson, "Contributions to low-complexity digital filters," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 2003.

[9] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 677–688, Oct. 1996.

[10] 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.

[11] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution algorithms for common subexpression elimination in digital filter design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 10, pp. 695–700, Oct. 2005.

[12] F. Xu, C. H. Chang, and C. C. Jong, "A new integrated approach to the design of low-complexity FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kos, Greece, May 2006, pp. 601–604.

[13] C. Y. Yao, H. H. Chen, T. F. Lin, C. J. Chien, and C. T. Hsu, "A novel common-subexpression-elimination method for synthesizing fixed-point FIR filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 11, pp. 2215–2221, Nov. 2004.

[14] D. Kodek and K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital filters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. CAS2-28, no. 1, pp. 28–32, Jan. 1981.

[15] J. H. Lee, C. K. Chen, and Y. C. Lim, "Design of discrete coefficient FIR digital filters with arbitrary amplitude and phase responses," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 40, no. 7, pp. 444–448, Jul. 1993.

[16] D. Li, J. Song, and Y. C. Lim, "A polynomial-time algorithm for designing digital filters with power-of-two coefficients," in *Proc. IEEE Int. Symp. Circuits Syst.*, Chicago, IL, May 1993, vol. 1, pp. 84–87.

[17] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete power-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 3, pp. 583–591, Jun. 1983.

[18] Y. C. Lim, "Design of discrete-coefficient-value linear phase FIR filters with optimum normalized peak ripple magnitude," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 37, no. 12, pp. 1480–1486, Dec. 1990.

[19] V. S. Rosa, E. Costa, J. C. Monteiro, and S. Bampi, "An improved synthesis method for low power hardwired FIR filters," in *Proc. 17th Int. Symp. Integr. Circuits and Syst. Des.*, 2004, pp. 237–241.

[20] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 36, no. 7, pp. 1044–1047, Jul. 1989.

[21] H. Samueli, "The design of multiplierless digital data transmission filters with powers-of-two coefficients," in *Proc. IEEE Int. Telecommun. Symp.*, Rio de Janeiro, Brazil, Sep. 1990, pp. 425–429.

[22] H. Shaffeu, M. M. Jones, H. D. Griffiths, and J. T. Taylor, "Improved design procedure for multiplierless FIR digital filters," *Electron. Lett.*, vol. 27, no. 13, pp. 1142–1144, Jun. 1991.

[23] C. Y. Yao, "A study of SPT-term distribution of CSD numbers and its application for designing fixed-point linear phase FIR," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2001, vol. 2, pp. 301–304.

[24] J. Yli-Kaakinen and T. Saramaki, "A systematic algorithm for the design of multiplierless FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, Geneva, Switzerland, 2001, vol. 2, pp. 185–188.

[25] K. Hwang, *Computer Arithmetic*. New York: Wiley, 1979.

[26] F. Xu, C. H. Chang, and C. C. Jong, "Hamming weight pyramid—A new insight into canonical signed digit representation and its applications," *J. Comput. Elect. Eng.*, vol. 33, no. 3, pp. 195–207, May 2007.

[27] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 457–467, Feb. 1997.

[28] *Programs for Digital Signal Processing*. New York: IEEE Press, 1979. Algorithm 5.1.