

# Information theoretic approach to complexity reduction of FIR filter design

Chang, Chip Hong; Chen, Jiajia; Vinod, Achutavarrier Prasad

2008

Chang, C. H., Chen, J., & Vinod, A. P. (2008). Information Theoretic Approach to Complexity Reduction of FIR Filter Design. *IEEE Transactions on Circuits and Systems—I*. 55(8), 2310-2321.

<https://hdl.handle.net/10356/93107>

<https://doi.org/10.1109/TCSI.2008.920090>

---

© 2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder. <http://www.ieee.org/portal/site> This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

# Information Theoretic Approach to Complexity Reduction of FIR Filter Design

Chip-Hong Chang, *Senior Member, IEEE*, Jiajia Chen, and A. P. Vinod, *Senior Member, IEEE*

**Abstract**—This paper presents a new paradigm of design methodology to reduce the complexity of application-specific finite-impulse response (FIR) digital filters. A new adder graph data structure called the multiroot binary partition graph (MBPG) is proposed for the formulation of the multiple constant multiplication problem of FIR filter design. The set of coefficients in any fixed point representation is partitioned into symbols so that common subexpression identification and elimination become congruent to information parsing for data compression. A minimum number of different pairs or groups of symbols and residues can be used to code a set of coefficients based on their probability and conditional probability of occurrence. This ingenious concept enables the notion of entropy to be applied as a quantitative measure to evaluate the coding density of different compositions of symbols towards a set of coefficients. The minimal vertex set MBPG synthesized by our proposed information theoretic approach results in direct correspondences between the vertices and adders, and edges and physical interconnections. Unlike the common subexpression elimination algorithms based on other graph data structures, the symbol-level information carried in each vertex and the graph isomorphism of MBPG promise further fine-grain optimization in a reduced search space. One such optimization that has been exploited in this paper is the shift-inclusive computation reordering to minimize the width of every two's complement adder to further reduce the implementation cost and the critical path delay of the filter. Experiment results show that the proposed algorithm can contribute up to 19.30% reductions in logic complexity and up to 61.03% reduction in critical path delay over other minimization methods.

**Index Terms**—Common subexpression elimination (CSE), finite-impulse response (FIR) digital filter, graph, information theory, multiple constant multiplication.

## I. INTRODUCTION

THE last decade has witnessed a perpetual growth of custom design integrated circuits for digital signal processing (DSP) blocks. Technological advancement on analog-to-digital converters and all-digital modem for broadband direct conversion wireless receiver has in particular, imposed extreme demands upon the DSP chain of digital radio [1]–[3]. Digital filtering at such high data rate on battery-powered device can not be substantiated by general purpose signal processors without ancillary hardware accelerators. Therefore,

the development of powerful computer-aided design tool for high-speed, low complexity and low power digital filters is an alluring goal. For dedicated applications, the programmability of a multiplier is not mandatory and efficient fixed coefficient finite-impulse response (FIR) filters can be realized in fully parallel architectures. The design turnaround time for application-specific FIR filters can be greatly reduced by algorithms that apply computational transformations to increase the performance with reduced implementation cost.

One widely adopted design methodology for reducing the complexity of fixed coefficient FIR filters is to replace the expensive and power-hungry multipliers by simpler arithmetic adders and shifters. The number of parallel adders required can still be reduced by further exploiting the computational redundancy present in the multiple constant multiplications (MCM) [4]. Recent approaches [5]–[11] to the design of low complexity and low power FIR filters focus on the common subexpression elimination (CSE) of MCM in transposed direct form structure. These algorithms primarily target at the reduction of the number of adders and the depth of the adder tree that implement the multiplier block. They use either exhaustive search, steepest descent heuristic or quasi-exhaustive search to eliminate common subexpressions from a given coefficient set. The approaches can be broadly summarized in two categories—value based and pattern based eliminations.

Valued based algorithms assume no specific numeral format for the detection and elimination of redundancy. A number of Graph-dependence (GD) algorithms [12]–[16], founded on the classical primitive operator graph [14], are representatives of this class. GD algorithms compose the decimal coefficient values from unity through a number of primitive arithmetic operations. Inner products of the input sample are encapsulated in the nodes and shifts are annotated on the edges of the directed acyclic graph synthesized by a typical GD algorithm. The most optimal primitive graph is generated by an exhaustive search of all possible graph topologies. The consequence of this is a very high computation complexity, and the wordlength of the filter coefficients is limited by the size of the look-up table. GD algorithms are inherently order-dependent because the coefficients are composed sequentially and the partial sums are formed from existing partial sums and the processed coefficients. Thus, the multiplier block optimized by conventional GD algorithms are likely to yield long critical path. Although some heuristic has been applied to GD algorithms to reduce computational complexity, the effort has been devoted exclusively to optimization of isolated constant multiplication. The problem of representation fluidity is the correlation between coefficients has been weakly exploited as the statistic of any composition

Manuscript received March 19, 2007; revised November 12, 2007. First published March 7, 2008; current version published September 17, 2008. This paper was recommended by Associate Editor R. Merched.

The authors are with the Nanyang Technological University, Singapore (e-mail: echchang@ntu.edu.sg; chen0183@ntu.edu.sg; asvinod@ntu.edu.sg).

Digital Object Identifier 10.1109/TCSI.2008.920090

of partial sums and how it influences later compositions are not known in advance.

Pattern matching is a technique that has historically been adopted in computer-aided design for high-level synthesis [17]. It has become a popular approach to the design of VLSI efficient FIR filters in recent years [5], [7], [9], [10]. The gist of pattern based CSE lies in the detection of patterns and pattern correlation in a set of filter coefficients to reuse the arithmetic operators. Canonical signed digit (CSD) representation is widely used to provide such a ‘global’ outlook of the potentially sharable subexpressions. The primary advantage of CSD code over the normal binary representation is the added flexibility of negative digits allows most coefficients to be represented with fewer nonzero digits [18]. Using integer linear programming (ILP) model, optimal subexpression sharing algorithms based on CSD coefficients have been developed [19], [20]. These algorithms have high computational complexity and are not suitable for large problem. Since the search for optimal common subexpressions is an NP-complete problem, most algorithms are heuristic in nature. To extricate from nonproductive exhaustive search, some algorithms reuse only specific types of weight-2 subexpressions [8], [9], [19], [21], [22]. The search for the maximally sharable common subexpressions is guided by static subexpression distribution statistics from the initial coefficient set. The CSE algorithms in [8], [9], [22] use weight-2 subexpressions as the primitive elements for reuse and then search for higher-weight common subexpressions from existing lower-weight common subexpressions. On the contrary, some algorithms [5], [7] search for the highest-weight common subexpressions at the outset followed by splitting them into lower-weight common subexpressions for further sharing. This is to insure that the higher-weight common subexpressions do not always give in to the lower-weight subexpressions when they are overlapped. Some algorithms emphasize on the reuse of fewer different common subexpressions with high frequency of reuse while others search for many different types of common subexpressions at the expense of the frequency of reuse of some conflicting common subexpressions, but the decisions are normally made based on experiential rather than theoretical ground.

A scrutiny of the current CSE algorithms shows that some dilemma exists due to the relatively simple exploitation of the statistical distribution of different subexpressions. Owing to the interdependencies among different subexpressions within and across the coefficients, a downright exploitation of the frequencies of common subexpressions is inadequate. This subtle pattern correlation has not been formally explored and optimized with graph theoretic property. From information theory, the notion of entropy can exploit the subexpression statistics to appraise the opportunity cost of interdependent subexpressions and optimize their net gain effectively. In this paper, we propose a new formulation of CSE for the design of multiplierless FIR filters. Our formulation has led to several distinctive advantages. First, an intuitive data structure called the Multirooted Binary Partition Graph (MBPG) is evolved which has a direct correlation to the circuit topology. Under this data structure, common subexpressions and graph isomorphism are congruence. Isomorphism is an inherent attribute of minimal MBPG

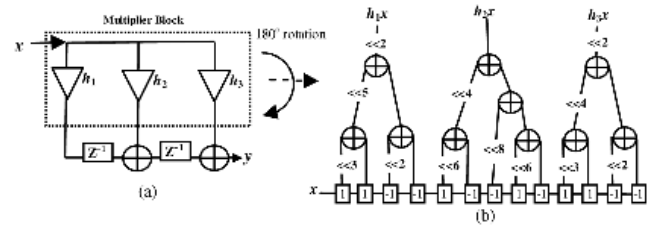


Fig. 1. Adder tree decomposition of transposed direct form FIR filter.

and is generated by construction. Second, information theoretic approach has been widely used in data compression, pattern recognition and classification [23]–[25] but never has the elegance of information theory been so closely coupled with graph to tackle the complexity reduction of FIR filters. The concept of entropy and conditional entropy has been applied for the first time in this research to maximize common subexpression sharings by set partitioning. The synthesis of a minimal MBPG provides a meaningful insight into judicious resource sharings in MCM problem based on the rigor of proven probabilistic measures. Finally, as the widths of operands and the number of shifts are annotated on the vertices and edges of the resultant MBPG, adder complexity due to different widths and relative shifts of operands can be readily evaluated by graph traversal. Fine grain reduction of logic complexity and logic depth can be made by reordering the vertices without perturbing the size of MBPG.

This paper is organized in six sections. In Section II, we formulate CSE as a set partitioning problem and introduce the notion of MBPG. Section III presents our proposed information theoretic approach to the minimization of MBPG. A strategy to further reduce the implementation cost and critical path delay in terms of full adders is proposed in Section IV. Section V presents the experimental results on commonly used benchmark filters and practical channelizers of wideband radio receivers. Our conclusion is provided in Section VI.

## II. MBPG PROBLEM FORMULATION

An  $N$ -tap FIR digital filter is a linear time invariant system governed by a linear convolution of a discrete-time process variable  $x(t)$  and a set of  $N$  finite-valued coefficients

$$y(t) = \sum_{i=0}^{N-1} h_i x(t - i) \tag{1}$$

where  $h_i$  is the  $i$ th coefficient, and  $x(t)$  and  $y(t)$  are the input and output data sampled at time  $t$ . (1) lends itself nicely into a fast transposed direct form realization whereby the critical path is limited to only a single multiply-and-add operation irrespective of the filter length,  $N$ . By input and output normalization/denormalization, the coefficients can be scaled to a set of constant integers for fixed point implementation.

Fig. 1(a) shows the architecture of a 3-tap transposed direct form FIR filter of a scaled integer coefficient set,  $H = \{1132, 815, 556\}$ . The multiplier block can be deemed as a single-input, multiple-output combinational function. If we rotate the multiplier block by  $180^\circ$ , and remap each constant multiplier into a network of hard-wired shifters and adders, a forest of binary trees can be derived where the roots are the

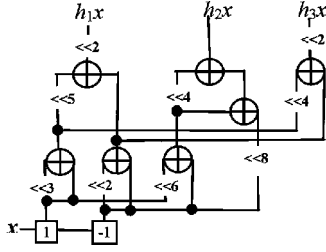


Fig. 2. MBPG.

coefficients, the intermediate nodes are adders of partial sums and the leaves are fundamental literals that define the number system for the coefficient representation. Since two's complement adders are generally available in high-level synthesis of digital system, it is natural to assume a minimal signed digit representation of the coefficients to reduce the number of dual polarity arithmetic operators of the tree. Fig. 1(b) shows the collapsing of the canonical signed digit (CSD) coefficients into a binary tree. By definition of a tree, there is one and only one path from the root to every intermediate vertex. The operators can be further reduced by merging isomorphic vertices, which leads to a directed-acyclic graph (DAG) shown in Fig. 2. This data structure is the backbone of our proposed information-theoretic approach. We called it the multiroot binary partitioning graph (MBPG).

An MBPG is a DAG  $G = \{V, E\}$  where  $V$  represents a set of resources (typically two's complement adders) and  $E$  is a set of interconnects. Two types of vertices are defined in  $V$ . A nonterminal vertex  $v$  has as attributes two children vertices,  $l(v)$  and  $r(v)$  and an output value,  $c(v)$ . The terminal vertices are vertices with no children and they represent the set of basic literals of the number system used to represent the coefficients. An edge,  $e \in E$  is a connection from a vertex  $v_1$  to a vertex  $v_2$ . It has a value,  $e_{v_1}$  that indicates the amount of shift to be applied to  $c(v_1)$  before feeding to  $v_2$ .

Every vertex except the roots has at least one parent. The roots of  $G$  have no parent and their output values are equal to the coefficients,  $h_i$ . The children vertices are related to the parent vertex,  $v$  as follows:

$$c(v) = 2^{e_{l(v)}}c(l(v)) + 2^{e_{r(v)}}c(r(v)) \quad (2)$$

where  $e_{l(v)}$  and  $e_{r(v)}$  are the edge values emanated from  $v$  and directed to  $l(v)$  and  $r(v)$ , respectively.

In radix-2 signed digit representation, three constant terminal vertices, 0, 1 and  $-1$  are defined. The constant coefficients are usually provided in a fractional form with  $l$  fractional digits such that  $0 \leq |h_i| < 1$ . By factoring out the unit in least significant position ( $ulp = 2^{-l}$ ), a fixed point representation of  $B$  whole digits can be obtained

$$c(v) = \sum_{i=1}^{L_{c(v)}} s_i 2^{-p_i} = ulp \sum_{i=1}^{L_{c(v)}} s_i 2^{b_i-1} \quad (3)$$

where  $s_i \in \{-1, 1\}$  is the sign of the  $i$ th nonzero term at bit position,  $p_i \in \{1, 2, \dots, l\}$  and  $L_{c(v)}$  is the number of nonzero digits of  $c(v)$ . Since fractional constant can be obtained by a shift operation at the output  $c(v)$  of any nonterminal vertex  $v \in V$  can now be succinctly represented by a set of signed digits  $\{s_i b_i\}$  where  $b_i \in \{1, 2, \dots, B\}$  signifying the signs and positions of the scaled signed-power-of-two (SPT) terms.

Using this simplified set representation of  $c(v)$ , from (3), the following set partitioning relations are observed when the coefficient set represented by the roots in MBPG is parsed

$$\begin{aligned} c(l(v)) + e_{l(v)}, c(r(v)) + e_{r(v)} &\in c(v), \\ \{c(l(v)) + e_{l(v)}\} \cup \{c(r(v)) + e_{r(v)}\} &= c(v) \\ \{c(l(v)) + e_{l(v)}\} \cup \{c(r(v)) + e_{r(v)}\} &= \emptyset. \end{aligned} \quad (4)$$

By recursively partitioning the output value of each vertex into two disjoint subsets of SPT terms, a complete binary tree can be formed and it can be reduced to a MBPG by sharing isomorphic vertices. Two nonterminal vertices,  $u$  and  $v$  are said to be isomorphic if  $c(v) = c(u)$ . Isomorphic vertices have indegree  $> 1$  and they represent common subexpressions. Therefore, the CSE problem in the design of multiplierless digital filter can be recast as a problem of synthesizing a minimal vertex set MBPG by set partitioning. Common subexpressions are generated by construction since isomorphism is an inherent graph theoretic property of MBPG. Unlike other graph dependence approaches and CSE methods, the adder topologies are explicitly implied by MBPG and the bit-level information of each operand is readily accessible, making it amenable to fine-grain optimization at operator level and/or further tradeoff in logic operator and logic depth by computation reordering.

### III. INFORMATION THEORETIC APPROACH TO MBPG MINIMIZATION

The basis of CSE in the context of MBPG is the creation of isomorphism. This section presents a new paradigm of graph minimization method substantiated by the veracity of information theory [24].

If we consider the communication between two vertices of MBPG as an information parsing process, then the messages being parsed are symbols in the set  $\{s_i b_i\}$  of the source vertex,  $v_i$ . Since an event with absolute certainty to occur conveys no information, the information content of a source is connected with the reciprocal of expectancy. To maximize the number of occurrences of an event of interest, it is important to quantify the information content of a source in message parsing.

*Definition 1:* Let  $X$  be a memoryless source of  $n$  elementary messages,  $X_i, i = 1, 2, \dots, n$ , each with a probability of occurrence of  $p(X_i)$  in an event and  $\sum_{i=1}^n p(X_i) = 1$ . The entropy  $H(X)$  is defined as [25]

$$H(X) = - \sum_{i=1}^n p(X_i) \log_2 p(X_i) \quad (5)$$

where  $p(X_i) \log_2 p(X_i)$  is taken to be 0 if  $p(X_i) = 0$ .

$H(X) = 0$  if  $p(X_i) = 1$  for only one event,  $X_i$ , and  $p(X_j) = 0$  for all other events,  $X_j \neq X_i$ . This is an extreme situation where the outcome of an event can be predicted with complete certainty. Maximum entropy occurs if each event  $X_i$  is equally likely, i.e.,  $p(X_i) = 1/n \forall i = 1, 2, \dots, n$ . Entropy is a measure of the amount of uncertainty about  $X$  that is resolved when  $X$  is observed. In this case, an elementary message is a symbol or a set of symbols associated with a vertex and the event of interest is the formation of isomorphic subgraphs in MBPG. The entropy of a set of coefficients is then related to the minimal average number of pairs of symbols needed to represent it and hence the maximal amount of adder redundancy. A memoryless source implies that each message selected is independent of the previous message. The following definition is useful for evaluating the information content for dependent message parsing.

*Definition 2:* Let  $X$  and  $Y$  be two sources of elementary messages,  $X_i$  and  $Y_j$  with their probability distributions given by  $p(X_j)$  and  $p(Y_j)$ , respectively, for  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, m$ .  $X_i$  and  $Y_j$  may be dependent.  $p(Y_j|X_i)$  is the conditional probability of the occurrence of  $Y_j$  in an event provided that  $X_i$  has occurred. The conditional entropy  $H(Y|X_j)$  of source  $Y$ , provided that  $X_j$  has occurred in  $X$ , is given by [25]

$$H(Y|X_i) = -\sum_{j=1}^m p(Y_j|X_i) \log_2 p(Y_j|X_i). \quad (6)$$

The probability of existence of isomorphic subgraphs in MBPG is determined from the signed digits of the subexpressions. Two patterns of signed expressions can be composed from any two nonzero digits,  $s_i b_i, s_j b_j \in c(v)$ . A positive subexpression is one that has  $d = |b_i - b_j| - 1$  zeros between two nonzero digits,  $b_i$  and  $b_j$ , and  $s_i \times s_j$  is positive. A negative subexpression is similarly defined except that  $s_i \times s_j$  is negative. We use the PT array of [8] to keep track of the number of occurrences of the positive and negative subexpressions for the estimation of probability and conditional probability for (5) and (6).

*Definition 3:* A PT array is a  $2 \times (B - 2)$  dimensional array with the entry in the upper (lower) row and the  $j$ th column represent the number of occurrences of positive (negative) subexpressions with  $j$  zeros.

*Example 1:* The PT array of the CSD coefficients of Fig. 1 is given by

$$\text{PT} = \begin{bmatrix} 2 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 1 & 1 & 2 & 3 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

$\text{PT}[s][d]$  is the number of occurrences of positive subexpressions with  $d$  zeros if  $s = 1$  and negative subexpressions if  $s = -1$ . For instance, the bottom row of the fifth column of PT, denoted by  $\text{PT}[-1][5] = 3$  since there are three occurrences of the negative subexpressions,  $100000\bar{1}$  and  $\bar{1}000001$ . It should be noted that during the construction of MBPG, the PT array is dynamically updated to ensure that the entropy and conditional entropy are calculated based on the latest statistics as the messages carried by the source are parsed.

The following probability computations from the PT array are used to evaluate the entropy and conditional entropy in Definitions 1 and 2.

Let  $s_i b_i, s_j b_j \in c(v)$ , the probability that  $s_i b_i$  and  $s_j b_j$  form a subexpression is given by

$$p(s_i b_i, s_j b_j) = \frac{\text{PT}[s_i s_j][|b_i - b_j| - 1]}{\sum_{j \neq i} \text{PT}[s_i s_j][|b_i - b_j| - 1]}. \quad (7)$$

Let  $m = |c(l(v))|$  and  $n = |c(r(v))|$ . The conditional probability that a signed digit,  $s_j b_j \in c(l(v))$  forms a subexpression with  $s_i b_i \in c(v)$  given that  $s_i b_i$  is assigned to  $l(v)$  can be computed by

$$\begin{aligned} p(s_j b_j | s_i b_i \rightarrow c(l(v))) &= \frac{\text{PT}[s_i s_j][|b_i - b_j| - 1]}{\sum_{k=1}^m \text{PT}[s_i s_k][|b_i - b_k| - 1] + \sum_{l=1}^n \text{PT}[s_i s_l][|b_i - b_l| - 1]} \end{aligned} \quad (8)$$

where  $s_k b_k \in c(l(v))$  and  $s_l b_l \in c(r(v))$ . The conditional probability of a signed digit in the right child of  $v$  forming a common subexpression with a designated signed digit,  $s_i b_i \in c(v)$  if  $s_i b_i$  is assigned to  $r(v)$  can be similarly defined.

To construct a minimal MBPG, the roots are first generated in  $G$ . The value of each root is the set of {sign, position} of the SPT terms of the CSD coefficient. If the value of a root is equal to 1,  $-1$  or 0, it is assigned to a constant terminal vertex 1,  $-1$  or 0 accordingly. The nonterminal vertices generated are inserted into a list,  $Q$ . For each nonterminal vertex  $v \in Q$  two information theoretic decisions will be made. First, when  $c(l(v)) = c(r(v)) = \emptyset$ , a pair of values  $s_i b_i$  and  $s_j b_j \in c(v)$  will be selected based on the entropy measure to give birth to  $l(v)$  and  $r(v)$  in  $G$ . Then, the remaining signed digits of  $c(v)$  are successively allocated to either  $c(l(v))$  or  $c(r(v))$  depending on their conditional entropies. The following propositions are suggested for these operations.

*Proposition 1:* The nonterminal vertex,  $u$  with the least entropy,  $u = \arg(\min_{v \in G} H(c(v)))$  will be decomposed first since the vertex of minimum entropy indicates the maximum likelihood that it subsumes isomorphic subgraphs. Since the PT array changed dynamically in the process of decomposition, it is preferable to generate the most potential isomorphic subgraph as early as possible to prevent their formation from being annihilated.

*Proposition 2:* For any  $s_i b_i \in c(v)$ , the lower the entropy  $H(s_i b_i)$ , the higher the confidence that  $s_i b_i$  is contained in an isomorphic subgraph subsumed by  $v$ . Therefore, the two signed digits that give the least entropy shall be split into  $c(l(v))$  and  $c(r(v))$ , provided that the probability of these two signed digits coexist in the same isomorphic subgraph in  $G$  is sufficiently low.

*Proposition 3:* Let  $s_i b_i \in c(v)$ , then  $s_i b_i$  will be assigned to  $l(v)$  if  $H(c(l(v))|s_i b_i \rightarrow c(l(v))) < H(c(r(v))|s_i b_i \rightarrow c(r(v)))$  and vice-versa. This is because the lower conditional entropy between two assignments of  $s_i b_i$  implies a greater certainty that the assignment will lead to isomorphic subgraphs whose SPT terms subsume the signed digit  $s_i b_i$ .

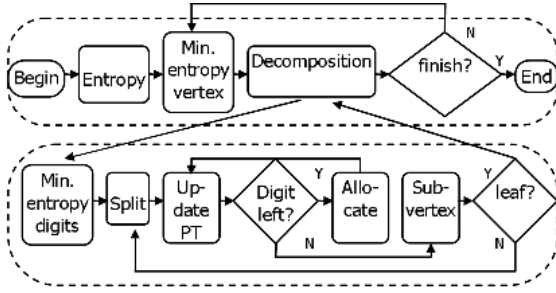


Fig. 3. Flowchart of the proposed information theoretic method.

```

decompose(v) {
  for (each  $s_i b_i$  in  $c(v)$ ) calculate  $\text{entropy}(s_i b_i)$ ;
   $D$  = signed digits of  $c(v)$  sorted in ascending order of entropy;
   $s_1 b_1$  = first entry of  $D$ ;  $s_2 b_2$  = second entry of  $D$ ;
  while ( $\text{entropy}(\{s_1 b_1, s_2 b_2\}) > 0.5$ )  $s_2 b_2$  = next entry of  $D$ ;
  transfer  $s_1 b_1$  from  $D$  to  $C_l$ ;
  transfer  $s_2 b_2$  from  $D$  to  $C_r$ ;
  update( $PT, s_1 b_1, s_2 b_2$ );
  while ( $D$  is not empty) {
     $sb$  = first entry of  $D$ ;  $H_l$  =  $\text{conditional\_entropy}(C_l, sb)$ ;
     $H_r$  =  $\text{conditional\_entropy}(C_r, sb)$ ;
    if ( $H_l < H_r$ ) transfer  $sb$  from  $D$  to  $C_l$ ; update( $PT, C_r, sb$ );
    else transfer  $sb$  from  $D$  to  $C_r$ ; update( $PT, C_l, sb$ );
  }
   $e_l(v)$  = minimum magnitude signed digit in  $C_l$ ;
   $C_l$  =  $\text{right\_shift}(C_l, e_l(v))$ ;
  if ( $u = \text{exist}(G, C_l)$ )  $l(v) = u$ ;
  else {  $l(v) = \text{new\_node}(C_l)$ ; insert( $Q, l(v)$ ); }
   $e_r(v)$  = minimum magnitude signed digit in  $C_r$ ;
   $C_r$  =  $\text{right\_shift}(C_r, e_r(v))$ ;
  if ( $u = \text{exist}(G, C_r)$ )  $r(v) = u$ ;
  else {  $r(v) = \text{new\_node}(C_r)$ ; insert( $Q, r(v)$ ); }
  insert( $G, v, l(v), r(v)$ );
}

```

Fig. 4. Algorithm for vertex decomposition.

Fig. 3 outlines the algorithmic flow for the construction of a minimal MBPG. Propositions 1 to 3 are exercised in the core decomposition operation. Its pseudo code,  $\text{decompose}(v)$  is detailed in Fig. 4.

The graph  $G$  is initialized with three terminal vertices, 1,  $-1$  and 0. A list  $Q$  is used to store the newly generated vertices and it is initialized with a list of roots. The function  $\text{decompose}(v)$  is called to decompose the vertex  $v$  in  $Q$  that has the minimum entropy until  $Q$  is empty. In Fig. 4, The function  $\text{entropy}(v)$  computes the entropy from the PT array. The two signed digits  $s_1 b_1$  and  $s_2 b_2$  with the least entropy and has low certainty to form subexpression themselves are split into two children lists of SPT terms,  $C_l$  and  $C_r$ . The function  $\text{update}(PT, L_1, L_2)$  updates the PT array by decrementing the occurrences of subexpressions that can be formed by one digit from  $L_1$  and the other from  $L_2$ . The function  $\text{conditional\_entropy}(C, sb)$  returns the conditional entropy defined in (6) with  $Y \equiv C$ ,  $X_i \equiv sb$  and  $m \equiv |C|$ . When all signed digits of a decomposed vertex,  $v$  have been exhausted, the left and right edge values of  $v$  are obtained from the least magnitude signed digits of  $C_l$  and  $C_r$ , respectively. The positions of the SPT terms in  $C_l$  and  $C_r$  are shifted accordingly before they are assigned to  $c(l(v))$  and  $c(r(v))$ . The function  $\text{right\_shift}(C, e)$  subtracts  $e$  from the magnitude of

every signed digit in  $C$ . New vertices  $l(v)$  and  $r(v)$  with values  $C_l$  and  $C_r$  are generated if they have not yet existed. The function  $\text{exist}(G, C)$  verifies if a vertex with  $c(v)$  equals the set  $C$  has already been generated in  $G$  to avoid duplication of vertices. If found, the vertex is returned. The function  $\text{new\_node}(C)$  creates a new vertex  $v$  with its output value  $c(v)$  equals the set  $C$  of SPT terms. Finally, the left and right children rooted at  $v$  are inserted into  $G$  by the function  $\text{insert}(G, v, l(v), r(v))$ .

*Example 2:* The 3-tap FIR filter with  $H = \{1132, 815, 556\}$  of Fig. 1 is used to illustrate the proposed method. The coefficients are coded in CSD form as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 0 \end{bmatrix}.$$

The numbers of occurrence of all the weight-two subexpressions are captured in a PT array shown in Example 1.  $G(V, E)$  is initialized with three constant terminal vertices, 0, 1 and  $\bar{1}$ . The roots  $v_1$  to  $v_3$ , corresponding to  $h_1$  to  $h_3$  are inserted into  $G$  and  $Q$ , i.e.,  $V = \{1, -1, v_1, v_2, v_3\}$  and  $Q = \{v_1, v_2, v_3\}$ . Trailing zeros in the CSD number are shifted out and the edge values are adjusted accordingly. In  $\{s_i b_i\}$  notation,  $c(v_1) = \{9, 6, -3, -1\}$ ,  $c(v_2) = \{11, -9, 7, -5, -1\}$  and  $c(v_3) = \{8, 5, -3, -1\}$  with  $e_{v_1} = e_{v_3} = 2$  and  $e_{v_2} = 0$ . The entropy of each vertex,  $v_i$  is calculated using (5) and (7). The calculation of  $H(v_1)$  is illustrated as follows:

$$\begin{aligned} & \sum_{j \neq 1} \text{PT}[s_1 s_j] [|b_1 - b_j| - 1] \\ &= \text{PT}[1] [|9 - 6| - 1] + \text{PT}[-1] [|9 - 3| - 1] \\ & \quad + \text{PT}[-1] [|9 - 1| - 1] \\ &= 2 + 3 + 1 = 6. \end{aligned}$$

From (7), for  $s_1 b_1 = 9$ , we have

$$\begin{aligned} p(\{9, 6\}) &= \text{PT}[1] [|9 - 6| - 1] / 6 = 0.333 \\ p(\{9, -3\}) &= \text{PT}[-1] [|9 - 3| - 1] / 6 = 0.5 \\ p(\{9, -1\}) &= \text{PT}[-1] [|9 - 1| - 1] / 6 = 0.167. \end{aligned}$$

$$H(9) = - \sum_{j \neq 1} p(s_j b_j) \log_2 p(s_j b_j) = 1.4595.$$

The entropies of the other three signed digits can be similarly computed as  $H(6) = 1.5219$ ,  $H(-3) = 1.4595$  and  $H(-1) = 1.5219$ . Therefore, the average entropy of  $v_1$  is  $H(v_1) = 1.4907$ . Since  $H(v_2) = 1.9153$  and  $H(v_3) = 1.5049$ ,  $v_1$  has the minimum entropy and it will be decomposed first.

The signed digits of  $c(v_1) = \{9, 6, -3, -1\}$  are first sorted in order of increasing entropy into  $D = \{9, -3, 6, -1\}$ . According to Proposition 2,  $C_l = \{9\}$  and we need to check the probability  $p(\{9, -3\})$  before assigning  $-3$  to  $C_r$ . Since  $p(\{9, -3\}) = 0.5 \leq 0.5$ ,  $C_r = \{-3\}$  and  $D(v_1) = \{6, -1\}$ . Signed digits that have split into two different children can no longer form a subexpression. Hence, PT array is updated to reflect the change in the statistic of positive and negative subexpressions. The updated PT array is given by

$$\text{PT} = \begin{bmatrix} 2 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

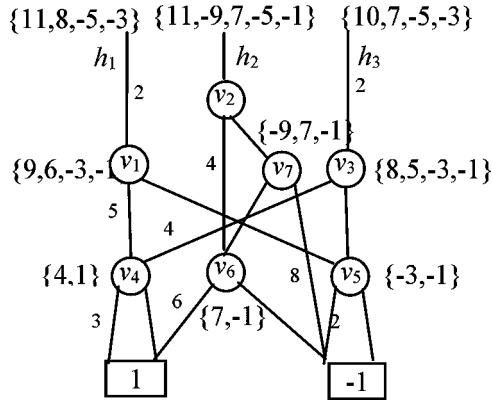


Fig. 5. Generation of minimal MBPG for  $H = \{1132, 815, 556\}$ .

If 6 is assigned to  $C_l$ , from (8),  $p(9|6 \rightarrow C_l) = (PT[1][|6 - 9| - 1]) / (PT[1][|6 - 9| - 1] + PT[-1][|6 - 3| - 1]) = 2 / (2 + 1) = 0.6667$  and the conditional entropy,  $H(9|6 \rightarrow C_l) = 0.39$ . If 6 is assigned to  $C_r$ , we have  $p(9|6 \rightarrow C_r) = 0.3333$ .  $H(9|6 \rightarrow C_r) = 0.5283$ . According to Proposition 3, 6 is assigned to  $C_l$ . Now,  $C_l = \{9, 6\}$ ,  $C_r = \{-3\}$  and  $D(v_1) = \{-1\}$ . The PT array is updated to invalidate  $\{6, -3\}$

$$PT = \begin{bmatrix} 2 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

If the last signed digit  $-1$  in  $D$  is assigned to  $C_l$ , we have

$$p(9|-1 \rightarrow C_l) = \frac{PT[-1][7]}{PT[-1][7] + PT[-1][4] + PT[1][3]} = \frac{1}{1 + 2 + 2} = 0.2$$

$$p(6|-1 \rightarrow C_l) = \frac{PT[-1][4]}{PT[-1][7] + PT[-1][4] + PT[-1][1]} = \frac{2}{1 + 2 + 2} = 0.4.$$

$H(C_l|-1 \rightarrow C_l) = -0.4 \log_2 0.4 - 0.2 \log_2 0.2 = 0.9931$ . Since  $H(C_r|-1 \rightarrow C_r) = 0.5288 < H(C_l|-1 \rightarrow C_l)$ , we have  $C_l = \{9, 6\}$ ,  $C_r = \{-3, -1\}$ . PT is updated to eliminate the statistics due to the subexpressions,  $\{9, -1\}$  and  $\{6, -1\}$ .

$$PT = \begin{bmatrix} 2 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Since  $D(v_1) = \phi$ ,  $v_1$  has been completely decomposed. To eliminate the trailing zeros,  $e_l(v_1) = 8$ ,  $e_r(v_1) = 3$ , giving  $c(l(v_1)) = \{4, 1\}$  corresponding to subexpression 1001 and  $c(r(v_1)) = \{-3, -1\}$  corresponding to the subexpression  $\bar{1}0\bar{1} \equiv -(101)$ . Neither  $c(l(v_1))$  nor  $c(r(v_1))$  exists in  $G$ , they are inserted into  $G$  as new vertices,  $v_4$  and  $v_5$ , respectively.  $v_1$  will be deleted from  $Q$  and  $v_4$  and  $v_5$  will be inserted for further decomposition. As there are only 2 nonzero digits in each of  $v_4$  and  $v_5$ , they will be decomposed into terminal vertices 1 and  $\bar{1}$ . To this end,  $G = \{1, -1, v_1, v_2, v_3, v_4, v_5\}$ . The process will be repeated and Fig. 5 shows the trace of computation towards the final MBPG. The final reduced MBPG of Fig. 5 has the same

topology of the architecture shown in Fig. 2. Since  $|V| = 7$  (excludes terminal vertices), seven adders are needed and the critical path has three-adder delay.

#### IV. SHIFT INCLUSIVE COMPUTATION REORDERING FOR ADDER COMPLEXITY REDUCTION

Many CSE algorithms [6]–[9], [12], [14]–[16] assume uniform adder cost and adder delay for all adders employed in their solution. In practice, although the hardwired shifters contribute zero cost to the logic complexity and logic depth of the multiplier block, they cause varying operand displacements to the adders. Therefore, a better metrology is needed to account for the area–time dependency of each discrete adder on operand lengths due to the relative positional shift of its operands [26]. Since the primary objective of most CSE algorithms is on complexity reduction, and the ripple carry adder (RCA) has been known to be the simplest and power efficient adder, we will use RCA consistently for two’s complement addition throughout this paper. Fig. 6 illustrates the full adder (FA) cost required to add/subtract two operands in the vertex,  $v$  of MBPG using two’s complement RCA. Let  $b_{msb}(v)$  and  $b_{lsb}(v)$  denotes the magnitudes of the most and least significant signed digits of  $c(v)$ , respectively. Assume that the wordlength of the input  $x$  to the filter is  $n$  bits, then the length of the input operand to the adder is  $b_{msb}(\text{child}(v)) + n + e_{\text{child}(v)} - 1$  with  $e_{\text{child}(v)}$  trailing zeros, where  $\text{child}(v)$  is either  $l(v)$  or  $r(v)$ . In Fig. 6, every solid dot represents an unknown bit and  $s$  represents a sign or sign extended bit. Carry free addition with a string of zeros can be hardwired. To add  $l(v)$  and  $r(v)$ , the number of FAs required is given by  $n + \text{msb}_{\max} - \text{lsb}_{\max}$  where  $\text{msb}_{\max} = \max\{b_{msb}(l(v)) + e_{l(v)}, b_{msb}(r(v)) + e_{r(v)}\}$  and  $\text{lsb}_{\max} = \max\{b_{lsb}(l(v)) + e_{l(v)}, b_{lsb}(r(v)) + e_{r(v)}\}$ . The FA at the least significant bit of RCA can be reduced to a half adder (HA) since there is no carry from the  $\max\{e_{l(v)}, e_{r(v)}\}$ th position. To subtract  $r(v)$  from  $l(v)$ , the same number of FAs are needed if  $e_{l(v)} \leq e_{r(v)}$ , since the two’s complement of a string of  $e_{\max}$  zeros results in a string of zeros with a carry-out of one which will be fed into the carry-in of the RCA. If  $e_{l(v)} > e_{r(v)}$ , then the operands can be swapped since  $r(v) - l(v) = -(l(v) - r(v))$  with the sign of the output toggled to preserve the same number of FAs, otherwise,  $e_{l(v)} - e_{r(v)}$  additional FAs will be needed. Output sign change is also necessitated if both operands are negative since  $-l(v) - r(v) = -(l(v) + r(v))$ . The toggling of output sign can be propagated to the next adder in MBPG, making the addition a subtraction and vice versa. In general, the width,  $W$  of the RCA for vertex  $v$  is given by

$$W(v) = n + \text{msb}_{\max} - \text{lsb}_{\max}. \tag{9}$$

The type of adder and its bit width also affect the critical path delay of the multiplier block. It should be noted that the real critical path of an FIR filter predicted by this shift inclusive delay metric may not be the path with the maximum number of discrete adders. Hence, besides the topology of adders in the multiplier block, the internal architecture of each adder needs to be considered to obtain a more accurate estimate of the critical

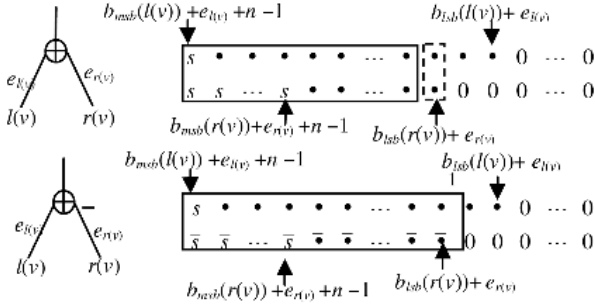


Fig. 6. Full adder cost for addition/subtraction using RCA.

```

Critical_path_delay(G) {
    max_delay = 0;
    for each (root, u of G) {
        delay = path_delay(v);
        if (delay > max_delay) max_delay = delay;
    }
    return max_delay;
}

path_delay(v) {
    if (v = terminal vertex) return 0;
    return W(v) + max{path_delay(l(v)), path_delay(r(v))};
}

```

Fig. 7. Critical path delay computation.

path delay. The worst case delay of the RCA is equal to  $W \cdot t_{FA}$  where  $W$  is the width of the RCA and  $t_{FA}$  is the delay of a FA. The path delay of each tap can be computed by a simple depth first traversal of the MBPG. The pseudo code to compute the critical path delay is shown in Fig. 7 assuming that  $W(v)$  for each vertex,  $v \in V$  has already been computed by (9) and stored in the vertex.

The disjunctive decomposition algorithm of Fig. 4 aims at synthesizing a multiplier block with reduced number of discrete adders. Further reduction of the FA cost and FA delay can be achieved by reducing the operand width of each nonterminal vertex without jeopardizing the cardinality of the vertex set and the depth of MBPG. This is obtained by computation reordering after the MBPG has succeeded in finding good common subexpressions to reduce the number of discrete adders. The steps involved are given as follows.

- Step 1) Select a root of MBPG, make a graph traversal to extract vertices with indegree  $\geq 2$ . The shift-inclusive input set  $\{c(v(\text{child})) + e_{v(\text{child})}\}$  (i.e., the input operand to the adder) of these vertices are inserted into a list  $U$ . The remaining nonzero digits,  $s_i d_i$  of the root that can not be composed directly from the common subexpressions in  $U$  are inserted as isolated nonzero digits into the list  $Q$ .
- Step 2) For each pair of signed digit sets,  $u_i, u_j \in U$ , calculate  $\text{msb}_{\max} = \max\{b_{\text{msb}}(u_i), b_{\text{msb}}(u_j)\}$  and  $\text{lsb}_{\max} = \max\{b_{\text{lsb}}(u_i), b_{\text{lsb}}(u_j)\}$ . For an isolated nonzero digit  $s_i b_i$ ,  $b_{\text{msb}} = b_{\text{lsb}} = b_i$ . The pair of vertices, or subexpressions,  $u_i, u_j \in U$  with the minimum  $\Delta b = \text{msb}_{\max} - \text{lsb}_{\max}$  among all possible pairs from  $U$  will be selected to compose a new

subgraph rooted at  $u_k$  with  $u_i, u_j \in \text{child}(u_k)$ . If there is a tie, the pairs that generates a subgraph with the minimum adder depth (which can be determined by the binary logarithmic of the hamming weight of  $c(u_k)$ ) will be selected.  $u_i$  and  $u_j$  are replaced by  $u_k$  in  $U$ .

- Step 3) Repeat Step 2 until there are only two vertices in  $U$ . These two vertices will compose the root. Repeat Step 1 with the next root until all roots have been recomposed.

Since the number of FAs used is determined by the operand width, Step 2 ensures that the average width of the operands generated by the nonterminal vertices composing the coefficients in MPBG has been reduced by reordering the additions. Since isomorphic subgraphs are kept intact, changes, if any, affect only the ordering of some operands and/or their shifts but not the total number of operators of MBPG. In computation reordering, priority is given to low hamming weight subexpressions so that the adder tree so formed is more balanced and the critical path may be simultaneously reduced.

*Example 3:* Consider the MBPG in Fig. 5 of Example 2. The final PT array is given by

$$\text{PT} = \begin{bmatrix} 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It identifies the weight-two common subexpressions for this coefficient set. The three common subexpressions are  $101 (v_5)$ ,  $1001 (v_4)$  and  $10000\bar{1} (v_6)$ . The shift-inclusive computation reordering is applied to Fig. 5 and the FA cost is calculated as follows.

For coefficient  $h_1$ ,  $v_1$  is a sum of two common subexpressions, so no further FA cost optimization is performed. From (9),  $W(v_4) = n + \max(4, 1) - \max(4, 1) = n$ .  $W(v_5) = n + \max(3, 1) - \max(3, 1) = n$  and  $W(v_1) = n + \max(9, 3) - \max(6, 1) = n + 3$ . Therefore, the adder cost of  $h_1$  is  $3n + 3$  FAs.  $\text{path\_delay}(v_1) = W(v_1) + \max\{W(v_4) + 0, W(v_5) + 0\} = (2n + 3)t_{FA}$ .

For coefficient  $h_2$ ,  $v_2$  is composed of one isolated nonzero digit  $\bar{1}$  and two different shifted versions of a weight-two common subexpression  $10000\bar{1} (v_6)$ . Shift inclusive FA cost optimization is performed as follows. In Step 1,  $U = \{u_1, u_2, u_3\}$  where  $u_1 = \{7, -1\} = 10000\bar{1}$ ,  $u_2 = \{11, -5\} = 10000\bar{1}0000$  and  $u_3 = \{-9\} = \bar{1}0000000$ . In Step 2,  $\Delta b_{12} = \max(7, 11) - \max(1, 5) = 6$ ,  $\Delta b_{13} = \max(7, 9) - \max(1, 9) = 0$ ,  $\Delta b_{23} = \max(11, 9) - \max(5, 9) = 2$ . Therefore,  $u_4 = \{-9, 7, -1\}$  is composed from  $u_1$  and  $u_3$ . Now,  $U = \{u_2, u_4\}$  and the root is generated from  $u_2$  and  $u_4$ . It happens that  $u_4 \equiv v_7$  and  $u_2 \equiv v_6$ , no change is made to the original composition of  $v_2$ . The FA cost is determined by (9).  $W(v_6) = n + \max(7, 1) - \max(7, 1) = n$ .  $W(v_7) = n + \max(7, 9) - \max(1, 9) = n$  and  $W(v_2) = n + \max(11, 9) - \max(5, 1) = n + 6$ . Therefore, the adder cost of  $h_2$  is  $3n + 6$  FAs.  $\text{path\_delay}(v_2) = W(v_2) + \max\{W(v_6) + 0, W(v_7) + \max\{W(v_6) + 0, 0\}\} = W(v_2) + W(v_7) + W(v_6) = (3n + 6)t_{FA}$ . Fig. 8 shows the three different compositions of MBPG by different computation



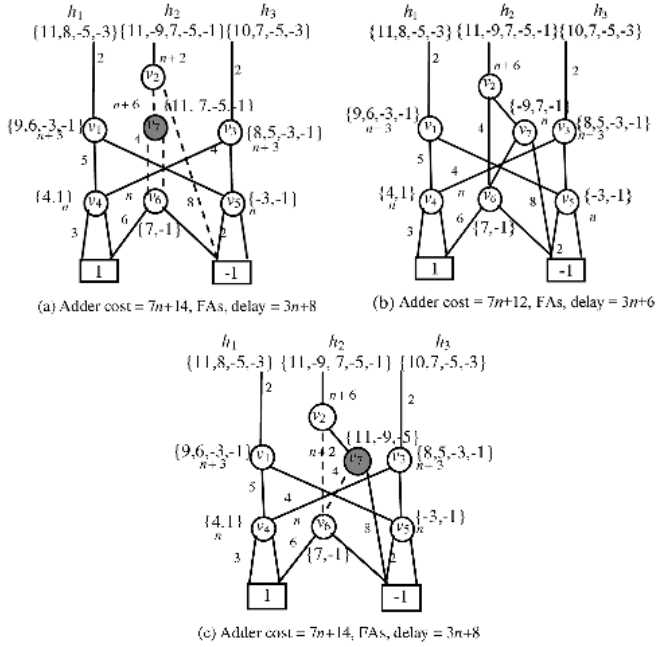


Fig. 8. MBPG after computation reordering: (a)  $\Delta b_{12} = 6$ , (b)  $\Delta b_{13} = 0$ , and (c)  $\Delta b_{23} = 2$ .

orderings by selecting  $\Delta b_{12}$ ,  $\Delta b_{13}$  and  $\Delta b_{23}$  in Step 2. The FA cost for each vertex is annotated.

For coefficient  $h_3$ ,  $v_3$  is a sum of two common subexpressions and no reordering is needed. Taking into account that  $v_3$  is the sum of  $v_4$  and  $v_5$  which are shared with  $h_1$ , only  $W(v_3) = n + \max(8, 3) - \max(5, 1) = n + 3$  additional FAs are needed.  $path\_delay(v_3) = W(v_3) + \max\{W(v_4) + 0, W(v_5) + 0\} = (2n + 3)t_{FA}$ .

The 3-tap filter as a whole requires  $7n + 12$  FAs and has a critical path delay of  $(3n + 6)t_{FA}$ .

V. EXPERIMENTAL RESULTS

In this section, we present numerical results to demonstrate the effectiveness of the proposed MBPG algorithm in reducing the implementation complexity of FIR filters. The adder cost and adder delay of our solutions are compared with several other CSE algorithms [7]–[10], [12] based on the number of FAs and number of  $t_{FA}$  in the critical path of the test filters. The FA cost and FA delay for the solutions of all algorithms are worked out from their shift-and-add topologies using the same computation model of Fig. 6. For FA cost evaluation, wordlength of the input signal is needed. We use an input wordlength of 12 bits for all the experiments since this resolution of ADC is very common. The targeted implementation structure is the multiplier block of transposed direct form filter realized with two’s complement RCAs. The accumulators in the tap-delay line are not considered because the numbers are equal for all algorithms. Symmetrical filters are realized with folded implementation [27] and only half the number of taps is considered. The performances are evaluated in two different sets of filters.

The first set consists of ten commonly referenced benchmark filters, Filters A to J. Their design parameters are listed in Table I.  $N$  is the total number of filter taps,  $L$  is the wordlength

TABLE I  
TEST FILTER SPECIFICATIONS

Filters	Type	$N$	$L$	$f_p$	$f_s$	$r_p$ (dB)	$r_s$ (dB)
A[18]	LP, LS	25	9	0.15	0.25	46	46
B[28]	LP, LS	36	9	0.23	0.29	0.36	94
C[28]	LP, LS	37	9	0.23	0.29	0.34	87
D[29]	LP, -	43	16	0.2	0.3	0.004	30
E[18]	LP, PM	60	16	0.021	0.07	0.2	18
F[28]	LP, LS	63	16	0.1	0.14	0.48	60
G[30]	LP, PM	120	16	0.2	0.22	6	60
H[7]	HP, LS	121	16	0.783	0.74	0.081	80
I[30]	LP, PM	200	12	0.2	0.22	6	60
J[31]	BP, -	281	16	0.01-0.99	0-0.002;0.998-1	0.5	60

TABLE II  
FA COST COMPARISON

Filter	CSD	BIN	BHM	Pasko	NRSCSE	Hartley	C1	BCSE	MBPG
A	105	161	78	75	75	75	72	94	75
B	60	100	64	60	60	60	60	64	60
C	60	134	63	60	60	60	60	84	60
D	934	1359	781	723	725	798	619	873	709
E	912	1429	482	563	555	712	546	684	548
F	706	1478	402	461	467	550	362	483	435
G	2031	2750	1021	1289	1290	1542	933	1266	1238
H	1984	2723	969	1170	1213	1549	884	1181	1147
I	682	935	414	455	398	446	375	413	397
J	2337	3059	1189	1355	1278	1650	1059	1284	1266

of the filter coefficients,  $f_p$  and  $f_s$  denote the pass band and stop band edge frequencies that are normalized to 1, respectively.  $r_p$  and  $r_s$  denote the pass band and stop band ripples in dB, respectively. The filter type is specified in the column labeled ‘Type’, where LP, HP and BP represent low-pass, high-pass and bandpass filters, respectively. The filter coefficients are generated using Parks-McClellan (PM) and least square (LS) algorithms. ‘-’ indicates unknown coefficient synthesis algorithm. The filter coefficient sets can be excerpted directly from the references listed in the first column, regardless of synthesis algorithms.

Tables II and III show the logic complexity and logic depth comparisons of our proposed MBPG algorithm with the baseline CSD and Binary implementations, and several other CSE algorithms such as NRSCSE [8], Pasko [7], Hartley [9], BHM [12], C1 [32] and BCSE [10]. CSD and BIN refer to the baseline implementations where all filter coefficients are recoded in CSD and Binary format with no further optimization [33] other than the adder tree for each coefficient is balanced to minimize the critical path delay.

From the results of Tables II and III, the proposed MBPG method performs well overall in terms of both logic complexity and logic depth. The average reductions in the FA delay of critical path over CSD, BIN, BHM, Pasko, NRSCSE, Hartley, C1 and BCSE are 12.73%, 56.27%, 58.49%, 23.99%, 9.35%, 7.59%, 5.68% and 33.63%, respectively. On average, the FA costs are reduced by 37.48%, 54.74%, 5.26%, 2.99%, 19.30% and 11.41% over CSD, BIN, Pasko, NRSCSE, Hartley and BCSE, respectively. However, the average FA cost of the proposed method is 3.05% and 12.71% worse than that of BHM and C1. This is because BHM and C1 algorithm search

TABLE III  
COMPARISON OF FA DELAY IN CRITICAL PATH

Filter	CSD	BIN	BHM	Pasko	NRSCSE	Hartley	C1	BCSE	MBPG
A	30	60	27	27	27	27	24	46	27
B	12	26	28	12	12	12	12	28	12
C	12	72	27	12	12	12	12	48	12
D	60	108	170	71	54	52	80	75	54
E	33	108	68	43	47	47	31	53	32
F	50	120	93	49	49	49	44	49	45
G	60	96	127	74	52	58	71	66	51
H	60	96	178	74	50	50	50	59	49
I	60	60	85	59	49	49	49	49	44
J	57	108	132	65	53	53	53	83	52

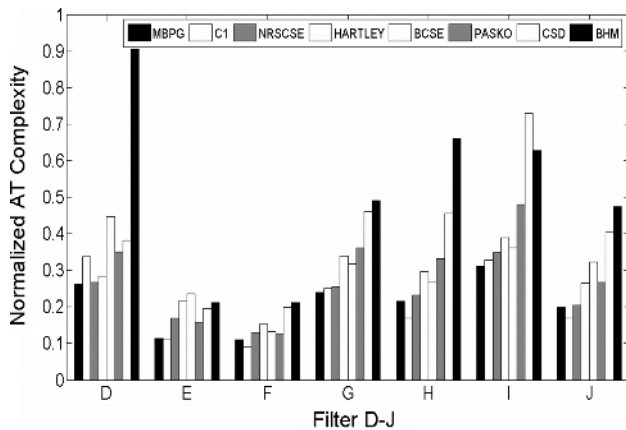


Fig. 9. Normalized AT complexity comparison for Filters D to J. For each filter, the algorithms from left to right are: MBPG, C1, NRSCSE, Harley, BCSE, Pasko, CSD and BHM.

exhaustively from all possible primitive adder graph topologies for the partial sums formed in any single coefficient multiplication. It is therefore capable of finding certain redundancies not identifiable by our heuristic pattern-based approach. The premium for the low adder cost of BHM is its low throughput rate. The critical path delays of its solutions are dramatically longer than those of the proposed MBPG. C1 [32] is one of the best algorithms among the latest proposed enhancements to BHM for its logic depth reduction. It maximizes the use of cost-1 partial sums to reduce the logic depth and searches exhaustively among all existing graphs to select a solution with best area-time performance. This results in a very high computational complexity.

The overall area-time (AT) performance is measured by the product of FA cost and FA delay. The AT results for Filters D-J with more than 40 taps from Table I are normalized by the AT of BIN (i.e., BIN method has a normalized AT of 1) and plotted in Fig. 9. For each filter, the normalized AT of the eight algorithms are displayed in the order listed in the legend. It is evident that MBPG and C1 have comparable AT complexities, which are much lower than those of all other algorithms.

The second set of filters involves the channelizers of wide-band receiver operate in the intermediate frequency (IF). The performance of our proposed MBPG method is evaluated and compared with other algorithms using two communication filter design examples with the same coefficient wordlength,  $L =$

TABLE IV  
FA COST COMPARISON OF FIR1 AND FIR2

Filter	FIR1					FIR2				
	N	200	460	610	940	1180	230	450	650	800
N	200	460	610	940	1180	230	450	650	800	
CSD	841	761	746	758	839	691	729	786	747	
BIN	1036	929	950	948	1041	815	923	1003	916	
BHM	500	466	476	476	523	420	462	500	450	
Pasko	481	447	429	437	504	395	421	449	422	
NRSCSE	460	433	430	430	475	408	430	458	431	
Hartley	538	477	474	474	501	446	460	517	490	
C1	446	421	421	423	468	402	426	454	421	
BCSE	492	449	438	457	499	417	475	479	444	
MBPG	454	423	420	420	465	398	424	452	430	

TABLE V  
FA DELAY COMPARISON OF FIR1 AND FIR2

Filter	FIR1					FIR2				
	N	200	460	610	940	1180	230	450	650	800
N	200	460	610	940	1180	230	450	650	800	
CSD	30	30	30	30	30	48	48	48	48	
BIN	60	60	60	60	60	60	60	60	60	
BHM	80	80	84	84	84	63	84	84	63	
Pasko	45	45	43	45	46	45	45	45	45	
NRSCSE	44	44	44	44	44	48	48	48	48	
Hartley	45	45	45	45	42	48	48	48	48	
C1	30	30	30	30	30	46	46	46	45	
BCSE	60	60	60	60	60	61	61	47	61	
MBPG	30	30	30	30	30	42	42	42	42	

12 and varying filter length,  $N$ . They are designed to demonstrate the potential of our proposed algorithm in realizing application-specific integrated filters that meet the stringent narrow transition band and high sampling frequency filtering requirements. FIR1 is a linear phase FIR filter (LPFIR) employed in the filter bank channelizer of Digital Advanced Mobile Phone Systems (D-AMPS) in [4]. The decimation is moved to the left of the bandpass filters using the noble identity and the sampling rate chosen is 34.02 MHz as in [4]. The channel filters extract 30-kHz D-AMPS channels from the input signal after down-sampling by a factor of 350. The pass band and stop band edges are 30 and 30.5 kHz, respectively. The peak pass band ripple is chosen as 0.1 dB. The filter stop band specifications are chosen as in the D-AMPS standard [4]. The lengths of the LPFIR filters are determined using the following equation from [34]:

$$N = \frac{-10 \log_{10} \partial_1 \partial_2 - 13}{14.6 \Delta f} + 1 \quad (10)$$

where  $\partial_1$  and  $\partial_2$  are the pass band and stop band ripples, respectively, and  $\Delta f$  is the normalized width of the transition band. The filter lengths,  $N = 200, 460, 610, 940,$  and  $1180$  corresponding to stop band attenuation of  $-24$  dB,  $-48$  dB,  $-65$  dB,  $-85$  dB, and  $-96$  dB, respectively, are tested. FIR2 is the channel filter employed in receivers for the personal digital cellular (PDC) standard. The sampling rate of the wide band signal is 25.6 MHz, which covers 1024 channels of 25.5-kHz spacing after down sampling by a factor of 350. The pass band and stop band edges are 25 and 25.5 kHz, respectively. The peak pass band ripple is chosen as 0.1 dB. The filter stop band specifications are chosen as in the D-AMPS standard and the length of

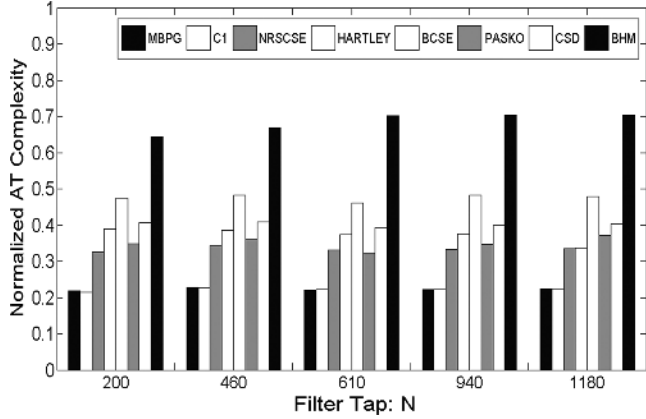


Fig. 10. Normalized AT complexity comparison for FIR1. For each  $N$ , the algorithms from left to right are: MBPG, C1, NRSCSE, Harley, BCSE, Pasko, CSD, and BHM.

the filter,  $N$  is determined by (10).  $N = 230, 450, 650$  and  $800$  are used to attain an attenuation of  $-30$  dB,  $-60$  dB,  $-80$  dB and  $-90$  dB, respectively, for this evaluation.

The FA cost of various algorithms for FIR1 and FIR2 are compared in Tables IV, and the FA delays are analyzed in Tables V.

For FIR1, the average reductions of FA cost by our proposed method are, respectively, 44.66%, 55.49%, 10.61%, 4.94%, 2.04%, 11.38%,  $-0.14\%$  and 6.51% over CSD, BIN, BHM, Pasko, NRSCSE, Hartley, C1 and BCSE. The solutions produced by MBPG are notably faster. The average reductions of the critical path delay are 0%, 50.00%, 63.57%, 33.00%, 31.82%, 32.38%, 0% and 50.00% over CSD, BIN, BHM, Pasko, NRSCSE, Hartley, C1 and BCSE respectively. For FIR2, in comparison with CSD, BIN, BHM, Pasko, NRSCSE, Hartley, C1 and BCSE, the average FA cost reductions are 42.29%, 53.31%,  $-1.01\%$ , 6.88%, .1.35%, 10.85%,  $-0.06\%$  and 6.02%, respectively, and the average critical path delay reductions are 12.50%, 30.00%, 41.67%, 6.67%, 12.50%, 12.50%, 8.19% and 26.02%, respectively. The rates of area reduction and speed improvement vary with the length of the filter. It is possible to have a lower FA cost for filters with a large number of taps compared to filters with relatively fewer taps when there is a substantial reduction of nonzero digits in the CSD coefficients in the former case. In a few cases where the FA costs of our algorithm are slightly higher than those of Pasko and BHM, our FA delays are much lower. For example, MBPG shows a critical delay reduction of 8.19% on average over C1 for FIR2. The AT performances of all algorithms normalized with respect to the AT complexity of BIN for FIR1 and FIR2 are plotted in Figs. 10 and 11. These figures show that our proposed MBPG can generate conspicuously faster transposed direct form FIR filters with low adder cost in general.

MBPG is more computational efficient than C1. By measuring the execution time of both Matlab programs on the same PC with 3.2 GHz Pentium IV and 512 MB of system memory for all the filters tested, it is found that our algorithm is at least thrice faster than C1. This is because the subexpression statistics in the PT table need to be established only once and updated locally thereafter. As the number of nonzero digits per vertex

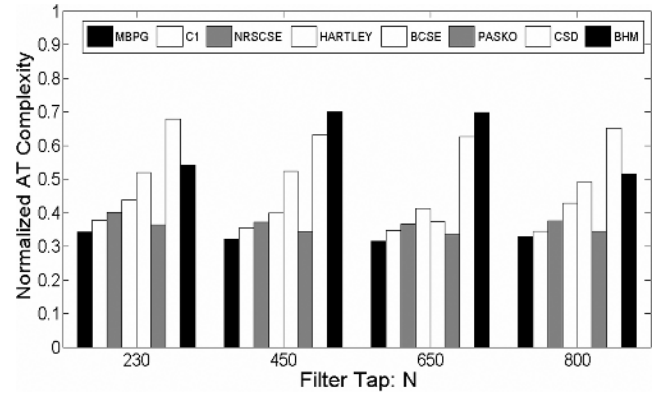


Fig. 11. Normalized AT complexity comparison for FIR2. For each  $N$ , the algorithms from left to right are: MBPG, C1, NRSCSE, Harley, BCSE, Pasko, CSD, and BHM.

decreases exponentially from the root, the number of computations per vertex reduces rapidly as more vertices and isomorphic subgraphs are formed. The computational complexity of our algorithm can be analyzed as follows.

It is cited by [9] that the number of nonzero digits is reduced by an average of 33% in CSD representation over the normal binary representation. Therefore, the average number of nonzero digits of an arbitrary number in CSD representation is  $L/3$ , where  $L$  is the wordlength. The number of computations required to decompose a vertex with  $j$  nonzero digits into its children vertices can be divided into two parts. It takes  $C_2^j = j(j-1)/2$  probability computations to split a vertex, followed by  $(j+1)(j-2)/2$  conditional probability computations to assign the  $j-2$  nonzero digits to either the left or right child. Thus, the total number of operations required is  $j^2 - j - 1$ . For each coefficient with an average number of nonzero digits,  $j = L/3$ , the number of operations required is  $(L^2 - 3L - 9)/9$  in the first level. After the first level of decomposition, the average number of nonzero digits in each child vertex is equal to  $L/6$ . Assuming the worse case scenario where there is no isomorphic subgraph, there will be two vertices per coefficient in the second level. Therefore, the number of operations required in the second level is given by  $(L^2 - 6L - 36)/18$ . The number of operations per coefficients becomes  $(L^2 - 12L - 144)/36$  with four offspring vertices in the third level. Assuming a balanced tree decomposition, the depth of the tree is given by  $\log_2(L/3)$ . Since each vertex in the last level has no more than two nonzero digits, no computation is required in the last level. The total number of operations per coefficient is given by

$$\sum_{i=1}^d \left\{ \frac{L^2}{9} \left( \frac{1}{2} \right)^{i-1} - \frac{L}{3} - 2^{i-1} \right\} = \frac{2L^2}{9} \left\{ 1 - \left( \frac{1}{2} \right)^d \right\} - \frac{L}{3}(d) - (2^d - 1) \quad (11)$$

where  $d = \lfloor \log_2(L/3) \rfloor$ .

Since  $d \leq \log_2(L/3)$ ,  $2^d \leq L/3$ . (11) is bounded by

$$\frac{2L^2}{9} \left\{ 1 - \left( \frac{3}{L} \right) \right\} - \frac{L}{3} \left( \log_2 \frac{L}{3} \right) - \left( \frac{L}{3} - 1 \right) = \frac{2L^2}{9} - L - \frac{L}{3} \left( \log_2 \frac{L}{3} \right) + 1. \quad (12)$$

In practice, the word length of the coefficients is limited. From (12), the average number of operations per coefficient is less than 51 even if  $L = 20$ . The actual number of operations per coefficient is much lower as there will be shared vertices. Thus, the computation complexity for an  $N$ -tap FIR filter is  $O(N)$ .

In summary, the three propositions in entropy based decomposition have efficiently harnessed isomorphism in the generation of MBPG and the arithmetic operators have been ordered to balance the depth of the graph without annihilating the isomorphic vertices. This is unlike BHM for which the shortening of every single coefficient adder distance with minimal number of adders from existing fundamentals causes an upsurge in logic depth. This critical path extension in BHM has been shown to be more severe as the filter length increases. In fact, high order filters actually benefited more from our method as the intricate redundancy of many different subexpressions in a large number of coefficients can be quantitatively displayed by the entropy measure and hence better trade-off can be made without resorting to computational intensive exhaustive search.

## VI. CONCLUSION

Multiplications with a set of constants are abundant in application-specific digital filters. Optimizing the implementation of FIR filters with a minimal number of shift-and-add operations has been well delved by many researchers. In this paper, we provide an entirely different insight and approach to this problem by exploiting the prowess of information theory on directed-acyclic graph representation of the transposed direct form structure of FIR filters. An appealing multirooted binary partition graph (MBPG) data structure has been devised for this purpose. Using this data structure, a set of fixed point coefficients can be decomposed into subsets of signed digit patterns whose coding redundancy can be theoretically assessed by their entropy and conditional entropy. To construct a reduced size MBPG, the partitioning of fixed point coefficients into different subsets of signed digits is guided by three propositions. The entropy and conditional entropy calculations are based on the probability of occurrences of symbol pairs. Therefore, the proposed method is applicable to any positional representation of coefficients. The correlation of operand lengths and adder complexity is also illustrated with two's complement ripple carry adder, from which the number of FAs required to realize each vertex of MBPG is analytically determined. This makes it possible to exploit the bit level information associated with each vertex to reduce the size of some arithmetic operators by partially modifying the graph topology. Experimental results on benchmark filters reported in the literature and design examples of communication filters based on D-AMPS and PDC cellular standards show that the proposed algorithm is capable of designing FIR filters with an average FA cost reduction of 40.38% over the baseline CSD implementation. The critical path delay has also been significantly reduced by 25.77% on average.

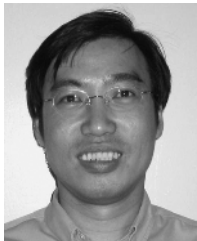
## ACKNOWLEDGMENT

The authors would like to thank Prof. A. G. Dempster for sharing with us the source codes of their algorithms.

## REFERENCES

- [1] J. E. Gunn, K. Barron, and W. Ruczcyk, "A low-power DSP core-based software radio architecture," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 574–590, Apr. 1999.
- [2] H. Samuelli and B. C. Wong, "A VLSI architecture for a high-speed all-digital quadrature modulator and demodulator for digital radio applications," *IEEE J. Sel. Areas Commun.*, vol. 8, no. 8, pp. 1512–1519, Oct. 1990.
- [3] A. P. Vinod and E. M. K. Lai, "On the implementation of efficient channel filters for wideband receivers by optimizing common subexpression elimination methods," *IEEE Trans. Comput.-Aided Design Integr. Circuits*, vol. 24, no. 2, pp. 295–304, Feb. 2005.
- [4] 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. Circuits Syst. I, Fundam. Theory Appl.*, vol. 15, no. 2, pp. 151–165, Feb. 1996.
- [5] Y. J. Yu and Y. C. Lim, "Design of linear phase FIR filters in subexpression space using mixed integer linear programming," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 10, pp. 2330–2338, Oct. 2007.
- [6] F. Xu, C. H. Chang, and C. C. Jong, "Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common subexpressions," *IEEE Trans. Comput.-Aided Design Integr. Circuits*, vol. 26, no. 10, pp. 1898–1907, Oct. 2007.
- [7] R. Paško, P. Schaumont, V. Derudder, S. Vernalde, and D. Đuračková, "A new algorithm for elimination of common subexpressions," *IEEE Trans. Comput.-Aided Design Integr. Circuits*, vol. 18, no. 1, pp. 58–68, Jan. 1999.
- [8] M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 3, pp. 196–203, Mar. 2002.
- [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] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for implementing low complexity FIR filters in software defined radio receivers," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kos, Greece, May 2006, pp. 4515–4518.
- [11] C. H. Chang, J. Chen, and A. P. Vinod, "Maximum likelihood disjunctive decomposition to reduced multirooted DAG for FIR filter design," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kos, Greece, May 2006, pp. 613–616.
- [12] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 9, pp. 569–577, Sep. 1995.
- [13] O. Gustafsson, A. G. Dempster, and L. Wanhammar, "Extended results for minimum-adder constant integer multipliers," in *Proc. IEEE Int. Symp. Circuits and Syst.*, Scottsdale, AZ, USA, 2002, vol. 1, pp. 1-73–1-76.
- [14] D. R. Bull and D. H. Horrocks, "Primitive operator digital filters," *Proc. Inst. Elect. Eng. G*, vol. 138, no. 3, pp. 401–412, Jun. 1991.
- [15] H. J. Kang and I. C. Park, "FIR filter synthesis algorithms for minimizing the delay and the number of adders," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 8, pp. 770–777, Aug. 2001.
- [16] F. Xu, C. H. Chang, and C. C. Jong, "Modified reduced adder graph algorithm for multiplierless FIR filters," *Electron. Lett.*, vol. 41, no. 6, pp. 302–303, 2005.
- [17] G. De Micheli, *Synthesis and Optimization of Digital Circuits*. New York: McGraw-Hill, 1994.
- [18] H. Samuelli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, no. 7, pp. 1044–1047, Jul. 1989.
- [19] A. Yurdakul and G. Dundar, "Multiplierless realization of linear DSP transforms by using common two-term expressions," *J. VLSI Signal Process.*, vol. 22, pp. 163–172, Sep. 1999.
- [20] O. Gustafsson and L. Wanhammar, "ILP modelling of the common subexpression sharing problem," in *Proc. IEEE 9th Int. Conf. Electron., Circuits Syst.*, Sep. 2002, vol. 3, pp. 1171–1174.
- [21] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution algorithm for common subexpression elimination in digital filter design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 10, pp. 695–700, Oct. 2005.

- [22] 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 filter," *IEEE Trans. Circuits Syst. I, Reg., Papers*, vol. 51, no. 11, pp. 2215–2221, Nov. 2004.
- [23] D. Miller, A. Rao, K. Rose, and A. Gersho, "A maximum entropy approach for optimal statistical classification," in *Proc. IEEE Workshop Neural Netw. Signal Process.*, Aug./Sep. 1995, pp. 58–66.
- [24] E. Halperin and R. M. Karp, "The minimum-entropy set cover problem," *Theor. Comput. Sci.*, vol. 348, no. 2, pp. 240–250, Dec. 2005.
- [25] A. I. Khinchin, *Mathematical Foundations of Information Theory*. New York: Dover, 1957.
- [26] A. P. Vinod and E. M.-K. Lai, "An efficient coefficient-partitioning algorithm for realizing low-complexity digital filters," *IEEE Trans. Comput.-Aided Design Integr. Circuits*, vol. 24, no. 12, pp. 1936–1946, Dec. 2005.
- [27] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley-Interscience, 1999.
- [28] Y. C. Lim and S. Parker, "Discrete coefficient FIR digital filter design based upon an LMS criteria," *IEEE Trans. Circuits Syst.*, vol. CAS-30, no. 10, pp. 723–739, Oct. 1983.
- [29] J. Laskowski and H. Samueli, "A 150-MHz 43-tap half-band FIR digital filter in 1.2  $\mu\text{m}$  CMOS generated by silicon compiler," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1992, pp. 11.4.1–11.4.4.
- [30] A. P. Vinod, E. M.-K. Lai, and P. K. Meher, "An improved common subexpression elimination method for realizing FIR filters with minimum logic operators and logic depth," *Int. J. Circuits, Syst. Signal Process.*, 2007, submitted for publication.
- [31] T. Raita-Aho, T. Saramaki, and O. Vainio, "A digital filter chip for ECG signal processing," *IEEE Trans. Instrum. Meas.*, vol. 43, no. 4, pp. 644–649, Aug. 1994.
- [32] A. G. Dempster, S. S. Demirsoy, and I. Kale, "Designing multiplier blocks with low logic depth," in *Proc. IEEE Int. Symp. Circuits Syst.*, Phoenix, AZ, May 2002, vol. 5, pp. 773–776.
- [33] Y. Wang and K. Roy, "CSDC: A new complexity reduction technique for multiplierless implementation of digital FIR filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 9, pp. 1845–1853, Sep. 2005.
- [34] J. G. Prokias and D. G. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1998.

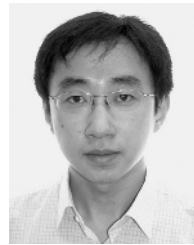


**Chip-Hong Chang** (S'92–M'98–SM'03) received the B.Eng. (Hons) degree from National University of Singapore, Singapore, in 1989, and the M.Eng. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 1993 and 1998, respectively.

He served as a Technical Consultant in industry prior to joining the School of Electrical and Electronic Engineering, NTU in 1999, where he is now an Associate Professor. He holds joint appointments at the university as Deputy Director of the Centre for

High Performance Embedded Systems (CHiPES) since 2000, and Program Director of the Centre for Integrated Circuits and Systems (CICS) since 2003. His current research interests include low power arithmetic circuits, constrained driven architectures for digital signal processing, and digital watermarking for IP protection. He has published three book chapters and more than 130 research papers in international refereed journals and conferences.

Dr. Chang serves as an Editorial Advisory Board Member of The Open Electrical and Electronic Engineering Journal (OEEE). He is listed in the 2008 Marquis Who's Who in the World and the 2000 Outstanding Intellectuals of the 21st Century by the International Biographical Center. He is a Fellow of IET.



**Jiajia Chen** received the B.Eng. degree from Nanyang Technological University, Singapore, in 2004. Since November 2004, he has been working towards the Ph.D. degree in electrical and electronic engineering at the same university.

He was working as a firmware engineer in industry in 2004. Since November 2004, he has been a Teaching Assistant at Nanyang Technological University, Singapore. His main research interest includes computational transformations of low-complexity digital filters, reconfigurable filters, and filter architectural optimization.



**A. P. Vinod** (M'01–SM'07) received the B.Tech. degree in instrumentation and control engineering from University of Calicut, Kerala, India, in 1994 and the M.Engg. and Ph.D. degrees in computer engineering from Nanyang Technological University, Singapore, in 2000 and 2004, respectively.

He was with Kirloskar, Pune, India, from October 1993 to October 1995, and Tata Honeywell, Pune, India, from November 1995 to May 1997. During June 1997 to November 1998, he was associated with Shell Singapore. From September 2000 to September 2002, he was a Lecturer in the School of Electrical and Electronic Engineering at Singapore Polytechnic, Singapore. He was a lecturer in the School of Computer Engineering at Nanyang Technological University (NTU), Singapore, from September 2002 to November 2004, and since December 2004, he has been an assistant professor in NTU. His research interests include digital signal processing, low complexity circuits for signal processing, number theoretic transforms and software radio.