Global Optimization of Common Subexpressions for Multiplierless Synthesis of Multiple Constant Multiplications

Yuen-Hong Alvin Ho, Chi-Un Lei, Hing-Kit Kwan and Ngai Wong
Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
Email: {alvin, kwanhk, yliu, nwong}@eee.hku.hk

Abstract— In the context of multiple constant multiplication (MCM) design, we propose a novel common subexpression elimination (CSE) algorithm that models the optimal synthesis of coefficients into a 0-1 mixed-integer linear programming (MILP) problem. A time delay constraint is included for synthesis. We also propose coefficient decompositions that combine all minimal signed digit (MSD) representations and the shifted sum (difference) of coefficients. In the examples we demonstrate, the proposed solution space further reduces the number of adders/subtractors in the MCM synthesis.

I. Introduction

Multiple constant multiplication (MCM) constitutes a typical fixed-point arithmetic operation in digital signal processing. It is the focus of a lot of research on high-speed and low-power devices in communication systems [1] and signal processing systems [2]. In multiplierless MCM, multipliers are replaced by simpler components such as adders and hard-wired shifts ("adders" in our paper include also subtractors as their hardware costs are similar). By using the negative digits (substractor in circuit) in their signed-digit representations, coefficients may be synthesized with fewer adders, therefore the area and power consumption of the circuit can be reduced. An example of a multiplier-based and a multiplierless-based MCM implementations are shown in Figs. 1 and 2, respectively, wherein 4 multiplications are replaced by 6 adders and 6 hard-wired shifts. Such multiplierless MCMs are utilized, for example, in the design of finite-impulse-response filters [3].

Given a set of integer coefficients, the common subexpression elimination (CSE) (e.g., [3–8]) is further used for minimizing the number of adders in a MCM block. The CSE algorithms search for common subexpressions from different signed-digit representations, and re-use the searched common subexpressions for later synthesis to reduce the total number of additions.

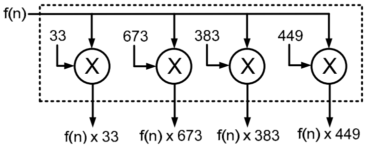

Fig. 1. A multiplier-based MCM example.

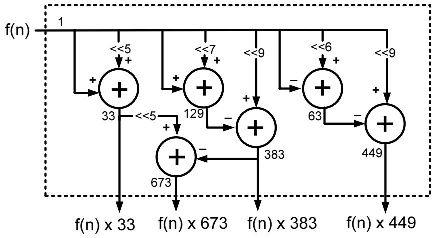

Fig. 2. A multiplierless-based MCM example. ($\ll$: binary shift)

For example, when searching for the synthesis of 673, the binary representation of 673 (1010100001) may be decomposed as $1 \times 2^9 + 161 \left(\boxed{1}010100001 = 1 \ll 9 + 10100001\right)$, $5 \times 2^7 + 33(\boxed{1}0\boxed{1}0100001 = 101 \ll 7 + 100001)$, or $17 \times 2^5 + 129 \left(\boxed{1}010\boxed{1}00001 = 10001 \ll 5 + 10000001\right)$. The digits surrounded by boxes are the portion of subexpression obtained by hard-wired shifts. One of such decompositions will then be chosen at the end of the algorithm. 673 can also be used as a subexpression for other coefficients.

In general, a larger search space allows a better chance of finding more common digit patterns between coefficients, and may lead to a better solution otherwise unobtainable in a smaller sub-space. However, as a nature of NP-complete problems, the complexity grows exponentially as the search space increases. Most CSE algorithms (e.g., [3–8]) are heuristic. They are limited to particular search spaces (signed-digit representations) such as binary [8], canonical signed digit (CSD) [4, 8], minimal signed digit (MSD) [5, 7, 8], and all signed digit [6]. Only the technique in [8] guarantees the optimal solution.

Specifically, Flores *et al.* [8] model the CSE problem as a Boolean network that covers all possible syntheses of coefficients which can be used in the MCM. The network is then converted into a 0-1 pure integer linear program-

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

ming (PILP) problem which is solved with a satisfiability (SAT)-based solver. The model of [8], however, requires a lot of binary variables whose number equals the total number of gates in the circuit plus the primary inputs. This leads to long computation time for the optimal solutions. Moreover, [8] compares the search spaces of binary, CSD, and MSD representations only. And it does not include a time-delay constraint to balance between circuit area and circuit speed, which is important for high-speed system design.

Many solvers use branch and bound (BB) algorithms to solve PILP problems (e.g., [9]). In short, BB algorithms first relax a PILP problem into a linear programming (LP) problem: if C_n is one of the binary variables, it is relaxed to $0 \leq C_n \leq 1$ and then solved with LP. If the C_n thus solved is fractional, BB algorithms subdivide C_n into two sub-problems with $C_n = 0$ and $C_n = 1$. This process goes on iteratively until all integrality requirements are met. Optimality of the solution is guaranteed by combining the BB framework as well as the simplex algorithm in each node of the branching tree.

In this paper, the minimization of the number of required adders to synthesize a MCM block is modeled as a 0-1 mixed-integer linear programming (MILP) problem. (In a MILP problem, only certain variables are integer while the rest are allowed to be continuous.) Compared to a PILP model, the problem contains fewer (binary and total) variables and fewer constraints, therefore it can be solved more efficiently, e.g., using a MILP solver [9].

The proposed 0-1 MILP model further allows the enforcement of a maximum logic depth of a MCM design, which has not been discussed in [8]. Moreover, we propose a new search space that combines all MSD representations and the shifted sum (difference) of coefficients. Each coefficient is decomposed according to a set of constraints in a defined search space and stored into a lookup table. Synthesis of a coefficient can then be retrieved instantly. The proposed search space is compared against the well known search space of MSD representations using our global optimization approach. Maximum logic depth constraint is proposed for trade-off designs between circuit speed and circuit area.

The paper is organized as follows. In Section II, different feasible solution spaces are explained. Section III presents the formulation of the 0-1 MILP problem. Section IV compares the results of various search spaces and contrasts them against some representative results in [8]. Section V draws the conclusion.

II. Feasible Solution Spaces

To begin with, some notational conventions are specified. "n" is the integer value of the coefficient. "#nzbit" denotes the number of non-zero bits in the CSD or MSD representations. "$Oset$" is the set of coefficients that connect to the output of the MCM block. "LT_n" is the lookup table of the coefficient n. "L_n" is the total number of rows in LT_n. "$Cset$" is the set of coefficients that may be used to synthesize coefficients in $Oset$. "D_{max}" is a constant representing the maximum logic depth.

An arbitrary positive integer coefficient n can be decomposed as:

$$n = \pm S_1 \times 2^p \pm S_2 \times 2^q, \tag{1}$$

where

1. n, S_1, and S_2 are positive and odd integers;
2. $S_1 \neq n$ and $S_2 \neq n$.

S_1 and S_2 are subexpressions of n; p and q are the numbers of left shifted bits of S_1 and S_2, respectively. Since there are infinite decompositions for a coefficient satisfying (1) (e.g., 3 may be cast as $2^2 - 1$, $2^{10} - 1021$, $2^{20} - 1048573$, $\cdots$), the feasible solution space is generally fixed at a finite set in the CSE algorithms. Traditionally, search space of binary representation (0 and 1) is used. CSE algorithm based on Canonical Signed Digit (CSD) with a signed digital representation (-1, 0 and 1) is proposed later [10]. The CSD is unique and has two properties: 1) The number of non-zero digits (i.e. -1 and 1) is minimal; 2) Two non-zero digits are not adjacent. CSE algorithm based on Minimal Signed Digit (MSD) is proposed in [5], which neglects the second property. Therefore, it allows more possible combinations for a coefficient and gives a further reduction in sharing subexpressions with other coefficients. An example of the binary, the CSD, and the MSD representations of 23 are shown as below (where $\underline{1}$ represents the digit -1):

1. Binary: 10111 (#nzbit = 4);
2. CSD: 10$\underline{1}$00$\underline{1}$ (#nzbit = 3);
3. MSD: 10$\underline{1}$00$\underline{1}$ (#nzbit = 3), 1100$\underline{1}$ (#nzbit = 3).

The following sub-section describes two search spaces with different mathematical constraints imposed on (1) for optimal synthesis. The MCM coefficients of $Oset$ = {673, 383, 449, 33} are used for illustration.

A. Search space of MSD representations

Search spaces from the MSD representations are generated based on all digit patterns in the MSD representations described in [8]. Table I shows the lookup table for the coefficients in $Oset$. Decompositions from rows 1 to 7, 1 to 5, and 1 to 3 show the search spaces of MSD representations in LT_{673}, LT_{383}, and LT_{449}, respectively. The "Search Space" columns describe different digit patterns extracted from the MSD representations. Since the #nzbit of 33 is two, it can be synthesized with one adder with the logic depth of one. It is considered to be the best synthesis of the coefficient and therefore LT_{33} is fixed at one row without any further expansion of the search space.

TABLE I
COMPLETE TABLE OF $Oset = \{673, 383, 449, 33\}$, WHERE $\underline{1}$ DIGIT REPRESENTS THE -1 DIGIT.

Row	LT_{673}	Search Space	LT_{383}	Search Space	LT_{449}	Search Space	LT_{33}
1	$1 \times 2^9 + 161$	$\boxed{1}010100001$	$1 \times 2^9 - 129$	$\boxed{1}0\underline{1}0000000\underline{1}$	$1 \times 2^9 - 63$	$\boxed{1}00\underline{1}000001$	$1 \times 2^5 + 1$
2	$1 \times 2^7 + 545$	$10\boxed{1}0100001$	$-1 \times 2^7 + 511$	$10\boxed{\underline{1}}000000\underline{1}$	$-1 \times 2^6 + 513$	$100\boxed{\underline{1}}000001$	(directly
3	$1 \times 2^5 + 641$	$1010\boxed{1}00001$	$-1 + 3 \times 2^7$	$10\underline{1}000000\boxed{\underline{1}}$	$1 + 7 \times 2^6$	$100\underline{1}00000\boxed{1}$	synthesized)
4	$1 + 21 \times 2^5$	$101010000\boxed{1}$	$1 \times 2^8 + 127$	$\boxed{1}1000000\underline{1}$			
5	$5 \times 2^7 + 33$	$\boxed{1}0\boxed{1}0100001$	$1 \times 2^7 + 255$	$1\boxed{1}000000\underline{1}$			
6	$17 \times 2^5 + 129$	$\boxed{1}010\boxed{1}00001$					
7	$513 + 5 \times 2^5$	$\boxed{1}010100000\boxed{1}$					
8	$145 \times 2 + 383$	$S_2 = 383$					
9	$33 \times 2^5 - 383$	$S_2 = 383$					
10	$7 \times 2^5 + 449$	$S_2 = 449$					
11	$-255 + 449 \times 2$	$S_2 = 449$					
12	$145 + 33 \times 2^4$	$S_2 = 33$					

To accelerate computation, the search space is pregenerated and stored in the hard disk for coefficients up to 13-bits (i.e., 8192). Coefficients that are not precomputed are included in the search space during execution and saved for future use. The size of our saved data measures about 10MB in Matlab 7.

B. Expanded search space based on shifted sum or difference of coefficients (SSD)

An expanded solution space of the MSD representations, called the shifted sum or difference of coefficients (SSD), is proposed to exploit new relationship between sub-expressions. The following constraints are applied to (1) in order to limit the search space to a finite set:

1. S_2 must be a coefficient in *Oset*;
2. #nzbit of n − #nzbit of $S_1 \geqslant 1$;
3. #nzbit of n − #nzbit of $S_2 \geqslant 1$.

Since (1) restricts n, S_1, and S_2 to be odd, one of the $S_1 \times 2^p$ or $S_2 \times 2^q$ must be odd and the other even. Subsequently, (1) can be re-expressed as

$$\begin{aligned}
S_1 &= (n + S_2) \times 2^{-p}; && \text{or} \\
S_1 &= (n - S_2) \times 2^{-p}; && \text{or} \\
S_1 &= (-n + S_2) \times 2^{-p}; && \text{or} \\
S_1 &= n + S_2 \times 2^q; && \text{or} \\
S_1 &= n - S_2 \times 2^q; && \text{or} \\
S_1 &= -n + S_2 \times 2^q. && \qquad (2)
\end{aligned}$$

Since both n and S_2 are in *Oset*, decompositions that satisfy (1) are determined based on the shifted sum (difference) of coefficients (n and S_2). Due to the additional constraints imposed on (1), the number of feasible solutions in (2) is finite. Linear search is then used to search for all feasible decompositions from (2).

The search space is expanded on top of the search space of MSD representations. As a result, the search space of the MSD representations is a subset of our proposed SSD search space.

Coefficients in *Oset* are first copied to *Cset*. A set of decomposition tables using one of the search spaces described in Section II is computed (e.g., Table I). Subexpressions in the lookup tables are copied into *Cset* if they are not already there. If a lookup table LT_n has not been created for the coefficient n, then n is decomposed using the MSD representations. This process continues recursively until all coefficients have been copied into *Cset* and LT_n is created for all $n \in Cset$. Considering the search space of SSD in Section IIB, when $Oset = \{673, 383, 449, 33\}$, the resultant *Cset* is {3, 5, 7, 17, 21, 31, 33, 63, 127, 129, 145, 161, 225, 255, 257, 383, 449, 511, 513, 545, 641, 673}. LT_3, LT_5, $LT_7, \cdots$, LT_{673} are formed for all $n \in Cset$.

Note that decompositions with the same pair of subexpressions are inserted into the lookup table once only. This is because the adder cost with the same pair of coefficients is the same. Decompositions from rows 1 to 12, 1 to 5, and 1 to 3 in Table I shows the proposed search space for coefficients 673, 383, and 449, respectively. The "Search Space" columns show different digit patterns extracted from the MSD representations as well as the value of S_2 when evaluating the corresponding decomposition.

III. PROBLEM FORMULATION

The multiplierless synthesis problem is formulated into a MILP formulation with a mimimum adder cost objective function and two group of constraints: constraints showing the relationship between subexpressions and the delay-time constraints, which is solved by a MILP solver.

A. Minimal adder number objective function

To minimize the adder cost (therefore, area and power consumption), a binary variable C_n is defined for each coefficient $n \in Cset$. As a result, the total number of binary variables equals to the number of coefficients in *Cset*. $C_n = 1$ indicates the coefficient n is synthesized. Otherwise, $C_n = 0$ and the coefficient n would not be synthesized in the MCM block. Since an adder is needed for every synthesized coefficients, the objective function minimizing the number of adders for the proposed 0-1 MILP reads:

$$\text{minimize} \sum_{\forall n \in Cset} C_n, \tag{3}$$

subject to

$$C_n \in \{0,1\}. \qquad \forall n \in Cset. \tag{4}$$

$$C_m = 1, \qquad \forall m \in Oset, \tag{5}$$

Here (5) forces the 0-1 MILP solver to synthesize all coefficients in $Oset$ for the output coefficients.

B. Decomposition expression constraints

When the #nzbit = 2 for a coefficient n (e.g., coefficients 3, 5, 7, 17, 31, 33, 63, 127, 129, 255, 257, 511, 513), it can be synthesized with one adder directly from the summation of two different shifted versions of MCM block input. When the #nzbit > 2 for a coefficient n, the coefficient must be synthesized from one of the decompositions in LT_n. These decomposition expressions have to be included as constraints in the formulation. For instance, (6) shows the constraints for synthesizing the coefficient 673 in the SSD search space:

$$\begin{aligned} C_{673} \leq C_{161} + C_{545} + C_{641} + C_{21} + \\ \text{Min}\{C_5, C_{33}\} + \text{Min}\{C_{17}, C_{129}\} \\ \text{Min}\{C_5, C_{513}\} + \text{Min}\{C_{145}, C_{383}\} \\ \text{Min}\{C_{33}, C_{383}\} + \text{Min}\{C_7, C_{449}\} \\ \text{Min}\{C_{225}, C_{449}\} + \text{Min}\{C_{33}, C_{145}\}. \end{aligned} \tag{6}$$

We note that each of the minimization terms, $\text{Min}\{C_a, C_b\}$ is a variable that equal to 1 when both C_a and C_b are 1, otherwise, $\text{Min}\{C_a, C_b\} = 0$. This relation is enforced by four linear inequality constraints.

There are 12 terms in (6) and each corresponds to a possible decomposition in LT_{673}. For example, the first term "C_{161}" indicates the coefficient 673 may be synthesized with one adder if the coefficient 161 exists in the MCM block (as shown in LT_{673} row 1). Similarly, the second term "C_{545}" indicates the coefficient 673 may be synthesized with one adder if the coefficient 545 exists in the MCM block (as shown in LT_{673} row 2). The fifth term in (6) is a minimization term: $\text{Min}\{C_5, C_{33}\} = 1$ if both C_5 and C_{33} equal one, otherwise $\text{Min}\{C_5, C_{33}\} = 0$. It indicates that the coefficient 673 may be synthesized with one adder if both of the coefficients 5 and 33 exist in the MCM block (as shown in LT_{673} row 5). Similar relationships apply to the 6th to 12th term in (6) from all different decompositions in LT_{673}. Constraints similar to (6) are required for all lookup tables LT_n where $L_n > 1$ (e.g., coefficients 21, 145, 161, 225, 383, 449, 545, 641, 673 in $Cset$).

C. Maximum logic depth (D_{max}) constraints

Both area and delay is considered in high-speed system design. Delay is related to the VLSI implementation (e.g. routing), so a generic definition, logic depth is used. In MCM design, same coefficients can be synthesized with different logic depth (shown in Fig. 3, $\underline{1}$ represents -1), so an optional parameter D_{max} is proposed to restrict the maximum logic depth of the MCM block as a delay constraint. If D_{max} is absent, the 0-1 MILP solver minimizes the problem without the logic depth restriction. If D_{max} is provided, constraints described in this section are applied on top of the 0-1 MILP problem in Section IIIA & B.

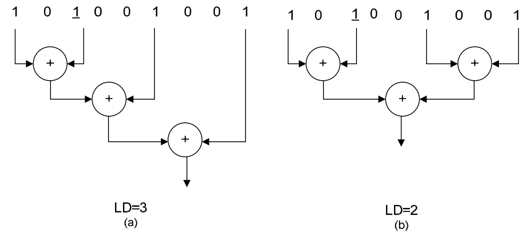

Fig. 3. Example of synthesizing a coefficient 201: (a) Logic depth = 3. (b) Logic depth = 2.

To form the set of constraints, a continuous variable D_n is defined for each coefficient $n \in Cset$. D_n is the logic depth of n if n is synthesized in the MCM block, i.e., when $C_n = 1$. Otherwise, when $C_n = 0$, D_n is set to the default value of D_{max}. This is implemented by the two constraints below. When $LD_{max} \geq 2$ and #nzbit of $n = 2$, the following constraint is applied:

$$D_n = LD_{max}(1 - C_n) + C_n. \tag{7}$$

If $C_n = 1$, (7) forces D_n to one. Otherwise, $D_n = LD_{max}$.

When $LD_{max} \geq 2$ and #nzbit of $n > 2$, the following constraints are applied:

$$LD_{max}(1 - C_n) + 2C_n \leq D_n \leq \quad LD_{max}. \tag{8}$$

If $C_n = 1$, (8) renders $2 \leq D_n \leq LD_{max}$. Otherwise, $D_n = LD_{max}$. The involvement of continuous variables converts the problem from a 0-1 PILP to a 0-1 MILP.

Assuming the pair of subexpressions S_1 and S_2 are used to synthesize the coefficient n, D_n must be equal to $\max\{D_{S_1}, D_{S_2}\} + 1$ because an adder is used to synthesize n at the output of S_1 and S_2. In other words, since n is synthesized from the decomposition in LT_n, the coefficients that are used to synthesize n must have their logic depths less than D_n. For coefficient 673, the following shows the required constraints produced from LT_{673}:

$$\begin{aligned} & C_{673} \leq (LD_{\max} - D_{161}) + (LD_{\max} - D_{545}) \\ & + (LD_{\max} - D_{641}) + (LD_{\max} - D_{21}) \\ & + (LD_{\max} - Max\{D_5, D_{33}\}) + (LD_{\max} - Max\{D_{17}, D_{129}\}) \\ & + (LD_{\max} - Max\{D_5, D_{513}\}) + (LD_{\max} - Max\{D_{33}, D_{145}\}) \\ & + (LD_{\max} - Max\{D_{145}, D_{383}\}) + (LD_{\max} - Max\{D_{33}, D_{383}\}) \\ & + (LD_{\max} - Max\{D_7, D_{44}\}) + (LD_{\max}.Max\{D_{225}, D_{449}\}). \end{aligned} \tag{9}$$

There are 12 terms (each term is in bracket) in (9) for the 12 possible decomposition in Table I. If C_{673} is one, at least one of the 12 terms has to be greater or equal to 1. For example, if the first term $(LD_{max} - D_{161})$ is greater or equal to 1, it indicates the coefficient 673 may be synthesized with the logic depth of $D_{161} + 1$ (from LT_{673} row 1) and $D_{161} + 1 \leq LD_{max}$. The fifth term in (9) is a maximization term, $\text{Max}\{D_5, D_{33}\}$. It indicates that the coefficient 673 may be synthesized with the

TABLE II
Result Comparison.

	Filter Specification				$\text{Proposed}_{\text{MSD}}$			$[8]_{\text{MSD}}$			$\text{Proposed}_{\text{SSD}}$			$\text{Proposed}^{*}_{\text{SSD}}$		
Filter	Pass	Stop	#tap	width	LO	LD	time/s	LO	LD	time/s	LO	LD	time/s	LO	LD	time/s
1	0.20	0.25	120	8	10	2	0.89	10	2	2.38	10	2	1.39	10	2	2.38
2	0.10	0.25	100	10	18	3	1.49	18	3	4.69	17	3	4.89	17	3	9.21
3	0.15	0.25	40	12	16	3	2.60	16	3	8.80	15	4	5.50	16	3	13.83
4	0.20	0.25	80	12	29	3	3.90	29	3	12.80	28	3	12.92	28	3	23.51
5	0.24	0.25	120	12	34	3	2.50	34	3	8.30	34	3	13.80	34	3	26.90
6	0.15	0.25	60	14	22	4	9.72	22	4	33.03	21	4	25.30	22	3	86.00
7	0.15	0.20	60	14	34	3	4.18	34	3	20.08	30	3	15.80	30	3	37.49
8	0.10	0.15	60	14	33	3	9.38	33	3	156.82	30	4	17.74	31	3	92.13

TABLE III
Complexity Comparison.

	$\text{Proposed}_{\text{MSD}}$			$[8]_{\text{MSD}}$			$\text{Proposed}_{\text{SSD}}$			$\text{Proposed}^{*}_{\text{SSD}}$		
Filter	#bvar	#cvar	#const.	#bvar	#cvar	#const.	#bvar	#cvar	#const.	#bvar	#cvar	#const.
1	36	19	119	383	0	732	51	71	354	51	193	833
2	71	69	430	818	0	1683	157	309	2130	157	775	4289
3	112	153	949	1443	0	3172	181	323	2440	181	872	4924
4	143	227	1639	2028	0	4694	271	812	6078	271	1895	11098
5	116	135	934	1433	0	3104	370	812	6797	370	1994	12854
6	222	710	4437	3768	0	9522	470	1604	12406	470	3678	21848
7	175	230	1540	2254	0	4979	439	938	7807	439	2315	14840
8	243	419	3268	3568	0	8426	433	1087	8696	433	2607	16122

logic depth of $\text{Max}\{D_5, D_{33}\} + 1$ (from LT_{673} row 5) and $\text{Max}\{D_5, D_{33}\} + 1 \leq LD_{max}$. Similar relationships apply to (9) from each decomposition in LT_{673}. Constraints similar to (9) are required for all lookup tables LT_n of which there are more than one possible decompositions in it (e.g., coefficients 21, 145, 161, 225, 383, 449, 545, 641, 673 in $Cset$).

D_n equals to two if and only if both D_{S_1} and D_{S_2} equal to one (i.e., $D_n \leq 2$ if and only if #nzbit of $S_1 \leq 2$ and #nzbit of $S_2 \leq 2$).

Minimum logic depth is also desired on the top of minimum adder cost, when $LD_{max} = 3$ and #nzbit of $n > 2$, the following shows the required constraints for LT_{673}:

$$D_{673} \leq LD_{max}(2 - C_{673}) - \text{Min}\{C_5, C_{33}\}, \tag{10}$$

$$D_{673} \leq LD_{max}(2 - C_{673}) - \text{Min}\{C_{17}, C_{129}\}, \tag{11}$$

$$D_{673} \leq LD_{max}(2 - C_{673}) - \text{Min}\{C_5, C_{513}\}, \tag{12}$$

$$D_{673} \geq LD_{max}(1 - \text{Min}\{C_5, C_{33}\} - \text{Min}\{C_{17}, C_{129}\} - \text{Min}\{C_5, C_{513}\}). \tag{13}$$

Equations (10) to (12) are produced from all possible decompositions in LT_{673} that may result in $D_{673} = 2$. Note that the #nzbit of {5, 17, 33, 129, 513} is two. Equations (10), (11), (12), and (13) force $D_{673} = 2$ if the coefficient 673 can be synthesized from coefficients that have their logic depth equal to one (combination of 5 and 33, 17 and 129, 5 and 513). Otherwise, (8) and (13) force $D_n = LD_{max}$. Constraints similar to (8), (10), (11) and (12) are required for all coefficients that have their #nzbit greater than two (e.g., coefficients 21, 145, 161, 225, 383, 449, 545, 641, 673 in $Cset$).

Finally, the relationship between subexpressions and the delay-time constraints are converted into variables and constraints of a MILP problem, which is then solved by a generic MILP solver.

IV. Results

We first demonstrate our result using a simple MCM synthesis. Fig. 2 shows the synthesis of the required $Oset = \{33, 383, 449, 673\}$ using the SSD search space and the proposed algorithm. Six adders are needed for synthesis. Note that the coefficient 673 is synthesized as $673 = 33 \times 2^5 - 383$ (row 9 in LT_{673}) which is not in the search space of the MSD representations. If the search space of the MSD representations is used, the MCM block requires seven adders. Consequently, 14% adder reduction is achieved by the proposed SSD search space.

More fixed-point filter examples are used to compare the improvement of the search space. To begin with, some notational conventions are specified. In Table II, "#tap" stands for the filter length or the number of multiplications of the MCM block and "width" is the maximum binary bit size of coefficients. The subscripts "MSD" and "SSD" indicate the search spaces. "LO" is the number of logic operations which equals the total number of adders and subtractors. "LD" is the logic depth and "time" is the solution time of the solver in seconds. The asterisk "*" indicates that the maximum logic depth restriction D_{max} is imposed.

The specifications and synthesized results of Filter 1 to Filter 8 are obtained from [8] and summarized in Table II. It is shown that the proposed search space of SSD further reduces the size of the MCM block in most cases. When comparing the adder cost between the search spaces of SSD and MSD, 6%, 3%, 5%, 12%, and 9% adder reductions are observed in filters 3, 4, 6, 7, and 8, respectively. This verifies the merits of SSD over MSD in getting more hardware-efficient filters.

The complexity performance of the proposed optimization framework is also investigated and compared to [8]. To begin with, some notational conventions are specified. In Table III, "#bvar" and "#cvar" are the numbers of binary and continuous variables, respectively, and "#const." denotes the number of constraints. The 0-1 PILP models of [8] are constructed and solved with [9] in the same experimental condition. Our experiments are run on a 1.5 GHz computer with 768MB memory.

The total solution time of synthesizing all eight filters is 34.66 seconds in our scheme while it takes 246.90 seconds for that in [8], or in other words, a 7X speedup. Instead of modeling the MCM problems into a Boolean network in [8], the problems are directly formulated as 0-1 MILP problems in our proposed method. As a result, the redundancy variables in [8] are greatly reduced. All of the filter examples shows that the proposed models have fewer variables and constraints than those in [8]: on average, our proposed models have 82% fewer variables, and 68% fewer constraints. Take filter 8 as an example, 243 binary variables and 419 continuous variables are defined in our proposed model, while 3568 binary variables are needed in [8]. Our reduced model therefore results in a significant improvement of solution time. In particular, the same problem is solved 17X faster than that in [8]. It shows that the proposed algorithm is a good MCM synthesizer for large search space.

The trade-off between adder cost (circuit area) and maximum logic depths (delay-time constraint) is also investigated. When the maximum logic depth is set to three ($LD_{max} = 3$), the adder costs in filters 3, 6, and 8 are increased slightly comparing to the unlimited case, but the logic depth is reduced by one-third, which allows faster operation. The most complicated problem in our design examples is to synthesize filter 8 in the search space of SSD when the maximum logic depth is set to 3. The problem contains 433 binary variables, 2607 continuous variables, and 16122 linear constraints. The resultant adders cost and logic depth are 31 and 3, respectively. The solution time of the 0-1 MILP problem is 92.13 seconds.

V. Conclusion

We have proposed a novel CSE algorithm that models the synthesis of MCM into a 0-1 MILP problem, and performed fixed-point arithmetic optimization. The formulated problem is solved efficiently in practically sized examples, with consideration of time-delay. In our test examples, our method solves the problems up to 17X faster than that in [8], which shows the capability of the algorithm to work with large search spaces. Our CSE algorithm obtained the global optimum in the SSD search space, which is currently the largest search space for global optimization in multiplierless MCM design. However, we note that better solutions may still exist outside the SSD search space. In our on-going work, we would develop CSE algorithms that cover the entire search space of all-digit representations [6]. The CSE algorithm may also be modified to optimize the die area [11] instead of the adder cost of the MCM.

Acknowledgment

The authors would like to thank Prof. P. Flores for sharing the filter coefficients. This work was supported in part by the Hong Kong Research Grants Council and the University Research Committee of The University of Hong Kong.

References

[1] 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. Computer-Aided Design*, vol. 24, no. 2, pp. 295–304, Feb. 2005.

[2] T.-S. Chang, J.-I. Guo, and C.-W. Jen, "A compact IDCT processor for HDTV applications," in *IEEE Workshop on Signal Processing Systems (SiPS)*, 1999, pp. 151–158.

[3] 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*, vol. 51, no. 11, pp. 2215–2221, Nov. 2004.

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

[5] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on an algorithm to generate all minimal signed digit representations," *IEEE Trans. Computer-Aided Design*, vol. 21, no. 12, pp. 1525–1529, Dec. 2002.

[6] A. G. Dempster and M. D. Macleod, "Digital filter design using subexpression eliminiation and all signed-digit representations," in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 3, May23–26, 2004, pp. III 169–III 172.

[7] M. D. Macleod and A. G. Dempster, "Multiplierless FIR filter design algorithms," *IEEE Signal Processing Lett.*, vol. 12, no. 3, pp. 186–189, Mar. 2005.

[8] P. Flores, J. Monteiro, and E. Costa, "An exact algorithm for the maximal sharing of partial terms in multiple constant multiplications," in *Proc. IEEE Intl. Conf. Computer-Aided Design*, San Jose, CA, Nov. 6–10, 2005, pp. 13–16.

[9] L. Schrage, *Optimization Modeling with LINGO.* 1415 North Dayton Street, Chicago, Illinois 60622: LINDO Systems, Inc., 2006.

[10] A. Dempster and M. Macled, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II*, vol. 42, no. 9, pp. 569–577, 1995.

[11] D. L. Maskell and J. Liewo, "Hardware-efficient FIR filters with reduced adder step," *IEE Electronics Letters*, vol. 41, no. 22, pp. 1211– 1213, Oct. 2005.