Tap delay-and-accumulate cost aware coefficient synthesis algorithm for the design of area-power efficient fir filters

Qiao, Rui; Faust, Mathias; Chen, Jiajia; Chang, Chip-Hong; Ding, Jiatao

2018

Chen, J., Chang, C.-H., Ding, J., Qiao, R., & Faust, M. (2018). Tap delay-and-accumulate cost aware coefficient synthesis algorithm for the design of area-power efficient fir filters. IEEE Transactions on Circuits and Systems I: Regular Papers, 65(2), 712-722.
doi:10.1109/TCSI.2017.2725916

<https://hdl.handle.net/10356/105545>

<https://doi.org/10.1109/TCSI.2017.2725916>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/TCSI.2017.2725916>

Downloaded on 26 Apr 2023 22:25:41 SGT

Tap Delay-and-Accumulate Cost Aware Coefficient Synthesis Algorithm for the Design of Area-Power Efficient FIR Filters

Jiajia Chen, Chip-Hong Chang, *Senior Member, IEEE*, Jiatao Ding, Rui Qiao and Mathias Faust

Abstract—Finite impulse response filters are widely used in digital signal processing applications. Prodigious research in the past two decades has substantially reduced the implementation cost of the multiple constant multiplication block. Further area and power consumption savings are stagnated by the structural adders and registers in the tap delay-and-accumulate line, which unfortunately dominate the overall hardware cost of FIR filter and are difficult to minimize by existing resource sharing approaches. Retiming or relocating the structural adders and registers can improve merely the throughput. To close the area-power efficiency gap, we reformulate the filter coefficient synthesis problem to explore the design space for the tap delay-and-accumulate line by bisecting at some tap position. An efficient Genetic Algorithm is proposed to solve this integer programming problem at quadratic computational complexity by refining the search space for finding an optimized solution to fulfill the frequency response specifications. FPGA and ASIC logic synthesis results from twelve benchmark filter specifications showed that the average area and power consumptions of the solutions generated by our proposed algorithm have been reduced by up to 26.8% and 27.5% respectively, in comparison with the solutions obtained by existing design methods.

Index Terms — FIR Filter Design, Structural Adder, Digital Signal Processing.

I. INTRODUCTION

FINITE Impulse Response (FIR) filter for signal conditioning is commonly realized in digital domain owing to its stability and linear phase response. The research in low complexity FIR filter design methodologies continues to thrive as footprint and power budget on smart devices are increasingly squeezed by technology revolution such as the upcoming 5G communication and recent proliferation of augmented and virtual reality multimedia applications. The emerging Internet of Things (IoT) [1] entails the digitization and interconnection of ubiquitous sensors and distributed

nodes with central computing resource. Industry 4.0 also mandates integration of heterogeneous sensors into intelligent sensor network to provide fully automated control. Appropriate signal conditioning circuits are needed to remove the undesired noises and interferences [2]–[5]. As application-specific digital filters play influential role in the SNR, response time, robustness and mobility of sensor network, the stagnation in their size, power, sensitivity and accuracy will impact the techno-optimism of IoT and the related technologies [6].

Existing algorithms [7]–[13] for the design of low complexity FIR filter mostly minimize the multiple constant multiplication (MCM) block and neglect the complexity of structural adders and registers in the tap delay-and-accumulate (TDA) block of FIR filter. The main reason is the operands of the structural adders are derived from different time-delayed input samples. Unlike the MCM block, there is neither exploitable correlation nor apparent redundancy to enable resource sharing by common subexpression elimination. Unfortunately, these structural adders are more expensive and power hungry than the sharable adders in the optimized MCM block, which makes them the hindrance to further area and power reduction of FIR filter implementation. Some recent algorithms, such as [14] and [15], investigated this problem and modified the structure of the TDA block to reduce the overall complexity or increase the throughput of the filter. In [14], splitting and remerging of operands to selected structural adders is used to control the progressive growth in the sizes of structural adders and registers in different segments of TDA block. In [15], pipeline registers are introduced to retime the critical path and increase the throughput of the filters. The effectiveness of these methods is still largely limited by the pre-determined filter coefficient set. The area and power consumption of the filter designed by [15] can and are likely to increase due to the additional pipeline registers. This is because the primary objective of this method is to reduce the filter delay at the expense of reasonable hardware overhead.

Taking cognizance of the limitation imposed by fixed filter coefficients on the splitting of structural adders for complexity reduction, we propose a new design algorithm to unleash the restriction of TDA block optimization. Instead of retiming the critical adder paths or synthesizing filter coefficients to maximize sharing of common subexpressions in MCM block without compromising the magnitude response specification, the finite-precision filter coefficients are synthesized to find an optimal splitting point to achieve the maximum effective cost saving of TDA block with a simple split-once technique. The

Manuscript received 9 May, 2017; Revised 7 July, 2017; Accepted 7 July, 2017.

The authors Jiajia Chen, Jiatao Ding and Rui Qiao are with Singapore University of Technology and Design, Singapore. (e-mail: jiajia_chen@sutd.edu.sg)

The author Chip-Hong Chang (corresponding author) is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. (e-mail: echchang@ntu.edu.sg)

The author Mathias Faust is with mfnet gmbh, Switzerland. (e-mail: mfaust@mfnet.ch)

coefficient synthesis problem is modelled as an integer programming problem and solved by a customized Genetic Algorithm (GA) [16]. To the best of our knowledge, this is the first filter coefficient synthesis algorithm that tackles the minimization of TDA block in the literature.

The rest of this paper is organized as follows. Section II provides an overview of TDA bisection technique and how the performance is affected by choosing different bisection position. Section III introduces our formulation of the filter coefficient synthesis problem and presents our customized GA solver to this problem. Section IV presents and compares the synthesis results of a group of benchmark filters, generated by our proposed algorithm and three other competing algorithms. Lastly, our conclusion is provided in Section V.

II. PRELIMINARIES AND MOTIVATION

A. Complexity of MCM and TDA blocks in FIR filter

An N -tap finite impulse response (FIR) digital filter can be realized from the following discrete time convolution.

$$y[n] = h[n] * x[n] = \sum_{i=0}^{N-1} h[i]x[n-i] \quad (1)$$

where $x[n]$ and $y[n]$ are the n -th time domain input and output data samples, respectively. $h[i]$, $i = 0, 1, \dots, N-1$, are the filter coefficients of the impulse response function

$$H(z) = \sum_{i=0}^{N-1} h[i]z^{-i}.$$

Fig. 1 shows the transposed direct form implementation of a FIR filter. For linear-phase FIR filter, $h[n] = h[N-1-n]$ and

$$H(z) = \begin{cases} h\left(\frac{N-1}{2}\right)z^{-\frac{N-1}{2}} + \sum_{i=0}^{\frac{N-1}{2}-1} h(i)(z^{-i} + z^{-N-1-i}) & \text{odd } N \\ \sum_{i=0}^{\frac{N}{2}-1} h(i)(z^{-i} + z^{-N-1-i}) & \text{even } N \end{cases} \quad (2)$$

Due to the coefficient symmetry, only $(N-1)/2$ distinct coefficients need to be computed within the MCM block of Fig. 1, but the numbers of structural adders (SAs) and registers required to sum the delayed partial sums (which are the product of the input sample $x(n)$ and filter coefficient $h(i)$) in the cascaded TDA line remain unchanged. Unfortunately, TDA block uses more than double the hardware resources of MCM block, as evidenced by the number of full adders (#FA), full subtractors (#FS) and flip-flops (#FF) used by the MCM and TDA blocks of four FIR filters [7] [18]-[20] that are widely

Fig. 1. Transposed direct form FIR filter with structural adders A_0 to A_{N-2} and registers R_0 to R_{N-2}

used for benchmarking MCM minimization algorithms in Table I. The filter tap N , coefficient wordlength w and coefficients of these filters are obtained by Park-McIlean algorithm [17] and the MCM blocks are optimized using Pasko algorithm [7]. 8-bit input variable x is assumed for all filters. The ratio of TDA area to MCM area is shown in the last row of Table I by assuming that the area of a FF is comparable to that of a FA. It is apparent that it takes only a small fraction of TDA area to outweigh any area savings derived from the MCM block minimization. For the case of FIR2, a trivial 5.5% saving from TDA block is more than sufficient to cover the area of the entire MCM block. This is because the SAs in TDA block are much larger than the adders and subtractors in MCM. Unfortunately, as the input signals to each SA are derived from the input signal sampled at different clock cycles, the addends are largely uncorrelated. Hence, the SAs cannot be reduced by common subexpression elimination as in MCM block.

TABLE I. Comparison of hardware resources between MCM and TDA blocks

Filters	FIR1 [18]	FIR2 [19]	FIR3 [20]	FIR4 [7]
N	24	37	43	121
w	12	8	16	16
MCM	#FA	129	21	323
	#FS	220	36	226
TDA	#FA	461	528	861
	#FF	459	523	853
TDA/MCM	2.6	18.4	3.1	4.5

As the impulse response is a sinc-like function, the outputs of MCM block have a $1/x$ envelope. As each addend is the product of a constant and a fixed-width integer variable x , the inputs to the SA are discrete random integer variables. Thus, the number of distinct integer output values is far lower than $2^{w_x N}$ for the accumulation of N continuous random integer variables, where w_x is the word length of the input variable x . For example, the 121-tap FIR filter [19] with 8-bit input x has less than $2^{25} \approx 3.3 \times 10^7$ possible output values as opposed to $256^{121} \approx 2.5 \times 10^{291}$ different output values. It has been shown in [21] that central limit theorem holds for the probability distribution of SA output values. As the probability of recurrent integer outputs increases after some number of accumulations, the bit width required to represent the output of a SA grows towards the output of the filter till the middle tap and remains relatively constant from the middle tap to the output. The probability of the output values of a fixed-coefficient FIR filter can thus be approximated by a normal distribution. In fact, the tail values of the actual distribution are even less likely to occur than those at the tails of the normal distribution. This means that the range of SA outputs in the later half of the TDA can be accurately estimated from w_x and the predetermined filter coefficient values to reduce the lengths of the corresponding SAs.

B. SA reduction by bisection of long operand

Let p_i and q_i be the two inputs to the i -th SA, denoted by A_i , where p_i is the output of the i -th register R_i of the TDA block and q_i is the i -th coefficient multiplier output $h(i) \cdot x[n]$ of MCM block in Fig. 1. Let $w(\alpha)$ denote the word length of a signal variable α . In [14], the sizes of some SAs are reduced by bisecting a selected long partial sum p_j at the j -th tap into two parts p_j^u and p_j^l as shown in Fig. 2. By saving the upper part

p_j^u for m ($1 \leq m \leq j$) delays, the lower part p_j^l can be accumulated to q_i by reduced word length SAs, A_i for $i = j, j-1, \dots, j-m$. After $m-1$ cycle delays, the saved p_j^u is added to the signed extended p_{j-m-1}^l where $w(p_{j-m-1}^l) > w(p_j^l)$. If $m = j$, then p_{-1} represents the output of the last SA A_0 . The bit width $w(A_i)$ of each reduced SA A_i , for $i = j, j-1, \dots, j-m$, can be estimated by interval arithmetic as follows:

$$w(A_i) > \left\lceil \log_2 \left(\max \left(2^{w(p_i^l)} - 1 + \sum_{i=j-m}^j v_i^+, -\sum_{i=j-m}^j v_i^- \right) \right) \right\rceil \quad (3)$$

where $\lceil \cdot \rceil$ is the ceiling function and $[v_j^+, v_j^-]$ is the dynamic range of q_j .

Fig. 2. Bisection of partial sum p_j before SA A_j

Fig. 3. TDA block with bisection technique at SA A_j with $m = j$

As $v_i^- \leq 0$ and $v_i^+ \geq 0$, $w(A_i)$ increases monotonically with i . Assuming ripple carry adder (RCA) is used for the SA implementation to conserve area and power, the total number of FAs saved by the m reduced SAs starting from the j -th tap is given by:

$$\Delta_{j,m} = \sum_{i=0}^m \left[w(p_j) - w(A_{j-i}) - \rho(w(A_{j-i}) - w(p_j^l)) \right] - \left[w(A_{j-m}) - w(p_j^l) \right] \quad (4)$$

In (3), $w(p_j) - w(A_{j-i})$ represents the FA cost saving and $w(A_{j-i}) - w(p_j^l)$ represents the register overhead required to save p_j^u for m cycles, where ρ is the area ratio of a FA to a FF. The last term $w(A_{j-m}) - w(p_j^l)$ refers to the FA cost required to merge the saved p_j^u .

Although $w(p_j^l)$ can be any value smaller than $w(p_j)$, to minimize the SAs, $w(p_j^l)$ is made equal to $w(q_j)$.

Apparently, $\Delta_{j,m}$ depends on the tap position j to split p_j and the number of delay cycles, which is $m-1$, before the merging. To investigate the effect of maximizing the SA cost saved by performing only one splitting of addend at some tap position j with $m = j$ such that all the SAs from the j -th tap to the output can be reduced, as shown in Fig. 3. Table II lists the values of $\Delta_j = \Delta_{j,j}$ at different splitting position j for FIR1 to FIR4 with $\rho = 1$. The first row of Table II with $j = -1$ represents the equivalent FA cost of the TDA block without applying the bisection method. It shows that the register overhead incurred by the bisection increases with m and outweighs the saving in SAs if the cost of a FF is comparable to a FA.

TABLE II. FA cost and FA saving Δ_j by direct application of TDA bisection technique at different j for FIR1 to FIR4

FIR1 ($N=25$)			FIR2 ($N=37$)			FIR3 ($N=43$)			FIR4 ($N=121$)		
j	FA	Δ_j	j	FA	Δ_j	j	FA	Δ_j	j	FA	Δ_j
-1	920	0	-1	1051	0	-1	1714	0	-1	5145	0
12	1160	-240	22	1355	-304	21	2176	-462	60	6585	-1440
11	1140	-220	20	1351	-300	19	2132	-418	55	6244	-1099
10	1100	-180	18	1339	-288	17	2018	-304	50	6045	-900
9	1064	-144	16	1307	-256	15	1984	-270	45	5955	-810
8	1047	-127	14	1215	-164	13	1918	-204	40	5785	-640
7	1032	-112	12	1195	-144	11	1890	-176	35	5705	-560
6	1011	-91	10	1131	-80	9	1837	-123	30	5565	-420
5	994	-74	8	1097	-46	7	1812	-98	25	5445	-300
4	983	-63	6	1083	-32	5	1774	-60	20	5381	-236
3	968	-48	4	1075	-24	3	1750	-36	15	5291	-146
2	949	-29	2	1057	-6	1	1724	-10	10	5224	-79
1	935	-15	-	-	-	-	-	-	5	5173	-28

The negative Δ_j results of Table II signify that limited savings can be obtained even if different combinations of j and m are explored simultaneously with multiple bisections and merges. The trade-off between the register overhead and SA savings is severely constrained by the pre-determined filter coefficients. In cognizant of the influence of filter coefficients on the word length distribution of q_i and the dominance of TDA block area on the overall filter complexity, we propose to synthesize the FIR filter coefficients to achieve notable TDA block minimization by simple bisection with $j = m$.

III. TDA MINIMIZATION ALGORITHM BY FILTER COEFFICIENT SYNTHESIS

A. FIR Filter Coefficients and Frequency Response

The frequency response of a linear-phase FIR filter can be described by a real-valued zero-phase frequency response function $H_R(\omega)$ and a real-valued phase function $\theta(\omega)$, i.e.,

$$H(\omega) = H_R(\omega) e^{j\theta(\omega)} \quad (5)$$

where ω is the angular frequency in radian per second.

Since $|e^{j\theta(\omega)}| = 0$, $|H(\omega)| = |H_R(\omega)|$. Ignoring the linear phase term, $e^{-j\omega(N-1)/2}$, the frequency response of a FIR filter of length N can be expressed as [22]:

$$H_R(\omega) = a_0 + 2 \sum_{i=1}^{\lfloor N/2 \rfloor} a_i T(\omega, i) \quad (6)$$

where a_i is a function of the filter coefficient $h(i)$, $T(\omega, i)$ is a

trigonometric function.

For a linear phase FIR filter, $T(\omega, i) = \cos(i\omega)$. If N is odd, then $a_0 = h\left(\frac{N-1}{2}\right)$ and $a_i = h\left(\frac{N-1}{2} - i\right)$. If N is even, then $a_0 = 0$ and $a_i = h\left(\frac{N}{2} - i\right)$. Thus, the magnitude response of $H(\omega)$ can also be expressed as:

$$H(\omega) = \begin{cases} h\left(\frac{N-1}{2}\right) + 2 \sum_{i=0}^{\frac{N-1}{2}-1} h(i) \cos\left(\left(\frac{N-1}{2} - i\right)\omega\right) & N \text{ is odd} \\ 2 \sum_{i=0}^{\frac{N}{2}-1} h(i) \cos\left(\left(\frac{N}{2} - i\right)\omega\right) & N \text{ is even} \end{cases} \quad (7)$$

For fixed point implementation, the real-valued coefficients $h(i)$ for $i = 1, 2, \dots, N$ can be directly truncated into a set of L -bit finite precision coefficients denoted by $\hat{H}$.

Different finite precision coefficient sets H' can be further derived from $\hat{H}$ to meet the minimum filter response specifications, which are typically specified in terms of the filter passband and stopband frequencies, maximum allowable passband ripple, and minimum stopband attenuation. The tolerance of frequency response error $E(\omega) = |H(\omega) - H'(\omega)|$, where $H'(\omega)$ is the approximated magnitude response obtained by H' , is exploited by filter design algorithms to synthesize H' with a minimal number of signed power-of-two terms [19] or a maximal number of common subexpressions [8], [22]. Unfortunately, the filter coefficient set H' synthesized by all existing design algorithms is helpful in reducing the complexity of only the MCM block but not the TDA block.

B. Problem Formulation

For an input signal x of word length $w(x)$, the word length of $q_i = h'(i)x(n)$, where $h'(i)$ is the i -th coefficient of H' , is given by:

$$w(q_i) = w(h'(i)) + w(x) - 1 \quad (8)$$

From (4), if the operand p_j of SA A_j is chosen for bisection and $m = j$, then the net FA saving Δ_j of the bisection method based on the coefficient set H' can be expressed as:

$$\Delta_j = \sum_{i=0}^j \left[w(p_j) - w(A_{j-i}) - \rho(w(A_{j-i}) - w(h'(j) - w(x) + 1)) \right] - \left[w(A_0) - w(h'(j) - w(x) + 1) \right] \quad (9)$$

The passband, transition band and stopband for different filter types can be specified by the four lower and upper transition band edge frequencies, $\omega_1, \omega_2, \omega_3$ and ω_4 , as shown in Table III, where $\omega_1 < \omega_2 < \omega_3 < \omega_4$. For low pass filter, ω_3 and ω_4 represent the passband and stopband edge frequencies,

respectively, and the lower transition band edge frequencies, $\omega_1 = \omega_2 = 0$. For high pass filter, ω_1 and ω_2 represent the stopband and passband edge frequencies, respectively, and the upper transition band edge frequencies, $\omega_3 = \omega_4 = \infty$. For band pass filter, ω_1 and ω_4 are the stopband edge frequencies while ω_2 and ω_3 are the passband edge frequencies. For the band stop filter, ω_1 and ω_4 are the passband edge frequencies while ω_2 and ω_3 are the stopband edge frequencies.

TABLE III. Passband, transition band and stopband for different filter types

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

Let δ_p and δ_s denote the passband ripple and stopband attenuation, respectively. Then, to reduce the TDA block complexity, H' should be synthesized such that its magnitude response fulfills $\delta_p \leq \delta_{pmax}$ and $\delta_s \geq \delta_{smin}$ with the maximum Δ_j , where δ_{pmax} and δ_{smin} are the maximum allowable passband ripple and the minimum stopband attenuation, respectively. The FIR filter design problem can be reformulated as:

Find N, j and H' to **maximize** Δ_j s.t.

$$H'(\omega) \leq \delta_{pmax} + 1 \text{ for } \omega \in \text{passband, and} \quad (10)$$

$$H'(\omega) \leq 1 - \delta_{smin} \text{ for } \omega \in \text{stopband}$$

where $j \in [1, N-1]$ is the optimal tap position to bisect the input operand p_j to the j -th SA, A_j , where the index j is numbered from the output tap ($j = 0$) towards the input ($j = N-1$) of the filter.

The solution to this integer linear programming problem can be evolved from an initial finite precision coefficient set $\hat{H}$ of word length L and filter order N determined by the Park-McClellan algorithm based on $H(\omega)$. Although N and L of $\hat{H}$ are not large, the search space for a solution coefficient set H' that possesses an optimal point to maximize Δ_j is huge. To ensure a fast convergence to an optimal solution, two additional constraints are imposed. First, the solution set H' must maintain the linear phase response. Secondly, the dynamic range of each coefficient $h'(i)$ should not be significantly different from that of its corresponding coefficient $\hat{h}(i)$. This is accomplished by limiting their precision difference to $\pm b$ units of least precision (ulp) of $\hat{h}(i)$. For $\hat{h}(i)$ with L -bit precision, its ulp = 2^{-L} . By setting $b = 2$, a more constrained integer linear programming problem is defined:

Find N, j and H' to **maximize** Δ_j s.t.

$$h'(i) = h'(N-i-1) \quad \forall i \in [0, N-1],$$

$$h'(i) \in [\hat{h}(i) - 2^{1-L}, \hat{h}(i) + 2^{1-L}] \quad \forall i \in [0, N-1], \quad (11)$$

$$H'(\omega) \leq \delta_{pmax} + 1 \text{ for } \omega \in \text{passband and}$$

$$H'(\omega) \leq 1 - \delta_{smin} \text{ for } \omega \in \text{stopband}$$

C. Proposed Coefficient Synthesis Algorithm

The integer linear programming problem described by (11) can be solved by Genetic Algorithm (GA) [16]. By introducing stochastic search [23] to escape local minima, GA helps in converging the solution towards greater optimality uniformly from many different initial coefficient sets $\mathbf{H}'$ without demanding precise modeling of magnitude response errors.

The GA is initialized with a population of K randomly selected solutions as the parent set. For the problem of (10), each of these K solutions is a combination of a bisection point j and a coefficient set $\mathbf{H}'$. A portion of this population is then selected to breed a new generation of solutions through an iterative process of crossover and mutation operations until the termination criterion is met, which is determined by the point of diminishing return in the net positive saving Δ_j .

The solutions in the parent set are paired to perform the crossover. Selecting the two most optimal solutions directly from the parent set to crossover can hardly evade the local minima problem. If an optimal solution is paired with a non-optimal solution and crossover, an additional optimal solution is more likely to be evolved. In our method, S solutions are first randomly selected from the parent set. Two solutions with the largest Δ_j in these S randomly selected solutions are paired and the remaining $S - 2$ solutions are returned to the parent set. Next, another S unpaired solutions in the parent set are randomly selected and two of them with the largest Δ_j are paired. This pairing continues until all solutions in the parent set are paired. Let $h'_A(\alpha)$ and $h'_B(\alpha)$ be the α -th coefficient ($\alpha \in [0, N-1]$) of a pair with solutions, A and B , respectively. Crossover is performed by swapping two different bits in the same position of $h'_A(\alpha)$ and $h'_B(\alpha)$ once, starting from the least significant bit (LSB) position. The two new solutions generated by a crossover will undergo the constraint check based on (11). The qualified new solutions that meet the constraints are added into the next generation parent set and the disqualified ones are discarded. Once a successful crossover between $h'_A(\alpha)$ and $h'_B(\alpha)$ is made, the remaining bits of $h'_A(\alpha)$ and $h'_B(\alpha)$ will not be swapped. The crossover will be performed for all coefficients of a pair and for all paired solutions in the parent set of current generation.

The pairing and crossover operations are described by the pseudocodes in Fig. 4. The function **top_two_saving** selects two candidate solutions with the greatest savings from the group of S solutions. The function **swap**($Paired_grp(i), \alpha$) swaps a digit in one solution with that in another solution of the i -th candidate solution pair at the same bit position. The **check**($new_solution$) function tests if the resulting new solution meets the constraints of (11), and returns 1 if it fulfills the constraints.

Mutation is performed after crossover, where one bit of a coefficient in a solution is changed to yield a new candidate solution, starting from the LSB. In practice, the constraints of (11) are unlikely to be met when bit weighted heavier than the second LSB is varied, as verified by rigorous experimentation. Therefore, mutation is only performed on the two LSBs of a coefficient in our algorithm, which produces at most four

possible new candidate solutions. Even then, only a few big coefficients of a FIR filter can afford to have their two LSBs changed without violating the constraints of (11). To reduce the search space of our algorithm, highly probable ineffective mutations are avoided by applying the mutation operation to only the M largest coefficients of a candidate solution. Thus, the maximum number of mutations performed is $4M$ for each solution. All the newly generated qualified solutions by mutation are added into the parent set of the next generation.

```

crossover(parent_set, S, K); {
  Initialize Unpaired = K; // number of unpaired solutions
  Initialize Unpaired_grp = parent_set; // unpaired group
  Initialize Paired_grp =  $\phi$ ; // paired group
  Initialize group //group stores S selected solutions
  while (Unpaired  $\neq$  0) {
    if Unpaired  $\geq$  S,
      group = random_select(parent_set, S);
      {A, B} = top_two_saving(group);
      Unpaired_grp = remove(Unpaired_grp, A, B);
      Paired_grp = add(A, B);
      Unpaired = Unpaired - 2;
    else
      group = random_select(parent_set, Unpaired);
      {A, B} = two_top_saving(group);
      Unpaired_grp = remove(Unpaired_grp, A, B);
      Paired_grp = add(A, B);
      Unpaired = Unpaired - 2;
    end
  end
  end
  for i = 1 to K/2
    for  $\alpha$  = 0 to N-1
      new_solution = swap(Paired_grp(i),  $\alpha$ );
      if check(new_solution) = 1
        parent_set = add(new_solution);
        break;
      end
    end
  end
  end
  return parent_set;
}

```

Fig. 4. Crossover operation of proposed GA-based filter coefficient synthesis algorithm

The size of the parent set has expanded after crossover and mutation due to many more newly generated candidate solutions. The size of the parent set is reduced back to K before the next iteration by keeping only the K candidate solutions with the largest net positive savings Δ_j . The overall saving of these K new candidate solutions is expected to be greater than or equal to the overall saving of the K candidate solutions in the parent set of the previous generation. As the average cost of the candidate solutions continue to reduce from generation to generation, the effect of crossover and mutation becomes less prominent. Therefore, the iteration will terminate if the increment in average cost saving between two iterations has reduced to below certain empirical threshold, which indicates that further iteration gains significantly less savings for the same computational effort. To limit the computation time, the number of iterations is also capped at some maximum value. The GA terminates when either criterion is met. Upon termination, the GA returns the L -bit precision coefficient set $\mathbf{H}'$ with the position j of the TDA block where the bisection is applied.

Fig. 5 presents the GA program flow of the TDA cost driven coefficient synthesis. The function $\mathbf{GA}(\delta_{pmax}, \delta_{smin}, \omega_1, \omega_2, \omega_3, \omega_4, \hat{\mathbf{H}})$ takes the design specifications and initial coefficient set as

inputs and generates the optimized finite coefficient set $\mathbf{H}'$. The function **random_generate** ($\hat{\mathbf{H}}$, *constraint*, K) randomly generates K candidate solutions into the parent set, by varying j and $\hat{\mathbf{H}}$. Each solution is represented by an integer j and a coefficient set $\mathbf{H}'$ fulfilling the constraints specified in (11). The function **saving_compute**(*parent_set*) evaluates and adds up the savings of all solutions in the parent set. The functions **crossover** and **mutation** perform crossover and mutation, respectively, as described earlier, and generate new candidate solutions into the parent set of the next generation. The function **select**(K , *parent_set*) selects the K solutions with the largest savings from the parent set for the next iteration. When one of the two termination criteria is met, the solution with the largest saving is selected by **select**(1, *parent_set*) as the final solution.

```

GA( $\delta_{pmax}, \delta_{smin}, \omega_1, \omega_2, \omega_3, \omega_4, \hat{\mathbf{H}}$ ) {
  Initialize constraint; //constraints as described in (10)
  Initialize saving_increment =  $\infty$ , Num_iteration = 0;
  Initialize  $K$ ; // choose parent_set size to be 50
  Initialize  $S$ ; // group size selected for pairing
  Initialize  $T$ ; // choose minimum number of iterations needed to be 6
  Initialize  $\epsilon$ ; // choose average saving increment threshold to be  $1/K$ 
  parent_set = random_generate ( $\hat{\mathbf{H}}$ , constraint,  $K$ );
  average_saving = saving_compute(parent_set)/ $K$ ;
  while (saving_increment  $\geq \epsilon$  or Num_iteration  $< T$ ) {
    parent_set = crossover(parent_set,  $S$ ,  $K$ );
    parent_set = mutation(parent_set,  $M$ );
    parent_set = select( $K$ , parent_set);
    new_average_saving = saving_compute(parent_set) /  $K$ ;
    saving_increment = new_average_saving - average_saving;
    average_saving = new_average_saving;
    Num_iteration = Num_iteration + 1;
  }
  end while; // end iterations
  best_solution = select(1, parent_set); // the best solution
  return best_solution( $j$ ,  $\mathbf{H}'$ );
}

```

Fig. 5. GA program flow for TDA cost driven coefficient synthesis

Fig. 6 describes the complete design flow of the proposed TDA block minimization (TDAdmin) algorithm. **TDAdmin** generates a succinct implementation topology of optimized TDA and MCM blocks based on the design specification of a FIR filter. The minimum filter length N is estimated by **PM_Min_N** and the N filter coefficients of $\hat{\mathbf{H}}$ are synthesized by **PM** using Park-McLellan algorithm. The coefficient set $\mathbf{H}'$ that minimizes the TDA block cost is then generated by calling our **GA** function to solve the integer programming problem formulated in (11). The function **Design_TDA** returns the design of TDA block by applying the bisection technique at SA position j . To complete the FIR filter implementation, the MCM block is designed and optimized by **Design_MCM** using one of the existing MCM algorithms with the finite precision coefficients of $\mathbf{H}'$ generated by our GA algorithm.

```

TDAdmin( $\delta_{pmax}, \delta_{smin}, \omega_1, \omega_2, \omega_3, \omega_4$ ) {
   $N$  = PM_Min_N( $\delta_{pmax}, \delta_{smin}, \omega_1, \omega_2, \omega_3, \omega_4$ ); //minimize filter length
   $\hat{\mathbf{H}}$  = PM( $\delta_{pmax}, \delta_{smin}, \omega_1, \omega_2, \omega_3, \omega_4, N$ ); //generate initial coefficient
  ( $j$ ,  $\mathbf{H}'$ ) = GA( $\delta_{pmax}, \delta_{smin}, \omega_1, \omega_2, \omega_3, \omega_4, \hat{\mathbf{H}}$ ); //solve (11) using GA
   $TDA$  = Design_TDA( $j$ ,  $\mathbf{H}'$ );
   $MCM$  = Design_MCM( $\mathbf{H}'$ );
  return  $TDA$ ,  $MCM$ ;
}

```

Fig. 6. Proposed TDAdmin algorithm

D. Algorithm Complexity

According to the constraints of (11), the search for $\mathbf{H}'$ in our proposed GA algorithm is bounded by $\pm 2^{(-L+1)}$ of each L -bit coefficient originated from $\hat{h}(i)$ of $\hat{\mathbf{H}}$. Due to carry and borrow, at most three LSBs of a coefficient of the candidate solution can possibly be changed regardless of L . Different bisection points $j \in (1, N-1)$ are tested to evaluate the net positive saving of each candidate solution. The search for the optimal bisection point of each candidate solution grows linearly with N . In practice, crossover is limited to the two LSBs of any coefficients. In each iteration, the parent set maintains K coefficient sets to form $K/2$ pairs and each coefficient set has N coefficients. So the complexity of crossover is $O(2NK/2) = O(NK)$. Similarly, four possible mutations can be performed on each of the M largest coefficients. So the mutation complexity for K candidate solutions of the parent set is $O(4MK)$. The overall complexity of crossover and mutation for each generation is $O(N(NK + 4MK)) = O(N^2K + 4MKN)$. From the experimental results of algorithmic convergence of practical FIR filters, both the number of iterations and the chosen K value are well bounded and independent of N . Hence, the overall complexity can be approximated by $O(N^2 + 4MN)$. It is reasonable to assume M is proportional to N . Hence, our proposed GA algorithm has a quadratic complexity of $O(N^2)$. In practice, the FIR filter order N rarely exceeds 200. The computation efficiency is attested by running the proposed algorithm on one exceptionally long filter of order $N = 293$ from [24]. It took only 8.2s to obtain a solution with optimized TDA cost on a PC equipped with Intel i7-4500CPU running at 1.8GHz and 16GB RAM.

E. Design Example

FIR1 from [18] is used to demonstrate the design flow of our proposed algorithm. This is a low pass filter with the normalized passband and stopband frequencies at $\omega_3 = 0.25$ and $\omega_4 = 0.3$, respectively. The maximum passband ripple and minimum stopband attenuation are 0.5 dB and 27 dB, respectively. The input signal wordlength s is assumed to be 8 bits. With these specifications, the filter order determined by Park-McLellan algorithm is $N = 24$. The 24 infinite-precision coefficients obtained by Park-McLellan algorithm are rounded at the 50th bit position. Their decimal values of the rounded coefficients are listed below.

$\hat{\mathbf{H}} = \{0.028225, -0.012702, -0.017768, 0.024334, 0.013875, -0.038491, -0.0078193, 0.066189, -0.017955, -0.11552, 0.10696, 0.48576, 0.48576, 0.10696, -0.11552, -0.017955, 0.066189, -0.0078193, -0.038491, 0.013875, 0.024334, -0.017768, -0.012702, 0.028225\}$.

With $\hat{\mathbf{H}}$ and the constraints specified in (11), we perform the coefficient synthesis algorithm presented in Fig. 5. The produced decimal values of the coefficients of this optimal solution set $\mathbf{H}'$ are listed below.

$\mathbf{H}' = \{0.027344, -0.012695, -0.017578, 0.023438, 0.013672, -0.038086, -0.0078125, 0.06543, -0.017578, -0.11523, 0.10645, 0.48633, 0.48633, 0.10645, -0.11523, -0.017578, 0.06543, -0.0078125, -0.038086, 0.013672, 0.023438, -0.017578, -0.012695, 0.027344\}$.

The frequency response of this coefficient set $\mathbf{H}'$ is plotted in Fig. 7, which meets all specifications of this FIR filter [18]. The frequency responses of this filter designed by three other methods [8], [10] and [15] are also plotted in Fig. 7 for comparison. All four responses meet the design specifications with different hardware implementation costs. The comparisons of their hardware area and power consumptions in FPGA and ASIC implementations will be presented and discussed in Section IV.

Fig. 7. (a) Frequency responses of FIR1 designed by TDAdmin, [8], [10] and [15], (b) Frequency responses in passband

The optimized TDA block produced by $\mathbf{H}'$ with $j = 9$ is shown in Fig. 8. Since $h'(9) = 0.000111010$ and the word length of input sample, $s = 8$, $w(q_9) = 13$. The word length of the partial sum p_9 , $w(p_9) = 25$, which is longer than 13 bits. Using conventional TDA block design, 25 FAs are required for the carry ripple adder (CRA) of A_9 . The word lengths of subsequent adders A_8 to A_0 as well as the registers R_8 to R_0 also increase accumulatively. Assume an area ratio of 1:1 between a FA and a FF, the equivalent FA cost from A_9 to A_0 in the TDA line is 590 without using the bisection technique. With our proposed algorithm, the 25-bit long p_9 is split into a 13-bit operand p_9^l and a 12-bit operand p_9^u . p_9^l is added with q_9 and p_9^u is latched until the output of A_0 is generated. As the wordlength $w(A_9)$ has reduced from 25 bits to 13 bits, the sizes of subsequent A_8 to A_0 and their accompanying registers R_8 to R_0 are all reduced. Their area costs total up to 370 FAs. Accounting for the area overhead of 37 FAs required for the

additional final adder and the registers to latch p_9^u , there is still a significant 31% FA cost saving of $\Delta_j = 590 - 370 - 37 = 183$.

Fig. 8. FIR filter (FIR1) designed by TDAdmin

The other three filters FIR2 to FIR4 in Table II are also designed by TDAdmin algorithm. Positive savings are obtained as opposed to the negative savings obtained using the coefficients generated from Park-McLellan algorithm directly. This comparison shows that simple TDA bisection technique can effectively reduce the overall complexity of FIR filter with the help of coefficient set generated by TDAdmin. To further evaluate the effectiveness of the proposed TDAdmin algorithm, in the next section, logic synthesis will be performed on the benchmark filters designed by TDAdmin and compared with solutions generated by other methods.

IV. LOGIC SYNTHESIS RESULTS AND DISCUSSION

In this section, a group of practical filters are designed by the proposed algorithm and a few recently published competing methods. The logic synthesis and power simulation results will be compared and discussed. As the filter length of most practical filters ranges from 25 to 200 and the precision of their coefficients ranges from 8 bits to 16 bits, the parent set size K of our algorithm is set to 50, the group size S for pairing the solutions is set to 10 and the number of large coefficients M is set to $N/10$ rounded to its nearest integer. The latter implies that only 10% of the largest coefficients of a coefficient set are eligible for mutation. The minimum number of iteration T is set to 6 and the threshold in the average saving increment for termination is set to $1/K$. After $\mathbf{H}'$ is generated by TDAdmin, Pasko algorithm [7] is used to minimize the MCM block.

The benchmark filter specifications are listed in Table IV, which are used by the proposed TDAdmin and its competing algorithms to synthesize the filter coefficients. FIR1 [18] is a low pass filter, which is used as design example in Section III.E. FIR2 to FIR12 are benchmark filters presented in the respective references cited next to them. These filters include practical filters with different specifications and filter lengths. For example, FIR2 and FIR4 are filters that have very large stopband attenuation (> 80 dB). These filters are used for the applications that demand very high SNR. FIR4, FIR8, FIR10 and FIR12 are high selectivity filters with very narrow transition band. These filters have lengths from 20 to 293, representing a good sampling of short, median and long filters. The designs generated by the TDAdmin are compared against those generated by [8], [10] and [15]. The design method in [8] is also a linear programming based algorithm. The objective of

its coefficient synthesis is to maximize the sharing of two most frequently encountered types of common subexpressions in the MCM block without considering the optimization of TDA block. As it is also formulated based on integer linear programming, it is a good candidate to compare the overall cost reduction achievable by TDA cost aware and typical MCM cost aware design algorithms. The design algorithm [10] is a recently proposed low complexity FIR filter synthesis algorithm. By using nonlinear quantization with uneven word length, it minimizes the bit width of filter coefficients. The solutions generated by the coefficients of this latest MCM block minimization algorithm [10] can further attest the dominance of structural adder costs and the savings achieved by TDAdmin is far more than the savings achievable by the state-of-the-art minimization of MCM block. The method proposed in [15] is the most recent design method targeting SA block minimization. Thus, its comparison provides the most compelling proof of the effectiveness of TDAdmin in producing a more optimized TDA block.

Table IV. Specifications of the twelve benchmark filters

Filter	Passband and stopband frequencies (cycle/sample)	Passband ripple (dB)	Stopband attenuation (dB)
FIR1 [18]	$\omega_3 = 0.25$; $\omega_4 = 0.3$	0.4	27
FIR2 [19]	$\omega_3 = 0.23$; $\omega_4 = 0.29$	0.34	86.6
FIR3 [20]	$\omega_3 = 0.2$; $\omega_4 = 0.3$	0.004	70
FIR4 [7]	$\omega_1 = 0.37$; $\omega_2 = 0.4$	0.08	80.3
FIR5 [24]	$\omega_3 = 0.1$; $\omega_4 = 0.14$	0.48	60
FIR6 [8]	$\omega_3 = 0.15$; $\omega_4 = 0.25$	0.27	43
FIR7 [25]	$\omega_3 = 0.1$; $\omega_4 = 0.2$	0.0549	50
FIR8 [26]	$\omega_1 = 0.4$; $\omega_2 = 0.45$	1.743	40
FIR9 [24]	$\omega_3 = 0.021$; $\omega_4 = 0.07$	0.2	60
FIR10 [24]	$\omega_3 = 0.0625$; $\omega_4 = 0.07$	0.086	46
FIR11 [27]	$\omega_3 = 0.4$; $\omega_4 = 0.5$	0.086	50
FIR12 [28]	$\omega_3 = 0.2$; $\omega_4 = 0.24$	0.086	40

All the designs are mapped to Xilinx Spartan6, xc6slx75t FPGA device and synthesized using Xilinx ISE WebPACK v14.7. The synthesized areas in number of LUTs and delays in *ns* are presented in Table V. After the design is implemented, the NCD file output from Place & Route (PAR) is read by Xilinx ISE Xpower Analyzer (XPA) to estimate the activity rates. Input frequency and supply voltage among others are specified in the settings of simulation activity files for XPA to perform the power analysis. A clock frequency of 70 MHz and a supply voltage of 1.2V are set consistently for all the designs. The power dissipation results in μ W simulated by XPA are listed in Table VI.

From Tables V and VI, the proposed TDAdmin algorithm reduces the LUT costs by 26.8%, 8.7% and 9.4% on average over [8], [10] and [15], respectively. As the MCM blocks are implemented by the same algorithm using the coefficient sets synthesized by all algorithms in comparison, the lower implementation complexity of the FIR filters designed by TDAdmin is attributed solely to the optimization of TDA block with appropriate choice of SA splitting position. The delays of FIR filters designed by TDAdmin are also 1.2%, 4.6% and 1.2% shorter than [8], [10] and [15], respectively. From the total power comparison, it is evident that the solutions generated by

TDAdmin are the most power efficient among all. The power reductions over [8], [10] and [15] are on average 9.7%, 3.0% and 27.5%, respectively.

Table V. Synthesized FPGA areas in #LUT and delays in *ns* for FIR filters designed by [8], [10], [15] and TDAdmin

Filters	Areas in #LUT				Delays in <i>ns</i>			
	[8]	[10]	[15]	TDAdmin	[8]	[10]	[15]	TDAdmin
FIR1	768	722	782	632	7.9	8.1	7.3	7.4
FIR2	1374	1261	1297	1205	10.1	10.0	8.2	9.3
FIR3	1407	1315	1680	1268	11.8	11.8	11.4	11.0
FIR4	6742	4622	5060	4327	6.4	6.8	11.1	6.8
FIR5	2047	1664	1289	1632	11.6	11.4	10.4	11.0
FIR6	1054	769	714	687	9.4	9.7	9.5	9.6
FIR7	1231	810	964	728	9.2	9.5	9.3	9.2
FIR8	2760	1616	1393	1547	10.0	10.6	8.7	11.3
FIR9	2830	2272	1980	2219	10.0	10.0	9.7	9.6
FIR10	12930	10797	13082	8399	12.5	12.0	12.3	12.8
FIR11	1012	896	772	743	8.2	8.6	7.7	7.6
FIR12	3387	2867	3740	2597	9.4	12.0	12.2	9.3

Table VI. Total power in μ W on FPGA implementation of FIR filters designed by [8], [10], [15] and TDAdmin

Filters	[8]	[10]	[15]	TDAdmin
FIR1	0.108	0.105	0.155	0.099
FIR2	0.141	0.138	0.191	0.136
FIR3	0.159	0.153	0.221	0.149
FIR4	0.301	0.270	0.434	0.268
FIR5	0.169	0.165	0.202	0.165
FIR6	0.123	0.116	0.165	0.113
FIR7	0.137	0.115	0.173	0.110
FIR8	0.186	0.154	0.201	0.151
FIR9	0.193	0.185	0.234	0.183
FIR10	0.765	0.700	1.015	0.637
FIR11	0.167	0.164	0.160	0.157
FIR12	0.288	0.265	0.333	0.259

All the generated designs are also mapped to STM 65nm standard cell library and synthesized by Synopsys DesignCompiler™. The synthesized areas in μm^2 and delays in *ns* of the designs generated by the proposed and competing algorithms are presented in Table VII. The logic synthesis and optimization effort is set to minimum delay. The total power, including dynamic power and leakage power, consumed by the FIR circuits are simulated by Synopsys Prime Time PX version: Z-2006.12. The supply voltage and clock frequency are set to 0.9V and 250 MHz, respectively. As switching power is input dependent, Monte Carlo power simulation [29] was adopted by applying randomly generated input vectors epoch by epoch until the maximum error in mean power falling within the 95% confident interval converges to 5% or lower. This goal was reached with slightly more than 360 test vectors. Eventually, 400 random input vectors were applied, which worked out to a statistical error bound of 4% within 95% confidence interval. The power simulation results are presented in Table VIII.

Table VII. Synthesized ASIC areas in μm^2 and delays in ns for FIR filters designed by [8], [10], [15] and TDAdmin

Filters	Areas in μm^2				Delays in ns			
	[8]	[10]	[15]	TDAdmin	[8]	[10]	[15]	TDAdmin
FIR1	21648	14705	13213	13423	1.39	1.40	1.40	1.56
FIR2	33436	27620	24400	25433	1.49	1.58	1.40	1.53
FIR3	37283	31661	32796	28295	1.89	1.84	2.06	1.85
FIR4	122615	101438	102294	87215	2.03	1.98	2.20	1.93
FIR5	41080	33590	31457	28534	1.82	1.78	1.68	1.81
FIR6	18779	14641	14795	12326	1.75	1.56	1.54	1.73
FIR7	25082	18117	17677	15219	1.73	1.56	1.46	1.52
FIR8	38719	30400	31328	24137	1.74	1.77	1.96	1.71
FIR9	39750	40172	38844	34003	1.78	1.74	1.70	1.71
FIR10	248608	231573	223689	170394	2.10	2.07	2.37	1.90
FIR11	19800	19566	16336	16074	1.24	1.31	1.42	1.22
FIR12	81068	78612	69422	61938	1.30	1.55	1.79	1.43

Table VIII. Simulated total power in μW on ASIC implementation of FIR filters designed by [8], [10], [15] and TDAdmin

Filters	[8]	[10]	[15]	TDAdmin
FIR1	8.06	4.53	4.28	4.61
FIR2	10.17	7.95	7.09	7.16
FIR3	9.18	8.3	7.87	7.55
FIR4	25.97	23.19	24.22	21.93
FIR5	9.32	8.63	8.21	7.68
FIR6	5.25	4.51	4.39	4.27
FIR7	7.62	5.48	4.74	4.51
FIR8	9.92	7.44	7.27	6.84
FIR9	9.77	10.85	11.19	10.03
FIR10	66.21	64.36	62.84	51.24
FIR11	3.28	3.32	2.70	2.72
FIR12	11.77	13.34	13.89	9.99

From Tables VII, the average area reductions of the FIR filters designed by the proposed TDAdmin algorithm are 28.8%, 15.8% and 11.2% compared with those designed by [8], [10] and [15], respectively. The noteworthy 28.8% area reduction over [8] indicates that the TDA block optimization contributes more significantly to the overall complexity reduction than the MCM block minimization. As for delay, TDAdmin produces solutions with 1.1%, 0.2% and 3.4% shorter delays than those of [8], [10] and [15], respectively. The timing optimization effort made by the synthesis tool is capable of closing the gap between the critical path delays of solutions generated by different design algorithms. From Table VIII, the total power reductions of the solutions generated by our proposed algorithm over [8], [10] and [15] are 22.1%, 11.3% and 6.7%, respectively. The savings in power consumption are ascribed to the less complicated adders and registers of TDA block. A couple of short filters, such as FIR1 of $N=24$ and FIR2 of $N=36$, designed by TDAdmin are found to consume slightly more area and power than those designed by [15]. This is because the short TDA line has limited the number of structural adders that can be benefited from the length reduction after the bisection point. The optimal bisection point depends on the design specifications. TDAdmin works better if the coefficient values after the bisection are much smaller than those values before bisection. If the design specifications of short filters are too restrictive, the low variance in coefficient values may push the optimal bisection point closest to the output. Even though the cost of the solution generated by the bisection technique may

not be the least under such circumstance, it is not too far off. FIR1 and FIR2 designed by TDAdmin consume on average only 2.9% more area than the best solutions designed by [15]. For most other short filters, such as FIR6 with bisection at $j=13$ for $N=20$, FIR7 with bisection at $j=13$ for $N=26$ and FIR11 with bisection at $j=17$ for $N=28$, the complexity of their structural adders after the bisection point can still be substantially reduced by TDAdmin, making them the least area and power consumption among all solutions in comparison. Excluding FIR1 and FIR2, TDAdmin saves on average 14% area over [15]. The same observation goes for power consumption. The complexity of TDA block increases faster than the MCM block as N increases. FIR10 is an exemplary case with an extremely narrow transition band. FIR10 has more than 270 coefficients of 16-bit precision. Its area, delay and power consumption have been reduced by 22.6%, 20.4% and 18.5%, respectively by TDAdmin compared with the solution generated by [15].

As the delays of the solutions generated by these algorithms are not appreciably different, to better compare their more significant overall performance deviation in logic complexity and power consumption, the area-power (AP) complexity is calculated as the product of silicon area in μm^2 and total power in μW . The AP complexities of the twelve benchmark filters designed by the four algorithms in comparison are listed in Table IX. It is evident that, on average, the AP complexity of the filters designed by the proposed TDAdmin algorithm is lower than those of [8], [10] and [15] by 43.8%, 25.1%, and 16.8%, respectively. The average AP complexities of the twelve filters designed by the four algorithms are plotted in Fig. 9.

Table IX. The power-area product in $\mu\text{W} \times \mu\text{m}^2$ for ASIC implementation of FIR filters designed by [8], [10], [15] and TDAdmin

Filters	[8]	[10]	[15]	TDAdmin
FIR1	174482.88	66613.65	56551.64	61880.03
FIR2	340044.12	219579.00	172996.00	182100.28
FIR3	342257.94	262786.30	258104.52	213627.25
FIR4	3184311.55	2352347.22	2477560.68	1912624.95
FIR5	382865.60	289881.70	258261.97	219141.12
FIR6	98589.75	66030.91	64950.05	52632.02
FIR7	191124.84	99281.16	83788.98	68637.69
FIR8	384092.48	226176.00	227754.56	165097.08
FIR9	388357.50	435866.20	434664.36	341050.09
FIR10	16460335.68	14904038.28	14056616.76	8730988.56
FIR11	64944	64959.12	44107.20	43721.28
FIR12	954170.36	1048684.10	964271.58	618760.60

Fig. 9. Average AP complexities in $\mu\text{W} \times \mu\text{m}^2$ of the twelve benchmark filters designed by [8], [10], [15] and TDAdmin

V. CONCLUSION

Motivated by the dominating cost of TDA block in FIR filter which cannot be effectively reduced by existing design algorithms that maximize sharing of adders in the MCM block, this paper presents a new design methodology for area-power efficient FIR filter implementation by synthesizing the filter coefficients to specifically maximize the cost savings of TDA block. The effective implementation cost of TDA block is determined based on operand bisection at some optimal tap position to reduce the sizes of subsequent structural adders and registers without violating the filtering specifications. The renewed coefficient synthesis problem is solved by a dedicated Genetic Algorithm. Our proposed solutions are at least 16.8% more area-power efficient based on the comparison of synthesis results in 65nm standard cell technology with designs generated by three other state-of-the-art design algorithms for twelve benchmark filter specifications.

ACKNOWLEDGEMENT

We would like to thank Dr. Lou Xin for his kind sharing of the source code of his recently published algorithm. We would also thank Dr. Sumedh Somnath Dhabu and Mr. Zhang Yufei for their kind support in part of the experiments. This research is supported by the SUTD-MIT International Design Centre (IDC) at Singapore University of Technology and Design.

REFERENCES

- [1] H. Ma, L. Liu, A. Zhou and D. Zhao, "On networking of internet of things: explorations and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 441-452, Aug. 2016.
- [2] J. Chen and C. H. Chang, "High-level synthesis algorithm for the design of reconfigurable constant multiplier," *IEEE Trans. on Compute Aided Designs*, vol. 28, no. 12, pp. 1844-1856, Dec. 2009.
- [3] T. Chen, Y. V. Zakharov and C. Liu, "Low-complexity channel-estimate based adaptive linear equalizer," *IEEE Signal Processing Lett.*, vol. 18, no. 7, pp. 427-430, Jul. 2011.
- [4] M. R. Zahabi, V. Meghdadi, J. P. Cances and A. Saemi, "Mixed-signal matched filter for high-rate communication systems," *IET Signal Processing*, vol. 2, no. 4, pp. 354-360, Dec. 2008.
- [5] M. S. Prakash and R. A. Shaik, "Low-area and high-throughput architecture for an adaptive filter using distributed arithmetic," *IEEE Trans. on Circuits and Syst. II*, vol. 60, no. 11, pp. 781-785, Nov. 2013.
- [6] H. Mahdaviifar, M. El-Khamy, J. Lee, and Inyup. Kang, "Performance limits and practical decoding of interleaved Reed-Solomon polar concatenated codes," *IEEE Trans. Communications*, vol. 62, no. 5, pp. 1406-1417, May 2014.
- [7] R. Paško, P. Schaumont, V. Derudder, S. Vernalde and D. Āuračková, "A new algorithm for elimination of common subexpressions," *IEEE Trans. Compute Aided Designs*, vol. 18, no. 1, pp. 58-68, Jan. 1999.
- [8] Y. Yu and C. Y. 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.
- [9] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," *IEEE Trans. Compute Aided Designs*, vol. 29, no. 2, pp. 275-288, Feb. 2010.
- [10] S. Hsiao, J. Z. Jian, and M. Chen, "Low-cost FIR filter designs based on faithfully rounded truncated multiple constant multiplication/accumulation," *IEEE Trans. Circuits Syst. II*, vol. 60, no. 5, pp. 287-291, May 2013.
- [11] J. Chen, C. H. Chang, F. Feng, W. Ding and J. Ding, "Novel Design Algorithm for Low Complexity Programmable FIR Filters Based on Extended Double Base Number System," *IEEE Trans. Circuits Syst. I*, vol. 62, no. 1, pp. 224-233, Jan. 2015.
- [12] F. Feng, J. Chen and C. H. Chang, "Hypergraph based minimum arborescence algorithm for the optimization and reoptimization of multiple constant multiplications," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 2, pp. 233-244, Feb. 2016.
- [13] J. Ding, J. Chen and C. H. Chang, "A new paradigm of common subexpression elimination by unification of addition and subtraction," *IEEE Trans. on Compute Aided Designs*, vol. 35, no. 10, pp. 1605 - 1617, Oct. 2016.
- [14] M. Faust and C. H. Chang, "Optimization of structural adders in fixed coefficient transposed direct form FIR filters," in *Proc. IEEE Int. Symposium on Circuits and Systems*, Taipei, Taiwan, pp. 2185-2188, May 2009.
- [15] X. Lou, Y. Yu and P. K. Meher, "Analysis and Optimization of Product-Accumulation Section for Efficient Implementation of FIR Filters," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 10, pp. 1701-1713, Oct. 2016.
- [16] J. E. Beasley, *Advances in Linear and Integer Programming*, Oxford Science, 1996.
- [17] J. H. McClellan and T. W. Parks, "A personal history of the Parks-McClellan algorithm," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 82 - 86, Mar. 2005.
- [18] A. G. Dempster, S. S. Dimirsoy, and I. Kale, "Designing multiplier blocks with low logic depth," in *Proc. IEEE Int. Symp. on Circuits Syst.*, pp. 773-776, Scottsdale, Arizona, May 2002.
- [19] Y. C. Lim and S. Parker, "Discrete coefficient FIR digital filter design based upon an LMS criteria," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 30, no. 10, pp. 723-739, Oct. 1983.
- [20] J. Laskowski and H. Samuelli, "A 150-mhz 43-tap Half-band Fir Digital Filter In 1.2-/spl mu/m Cmos Generated By Silicon Compiler", in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 11.4.1 - 11.4.4, May 1992.
- [21] M. Faust, "Design methodologies for complexity reduction of FIR filters," PhD Thesis, Nanyang Technological University, Singapore, 2013.
- [22] J. Chen; J. Tan; C. H. Chang; F. Feng, "A New Cost-Aware Sensitivity-Driven Algorithm for the Design of FIR Filters," *IEEE Trans. Circuits Syst. I*, vol. 64, no. 6, pp. 1588-1598, Jun. 2017.
- [23] K. Deep, K. P. Singh, M. L. Kansal and C. Mohan, "A real coded genetic algorithm for solving integer and mixed integer optimization problems," *Applied Mathematics and Computation*, vol. 212, no. 2, pp. 505-518, Jun. 2009.
- [24] M. Aktan, A. Yurdakul and G. Dunder, "An algorithm for the design of low-power hardware-efficient FIR filters," *IEEE Trans. on Circuits and Syst. I*, vol. 55, no. 6, pp. 1536-1545, Jul. 2008.
- [25] S. R. Kotha, S. Vij, and S. K. Sahoo, "A study on strategies and Mutant factor in differential evolution algorithm for FIR filter design," in *Proc. Int. Signal Process. Integr.* pp. 50-55, Netw., Noida, India, Feb. 2014.
- [26] S. Mandal, S. P. Ghoshal, R. Kar, and D. Mandal, "Design of optimal linear phase FIR high pass filter using craziness based particle swarm optimization technique," *Journal of Comput. Inf. Sci. J. King Saud Univ.*, vol. 24, no. 1, Jan. 2012.
- [27] C.-L. Chen and J. Willson, A. N., "A trellis search algorithm for the design of FIR filters with signed-powers-of-two coefficients," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 29-39, Jan. 1999.
- [28] A. Shahein, Q. Zhang, N. Lotze, and Y. Manoli, "A Novel Hybrid Monotonic Local Search Algorithm for FIR Filter Coefficients Optimization," *IEEE Trans. Circuits Syst. I*, vol. 59, no. 3, pp. 616-627, Mar. 2012.
- [29] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. VLSI Syst.*, vol. 1, no. 1, pp. 63-71, Mar. 1993.

Jiajia Chen (M'14) received his B. Eng. (Hons) and Ph.D. from Nanyang Technological University, Singapore, in 2004 and 2010, respectively. Since April 2012, he has been working in Singapore University of Technology and Design where he is currently a Senior Lecturer. His research interest includes computational transformations of low-complexity digital filters, programmable filters and digital signal processing. He has published one book chapter and over 30 research papers in SCI journals and international conferences. Dr. Chen served as Web Chair of *Asia-Pacific Computer Systems Architecture Conference* 2005, Technical Program Committee member of *European Signal Processing Conference* 2014 and Associate Editor of *Springer EURASIP Journal on Embedded Systems* since 2016.

Chip-Hong Chang (S'92–M'98–SM'03) received the B.Eng. (Hons.) degree from the National University of Singapore in 1989, and the M. Eng. and Ph.D. degrees from Nanyang Technological University (NTU) in 1993 and 1998, respectively. He served as a Technical Consultant in industry prior to joining the School of Electrical and Electronic Engineering (EEE) of NTU in 1999, where he is currently an Associate Professor. He holds joint appointments with the university as Assistant Chair of Alumni of the School of EEE from 2008 to 2014, Deputy Director of the Center for High Performance Embedded Systems from 2000 to 2011, and the Program Director of the Center for Integrated Circuits and Systems from 2003 to 2009. He has edited four books, published ten book chapters, 87 international journal papers (two-thirds are IEEE) and more than 160 refereed international conference papers (mostly in IEEE), and delivered over 30 colloquia. His current research interests include hardware security and trustable computing, low-power and fault-tolerant computing, residue number systems, and application-specific digital signal processing.

Dr. Chang serves as the Associate Editor of IEEE Transactions on Very Large Scale Integration (VLSI) Systems since 2011, IEEE Access since 2013, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems and IEEE Transactions on Information Forensic and Security since January 2016, IEEE Transactions on Circuits and Systems-I from 2010-2013, Integration, the VLSI Journal from 2013-2015, Springer Journal of Hardware and System Security since June 2016 and Microelectronics Journal since May 2014. He was the editorial advisory board member of Open Electrical and Electronic Engineering Journal from 2007 to 2013 and Journal of Electrical and Computer Engineering from 2008 to 2014, as well as the guest editor of special issues for IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Circuits and Systems-I, Journal of Circuits, Systems and Computers and Journal of Electrical and Computer Engineering. He also served in the organizing and technical program committee for more than 50 international conferences (mostly IEEE). He is a Fellow of IET.

Jiatao Ding is pursuing his B.Eng. degree in computer engineering at Singapore University of Technology and Design. He is currently working as a part-time research assistant at SUTD-MIT International Design Center. His research interests include digital filter design and implementation, CAD algorithms for high-level synthesis and new logical operator design.

Rui Qiao is pursuing his B.Eng. degree in computer science at Singapore University of Technology and Design. He is currently working as student research assistant at SUTD-MIT International Design Center. His research interests include digital filter design and system optimization and data analysis.

Mathias Faust (S'07, M'14) received the Dipl. Ing. FH (B.Eng.) degree from the University of Applied Sciences Rapperswil (HSR), Switzerland, in 2007, and Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2014, respectively. He is currently working as IT Manager, System Engineer and Consultant in industry. His research interest includes low complexity digital signal processing, digital filter design and optimization, and computer-aided design for computer

arithmetic circuits. Dr. Faust is a member of the IET and of Swiss Engineering STV.