# COEFFICIENT OPTIMIZATION FOR AREA-EFFECTIVE MULTIPLIER-LESS FIR FILTERS\*

Tay-Jyi Lin, Tsung-Hsun Yang, and Chein-Wei Jen

Department of Electronics Engineering National Chiao Tung University, Taiwan

## ABSTRACT

This paper presents the systematic synthesis of multiplier-less FIR filters with a novel complexity-aware quantization algorithm. Both signed-digit representations and common subexpression elimination (CSE) are investigated to reduce the computational complexity at the bit level. For comparable filter responses, the simulation shows that our approach requires only half (49.06%~ 50.94%) additions of the straightforward quantized filters. Both with CSE, our FIR synthesizer has comparable results with the CSD-encoded coefficients, which have the theoretically minimum non-zero terms. Moreover, our approach outperforms CSD in most cases because of the direct control over additions and the zero-overhead insertion of non-zero terms.

## **1** INTRODUCTION

Most DSP kernels are transforms with fixed coefficients, such as FIR filters, discrete cosine transform (DCT), and fast Fourier transform (FFT), etc. Adders and shifters can replace the constant multipliers for area-efficient implementations, where the area can be further reduced by common subexpression elimination (CSE). But the hardware complexity is not under control during the rounding of the infinite-wordlength coefficients to the quantization levels. An iterative algorithm was proposed by Li et al. [1] to distribute a pre-defined SPT (signed power-of-two terms) budget, which gives an approximate estimate of the additions of the implementation, to the quantized coefficients (QC). This may lead to a less optimal design, especially when CSE is applied. In this paper, we propose a complexity-aware algorithm to quantize the coefficients by allocating SPT terms under an exact adder budget, while taking into account of the common subexpressions of the computations. Different coefficient approximation strategies are investigated, where the combinations with a novel CSE scheme can reduce up to 50.94% budget for comparable filter responses.

The rest of this paper is organized as follows. Section 2 reviews the common subexpressions in FIR filters and the coefficient quantization by successive approximation. Section 3 describes the proposed complexity-aware quantization algorithm with the discussions of different approximation strategies in Section 4, where a new kind of common subexpression is also introduced. Section 5 summarizes the simulation results and finally Section 6 concludes this work and outlines our future research.

#### 2 PRELIMINARY

Consider a 4-tap FIR filter with the coefficients:  $h_0=0.0111011$ ,  $h_1=0.0101011$ ,  $h_2=1.0110011$ , and  $h_3=1.1001001$ , which are four fractional numbers represented in the 8-bit 2's complement format. The filter output is computed as

 $y_n = h_0 \cdot x_n + h_1 \cdot x_{n-1} + h_2 \cdot x_{n-2} + h_3 \cdot x_{n-3}$ 

Additions and shifts can be substituted for the multiplications as

$$y_{n} = x_{n} \gg 2 + x_{n} \gg 3 + x_{n} \gg 4 + x_{n} \gg 6 + x_{n} \gg 7$$
  
+  $x_{n-1} \gg 2 + x_{n-1} \gg 4 + x_{n-1} \gg 6 + x_{n-1} \gg 7$   
-  $x_{n-2} + x_{n-2} \gg 2 + x_{n-2} \gg 3 + x_{n-2} \gg 6 + x_{n-2} \gg 7$   
-  $x_{n-3} + x_{n-3} \gg 1 + x_{n-3} \gg 4 + x_{n-3} \gg 7$  (1),

where ">>" denotes the arithmetic right shift. Each output needs 17 additions (including subtractions), and 16 shifts.

#### 2.1 Common Subexpression Elimination

The number of additions and shifts of the multiplier-less directform FIR filter can be reduced by eliminating the common subexpressions across coefficients (CSAC), within coefficients (CSWC) [2], and across iterations (CSAI) [3].

*CSAC*: The  $h_0$  and  $h_2$  multiplications, i.e. the first and the third rows in Equation (1), have four common terms with identical shifts. Restructuring (1) by adding  $x_n$  and  $x_{n-2}$  first can eliminate the redundant CSAC between  $h_0$  and  $h_2$  as

$$y_n = (x_n + x_{n-2}) + (x_n + x_{n-2}) + (x_n + x_{n-2}) + (x_n + x_{n-2}) + x_n + x_{n-2} + x_{n-2} + x_{n-1} + x_{n-2} + x_{n-1} + x_{n-2} + x_{n-1} + x_{n-2} + x_{n-1} + x_{n-2} +$$

where the total additions and shifts are reduced to 14 and 12, respectively. The extraction and elimination process of CSAC can be more concisely manipulated in tabular form as Fig 1.

	b7	b6	b5	b4	b3	b2	b1	b0			b7	b6	b5	b4	b3	b2	b1	b0
h0	0	0	1	1	1	0	1	1		h0	0	0	0	0	1	0	0	0
h1	0	0	1	0	1	0	1	1		h1	0	0	1	0	1	0	1	1
h2	-1	0	1	1	0	0	1	1		h2	-1	0	0	0	0	0	0	0
h3	-1	1	0	0	1	0	0	1		h3	-1	1	0	0	1	0	0	1
									-	h02	0	0	1	1	0	0	1	1

#### Figure 1 CSAC extraction & elimination

*CSWC*: Within a coefficient or a CSAC term, bit-pairs with identical bit displacement are recognized as CSWC, which can also be eliminated by reusing the computation. For example, the subexpression in the first row of Equation (2) can be simplified as  $(x_{02}+x_{02}\gg1)\gg2+(x_{02}+x_{02}\gg1)\gg6$  to reduce one addition and one shift, where  $x_{02}$  stands for  $x_n+x_{n-2}$ .

*CSAI*: The subexpression  $x_n+x_{n-1}+x_{n-2}+x_{n-3}$  in Equation (1) can be restructured as  $(x_n+x_{n-1})+z^{-2}\cdot(x_n+x_{n-1})$  to reduce one addition,

<sup>\*</sup> This work was supported by the National Science Council, Taiwan under Grant NSC91-2218-E009-011

which is referred to as the elimination of CSAI. But the  $z^{-1}$  is costly to implement, especially in the bit-serial architectures, which requires *w* flops, where *w* denotes the input word-length. The silicon area is roughly equivalent to *w* adders. Therefore, we do not consider CSAI in this paper.

CSE quality depends on the elimination order. A steepestdescent approach is applied as a heuristic to reduce the search space, which first eliminates the redundant CSAC between the coefficient pair with the most non-zero terms in identical bitpositions. One-level look-ahead can further distinguish the candidates of the same weight. CSWC elimination can use a similar strategy. Fig 2 shows the CSE algorithm for CSAC and CSWC.

Eliminate zero coefficients

Merge coefficients with the same value (e.g. linear-phase FIR filters) Construct a coefficient matrix of size N×W, where N is the number of coefficients for CSE and W is the word-length (as shown in Fig 1)

WHILE (highest weight > 1) // CSAC elimination { Find the coefficient pair with the highest weight

Update the coefficient matrix }

FOR each row in the coefficient matrix // CSWC elimination { Find bit-pairs with identical bit displacement

Extract the distances between those bit-pairs Update the coefficient matrix and record the shift information }

Output the SDFG with addition, shift and negation

Figure 2 CSE algorithm for direct-form FIR filters

#### 2.2 Coefficient Quantization by Successive Approximation

Straightforward quantization by truncating or rounding the ideal coefficients is not aware of its implementation complexity. Small variation on the coefficient magnitudes that slightly impacts the filter characteristics can reduce the non-zero terms or improve the CSE result significantly. Quantization by iteratively allocating the signed power-of-two (SPT) terms [1] is an alternative. It gradually assigns SPT terms to the quantized coefficients (QC) under a pre-defined SPT budget. Besides, an additional scale factor (SF) is applied to collectively settle the coefficients into an optimal quantization space.

**Normalize** IC so that the maximum coefficient magnitude is 1 SF = lower bound

WHILE (SF < upper bound)

{ Scale the normalized IC with SF

WHILE (budget >0 & the largest difference between QC & IC >2<sup>-w</sup>) { Allocate an SPT to the QC that differs most from the scaled NIC } Estimate the square error between IC & the normalized QC

SF = SF × [min |QD| + |coef(min QD)|] / |coef(min QD)| }

Choose the QC that has the minimum square error

(a)

$$\begin{split} & \text{IC} = [0.26 \ 0.131 \ 0.087 \ 0.011] \\ & \text{Normalized IC} (\text{NIC}) = [1 \ 0.5038 \ 0.3346 \ 0.0423], \text{NF} = \text{max}(\text{IC}) = 0.26 \\ & \text{When SF} = 0.5 \\ & \text{Scaled NIC} = [0.5 \ 0.2519 \ 0.1673 \ 0.0212] \\ & \text{QC}\_0 = [0 \ 0 \ 0 \ 0] \\ & \text{QC}\_1 = [0.5 \ 0 \ 0 \ 0] \\ & \text{QC}\_2 = [0.5 \ 0.25 \ 0 \ 0] \\ & \text{QC}\_2 = [0.5 \ 0.25 \ 0 \ 0] \\ & \text{QC}\_4 = [0.5 \ 0.25 \ 0.15625 \ 0] \\ & \text{QC}\_5 = [0.5 \ 0.25 \ 0.15625 \ 0.015625] \\ & \text{(b)} \end{split}$$

Figure 3 Quantization by successive approximation

Fig 3(a) is our modified algorithm [4] with the improved SF exploration. The ideal coefficients (IC) with infinite word-length

are first normalized so that the maximum coefficient is one. An optimal SF is explored within the given gain tolerance (e.g.  $\pm 3$ dB). Instead of the fixed  $2^{-w}$  stepping [1] from the lower bound (*w* denotes the wordlength), the next SF is calculated as

SF' = SF × [min |QD| + |coef(min QD)|] / |coef(min QD)|, where QD denotes the distance of an ideal coefficient to its next quantization level, which depends on the approximation strategy (e.g. rounding to the nearest value, toward 0, or toward  $-\infty$ , etc), and |coef(min QD)| denotes the magnitude of the ideal coefficient with the minimum QD. In brief, the next SF is the minimum value that scales the magnitude of any coefficient to its next quantization level. In 16-bit quantization with  $\pm 3dB$ acceptable gain, the fixed 2<sup>-w</sup> stepping SF exploration has 45,875 SF candidates, while our proposed variable step-size approach has 14,986 to 20,429 candidates only, depending on the coefficients in our simulation. This is because the former one steps over multiple SF's that have an identical QC result [4].

For each SF, the QC's are first initialized to zeros and SPT is iteratively allocated to the QC that differs most from the scaled NIC (normalized IC). The allocation stops whenever the SPT budget is exhausted or all differences between IC and QC pairs are less than  $2^{-W}$ . Finally, the normalized QC with the least square error from the IC is chosen. Fig 3(b) is an example to illustrate the SPT allocation when SF=0.5.

# 3 PROPOSED COMPLEXITY-AWARE COEFFICIENT QUANTIZATION

Allocating an SPT term to the quantized coefficient (QC) set either results in an isolated term to sum up, or just enlarges a common subexpression without any addition overhead. But the number of additions during the SPT allocation process is not always 'non-decreasing' because of the steepest-descent CSE heuristic. For example, if the optimum CSE for a QC set does not start with the coefficient pair with the most CSAC terms (i.e. the steepest-descent heuristic cannot find the minimum number of additions). Allocating a zero-overhead SPT term, which increases the weight of the allocated pair by one, may alter the CSE order and possibly leads to a better CSE result with reduced additions.



Figure 4 Example – inserting a zero-overhead SPT term reduces the required number of additions

Fig 4 is an example where the number of additions decreases after extra SPT insertion. The left three matrices are coefficients before CSE with marked CSAC terms to be eliminated. The right matrix in (a) is the heuristic CSE result with the CSWC terms highlighted, which requires 19 additions. A zero-overhead SPT term is then allocated to the LSB of  $h_1$  as shown in (b), which reorders the CSE and introduces a new budget of 2 additions by a better CSAC elimination (it needs 17 additions only). Changing the order of (a) into that of (b), the CSE result of (c) also requires 17 additions only. Thus, the number of additions is non-decreasing only if CSE is optimum.

Fig 5(a) is our complexity-aware quantization algorithm that controls the CSE heuristic to ensure the minimum allocated additions during successive approximation. The QC's are first initialized to zeros and an SPT term is continuously assigned to the QC that differs most from the scaled NIC. Once the allocated SPT terms amount to the remnant budget, CSE is performed to introduce a new budget. The iteration of SPT allocation continues until no budget is found. Zero-overhead SPT terms are then inserted by pattern matching. The postprocessing can introduce a new budget as the illustrating example given in Fig 5(b). Therefore, a skip queue is needed to insert more significant SPT terms, once such budget is available. The less significant zero-overhead SPT terms that are already allocated should be completely removed. Finally, additional CSE is performed to check if there exists a new order for better CSE. If a new budget is found and the skip queue is empty, the iterative SPT allocation resumes. Otherwise, the original CSE order is used.



Figure 5 Complexity-aware coefficient quantization

The steepest-descent heuristic CSE can have a worse result after the SPT insertion, and the remnant budget will be negative (i.e. the required adders of the resultant QC set exceeds the predefined budget). We save this situation just by canceling the latest SPT allocation and using the previous CSE order instead, as the procedures in the right-hand side of Fig 5(a). The remnant budget is used up with the fixed CSE order, where the overhead is estimated with pattern matching. The procedure is similar to the insertion of zero-overhead SPT terms except that no skip queue is implemented. By the way, the algorithm terminates of course, whenever the maximum difference between each QC & IC pair is less than  $2^{-w}$ , because the QC cannot improve anymore.

## 4 COEFFICIENT APPROXIMATION STRATEGIES & SHIFTED CSAC

The strategy for coefficient approximation strongly affects the implementation complexity. Fig 6 shows an illustrating example of several rounding strategies to quantize 0.484375 via successive approximation. If the quantized coefficient must be represented in the 2's complement form (i.e. only the MSB has a negative weight), always approximating the ideal coefficient with the nearest quantization step may cause bit flips. It may destroy the previous CSE result in our proposed complexity-aware algorithm described in Section 3. Besides, no single SPT term can improve the error anymore, as depicted in Fig 6. The "always below" strategy (i.e. rounding to the nearest quantization level toward  $-\infty$ ) solve this problem. A compensation factor is used as the post-processing to normalize the mean quantization error to be zero.

Always using the nearest SPT to approximate the coefficients can significantly reduce the number of non-zero terms for an acceptable quantization error. But CSE of the resultant signeddigit coefficients is somewhat more complicated. The  $N \times (N-1)/2$ candidates to be first eliminated of the *N*-row coefficient matrix are doubled to consider the polarity of two rows. Fig 7(a) is an example for CSE on signed-digit coefficients.

	2's comp	signed digit		
	the nearest	toward -∞	the nearest	
0.25 (iteration error) 0.125 0.0625 0.03125	0.10000000	0.01000000 0.01100000 0.01110000 0.01110000	0.10000000	
0.015625	0.011 (locked)	0.01111100	0.10000 <u>1</u> 00	

Figure 6 Quantization of 0.484375

We propose the shifted CSAC in this paper for sparse coefficient matrices (such as the CSD-encoded coefficients), which considers the common subexpressions across the shifted versions of the coefficients. The shift amount is limited to reduce the search space and to constrain the truncation error during the arithmetic operations. The notation of the shifted CSAC is left aligned with the other term right shifted, such as  $x_0 - x_1 \gg 1$  shown in Fig 7(b), to simplify the hardware generation process. Besides, a row pair with shifted CSAC is searched only if its overall displacement is within the shift limit. Our experiment shows that  $\pm 2$ -bit shift with maximum 5-bit span is enough for most cases.



Figure 7 (a) CSAC in signed-digit coefficients (b) shifted CSAC

### **5** SIMULATION RESULTS

All ideal coefficients (IC) in our simulation are synthesized using the Parks-McClellan's method [5] with its passband and stopband frequencies at  $0.4\pi$  and  $0.6\pi$  respectively. The synthesized IC set are represented in the IEEE754 doubleprecision format. The optimal scale factor is explored between 0.7 and 1.4 for an octave about  $\pm 3$ dB gain tolerance. Note that the range is complete because the quantization results repeat when the IC's are scaled with a power-of-two factor.

Fig 8 shows the comparison of the different approximation strategies in Section 4 and demonstrates the effectiveness of our proposed shifted CSAC elimination. The IC set is for a 20-tap linear-phase FIR filter. First, the two dash lines show the square error versus the pre-defined adder budgets without CSE for the 2's complement (left) and the signed-digit (right) quantized coefficient (QC) sets respectively. The number of the allocated SPT terms is just one more than the given addition budget. For comparable filter responses, the nearest approximation approach with the signed-digit QC saves 37.88%~43.14% SPT budgets. The saving is even greater than the 29.1%~33.3% by CSE [4], which is shown as the solid line between.



Figure 8 Comparison of approximation strategies & CSE

Signed-digit CSE also helps to reduce the additions, but much less significantly. As shown in the figure, the two curves almost go in parallel as the budget decreases, which means no more common subexpressions are extracted and eliminated. The rightmost three curves are results from our proposed shifted CSAC elimination with different shift limits. For comparable filter responses, the proposed method saves  $10.34\% \sim 19.51\%$  addition budgets for the signed-digit QC set without CSE, while reducing  $49.06\% \sim 50.94\%$  budgets of the 2's complement one. CSAC with a shift limit  $\pm 2$  is enough for most cases.

Table 1 Comparison between CSD & SPT coefficients

		CSAC		Shifted CSAC (±2)				
taps	add	CSD*	SPT*	add	CSD*	SPT*		
12	23	8.817235	2.727223	21	5.084159	2.727223		
16	31	6.773190	3.696292	28	5.209612	3.835811		
20	39	5.645929	4.975382	33	17.641685	15.349970		
24	44	11.626458	20.547154	40	9.803638	17.781817		
28	53	18.317564	8.483186	48	7.218225	20.590703		
32	57	20.067199	15.768930	52	23.353057	17.632664		

<sup>(\*</sup>square error in the unit of  $10^{-10}$ )

Approximation with the nearest SPT has comparable coding performance with CSD [6], which has the theoretically minimum non-zero terms to represent a signed value. Table 1 summarizes the square error of the linear-phase low-pass FIR filters with different taps. For all SF candidates, the scaled NIC's are first rounded to the nearest 16-bit quantization levels and represented in the 2's complement form. The QC's are then converted into the CSD format for signed-digit CSE. Finally, the SF that results in the minimum number of additions is chosen. The second column in Table 1 lists that minimum numbers of adders with the elimination of CSAC (no shift), and the square errors are shown in the third column. The minimum adders are then used as the budgets for our proposed complexity-aware algorithm to quantize the coefficients and the resultant minimum square errors are shown in the fourth column. Similarly, the rest three columns show the simulation results for  $\pm 2$  shifted CSAC.

#### **6** CONCLUSIONS

This paper presents a complexity-aware algorithm to quantize the coefficients of FIR filters, which precisely distributes a predefined addition budget to the quantized coefficients by eliminating the common subexpressions. The proposed algorithm applies an optimal scale factor within the gain tolerance to settle the coefficients collectively into the quantization space. Our simulation shows only 49.06%~50.94% adders are required for similar square error. Moreover, the proposed algorithm has comparable performance with the CSD coefficients, which have the theoretically minimum non-zero terms. We have implemented a design automation tool that takes the double-precision FIR coefficients and a pre-defined budget as input to synthesize the minimum filter structure with a compensation factor. Automatic generation of the synthesizable RTL is still ongoing. Common subexpressions for transposedform FIR filters (also known as multiple constant multiplication; MCM [7]) and the optimal signed-digit representation for CSE will be studied in the future.

#### REFERENCES

- D. Li, Y. C. Lim, Y. Lian, and J. Song, "A polynomial-time algorithm for designing FIR filters with power-of-two coefficients," *IEEE Trans. Signal Processing*, vol. 50, pp.1935-1941, Aug. 2002
- [2] M. Mehendale, S. D. Sherlekar, VLSI Synthesis of DSP Kernels -Algorithmic and Architectural Transformations, Kluwer Academic Publishers, 2001
- [3] Y. Jang, and S. Yang, "Low-power CSD linear-phase FIR filter structure using vertical common subexpression," *Electronics Letters*, vol. 38, pp.777-779, July 2002
- [4] T. J. Lin, T. H. Yang, and C. W. Jen, "Area-effective FIR filter design for multiplier-less implementation," *International Symposium on Circuits and Systems (ISCAS)*, May 2003
- [5] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd Edition, Prentice Hall, 1999
- [6] K. K. Parhi, VLSI Digital Signal Processing Systems Design and Implementation, Wiley, 1999
- [7] M. Potkonjak, M. Srivastava, and A. Chandrakasan, "Multiple constant multiplication – efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Trans. Computer-Aided Design*, pp.151-165, Feb 1996