AN EFFICIENT DESIGN & IMPLEMENTATION OF FIR FILTER ON WALLACE TREE MULTIPLIER FOR LOW POWER AND HIGH SPEED OPERATIONS

[1]M.Swarna, PG Scholar , PVKK Engineering College, Anantapuram

[2]S.Raghavendra, Asst.Professor, PVKK Engineering College, Anantapuram

[3]S.Ravi Kumar, Asst.Professor, PVKK Engineering College, Anantapuram

ABSTRACT: *The most area and power consuming arithmetic operation in high-performance circuits like Finite Impulse Response (FIR), multiplication is one. There are different types of multipliers to reducing the cost and effective parameters in FIR filter design. Among those this paper use truncated multiplier and modified Wallace multiplier in the fir design. The structural adders and delay elements occupies more area and consumes power in this form so it was a reason to forward the proposed method. In prior FIR filters design with low cost effective results will done by the faithfully rounded truncated multipliers with the carry save additions. In MCMAT design the low cost FIR filters within the best area and power results are implement in this paper by using the improved truncated methods. Along with that the proposed method modified Wallace multiplier based fir filter is also designed in this paper to make the fir filter design is suitable for low power applications*.

I.INTRODUCTION

Multiplication operation is one of the area which consuming more arithmetical operations in high performance circuits. As for importance many of the researchers deal with high speed multipliers of low power design. Multiplication operation contains two basic operations, one to generate partial products and another one to generate their sum and this performed using two types of multiplication algorithms, parallel and serial. Whereas the Serial multiplication algorithms use sequential circuits with feedbacks, whereas the inner products are sequentially produced and the computed. Parallel multiplication algorithms often use combinational circuits and these never contain any feedback structures.

Multiplication operation of two bits output which twice of the original bit. It is necessary to truncate the partial product to the required precision to minimize the area cost. Fixed-width multipliers, a subset of truncated multipliers, contain only n most significant bits i.e. MSB bits of the 2n-bits product for n × n

multiplication operation and use extra correction circuits to minimize the truncation errors. In previous papers, to minimize the truncation error by where a multiplier plays a vital role include digital filtering, digital communications and spectral analysis adding the error compensation circuits. So that the output will be described. In this method combine the truncation, tree reduction and rounding of the Partial Product bits during the design of fast parallel truncated multipliers so that the final truncated product satisfies the precision requirement. In this way truncation error is not more than 1ulp i.e. one unit of least position, so there is no need to go on error compensation circuits, and the output will be derived.

2.LITERATURE SURVEY

2.1. Introduction

In most digital signal processing (DSP) systems, a multiplier operation is one of the important hardware blocks. Some of the Signal Processing applications where a multiplier plays a vital role include digital filtering, digital communications and spectral analysis. Many current DSP applications are targeted at the portable battery operating systems, so as the primary design constraint the power dissipation become.

A multiplication is very expensive and slows the overall operation. Let us consider two unsigned binary numbers as X and Y and their bit length as M and N bits for them. For multiplication operation, it is useful to give X and Y values in the binary format.

$$X = \Sigma\, X_i\, 2^i \quad i = 0 \text{ to } M \tag{5.1}$$

$$Y = \Sigma\, Y_j\, 2^j \quad j = 0 \text{ to } N \tag{5.2}$$

$$
\begin{aligned}
Z = X \times Y \;&=\; \Sigma\, Z_k 2^k \quad k = 0 \text{ to } M+N-1 \\
&=\; (\Sigma\, X_i\, 2^i \; i = 0 \text{ to } M)\,(\Sigma\, Y_j\, 2^j \; j = 0 \text{ to } N) \\
&=\; \Sigma\,(\Sigma\, X^i\, Y^j\, 2^{i+j})\, i = 0 \text{ to } M\text{-}1, j = 0 \text{ to } N\text{-}1
\end{aligned}
$$

The use of a single two input adder is a simplest way to perform a multiplication. The multiplication tasks M cycles for inputs that are M and N bits wide. This shift and addition of PP algorithm for multiplication adds together M partial products. It is very easy method to perform binary multiplication between them than decimal multiplication. In the value of each digit of a binary number can only be either 0 or 1 that is depending on the multiplier bit length, If the value is 0 the partial products can take as the copy of bits of the multiplicand. In digital logic circuits, this is simply an AND logic gate functionality. One off the speed multiplication operation is to resort to an approach similar to the manually computing of a multiplication process. The entire partial product are generated at the same time and organized in an array. A multi-operand addition is applied to compute the final product.

The process is shown in the below figure. This is set of operations that can be able map directly into hardware design then resulting structure is called array multiplier.

So the adder makes vital role in the designing any multiplier operation (John Rabaey (2006)).There are different methods of adders and their operations were discussed by various researchers like Oklobdzija.V.G in 1995,Pucknell in 2004, Shalem.R in 1999 and Zimmermann.R and Fichtner.W in 2007 from all of these approaches we came to learn that the new improved 14 Transistor having full adder blocks shows better result in the Threshold loss problem, power omission and speediness by designing using MOS transistor count.

In this approach, the given four important types of multiplier operations are there those are Array, Baugh wooly, Braun and Wallace tree, all of these are implemented using different types of adder blocks presented in Shalem.R, then after we find out the better one in the operation performance like power, speed and area.

3.PROPOSED METHOD

In all Digital Applications Filter is a frequency selective network. It passes only a limited band of frequencies signal while limit the others signals. Filters are divided as either analog or digital based on nature of inputs and outputs of a circuit. Filters are further divided based on its impulse response as finite impulse response filters and infinite impulse response filters.

3.1 Linear phase FIR filters

The linear phase defines the real that the phase response of the filter operation is a straight line function of many frequency signals. This means that we define that as the delay given through the filter will be the same at all frequencies signals are as a result, the filter does not cause phase distortion or the delay distortion, which is major advantage over IIR filter and analogue filters in certain applications approaches. An FIR filter **3.2 Different forms in linear phase even order filters**

is linear phase if and only if its co efficient are symmetrical that means similar around the center coefficients i.e. the first coefficient in the approach is the same as the last co efficient in the same approach and the second is the same as the second last in the approach etc.

There are two basic FIR structures, direct form and transpose form, for a linear phase operation approach even order FIR filter.

Direct form

In the direct form in the multiple constant of multiplication operation (MCM) or

accumulation (MCMA) method performs the serial multiplication operation of individual delayed signals and respective filter coefficients operation approach, followed by accumulation approach of all the products in the result. Thus, the operands of the multipliers in MCMA are delayed input signals coefficients. For FIR filter implementation the important thing is the derived the bit widths for filter coefficients in the operation approach, which has direct performance on the area cost of arithmetic units and registers. In this design moreover, since the bit widths in the operation approach after multiplications grow, many Digital Signal Process requirements do not need full precision outputs values .Instead of to generate exact expected rounded outputs where the total error introduced in quantization operation and rounding process is no more than one unit of the last place (ULP) defined as the weighting of the least significant bit that is LSBvalue of the outputs.

Low cost finite impulse response filter designs are given using the concept of expected exact rounded truncated multipliers. Whereas the non-uniform coefficient quantization value with the proper filter order is given to reducethe total area cost.

The Multiple constant process of multiplication or accumulation in a direct FIR structure is design using anadvanced version of

truncated multipliers operation. Comparisons with previous different multipliers approaches show that the proposed designs achieve the best area and power results.

We compare the different multipliers with FIR filter form to reduce the parameters area power and delay. In these fir form when we use common multiplier we tackle the effective parameters in the design of chips with the worst case, but by the common multipliers we didn't approach that levels of low values in the values of effective parameters. So we move to the different multipliers logics to achieve the values as low as possible with the different techniques in the multiplication methods.

Truncated multiplication operation is a method of approach where only the most significant columns of the multiplication matrix operation are used and therefore area requirements can be reduced. In Truncation is a method where the least significant columns in the partial product matrix are not formed.

The method of truncation will follows some steps in the process of multiplying of the partial product bits in the multiplier by the adders.

The three steps involved in method are

1. Deletion,
2. Truncation

3. Rounding.

Deletion:

In truncated multiplier operation we always start with the multiplication process only. In the PP bits values generation we delete the more than half of the bits, then remaining bits are became the partial products in the given process. This is the main process of deletion in the circuitry.

Truncation:

Truncation is a method where the least significant columns in the partial product matrix are not formed. The amount of columns not formed in this way, T, defines the degree of truncation and the T Least Significant Bits (LSB) of the product always results in 0. The algorithm behind fixed width multiplication is the same as when dealing with non-fixed width multiplication regardless of the truncation degree.

Rounding:

Generally as we discussed an n bit width multiplicand and an n bit width multiplier would render a 2n-bit product.. By the rounding process helps in the obtain of the faithfully rounded value.By these steps the truncated multiplier will gives the faithfully rounded values after truncate of the least significant part in the result. Truncated multiplication operation gives an efficient method operation for decreasing the power dissipation and area of rounded parallel multiplier.

When the error occurs at the step of deletion we got the deletion error(E_d), then the error occurs at the step of truncation it defined as the truncation error(E_t), and the error occurs at the final step of rounding process call it as the rounding error(E_d). Those three types of errors combined and it gives the whole error of truncated multiplier as E.

$$E_{r\_low} = -\left((r-1) \cdot 2^r + 1\right) \cdot 2^{-r-k} \text{ ulps}$$

When we consider the 8×8 bit width truncated multiplier operation approach , where r term refer as the number of no formed columns and k term refer as the number of columns that are occurred but removed in the final result of multiplication operation. This error ranges from Er low, which occurs when each of the unformed partial product bits is a '1'. Since Er high is zero, the range of the reduction error,in the the range of reduction error is −1.502 ulps≤ Er≤ 0 ulps. the comparison process for the error due to rounding a product to nearest has a range of −0.5 ulps<Ernd≤ 0.5 ulps.

$$-\text{ulp} \leq E'_D \leq 0 \qquad -\frac{1}{2}\text{ulp} \leq E_D = E'_D + \frac{1}{2}\text{ulp} \leq \frac{1}{2}\text{ulp}$$

$$-\text{ulp} < E'_R \leq 0 \qquad -\frac{1}{2}\text{ulp} < E_R = E'_R + \frac{1}{2}\text{ulp} \leq \frac{1}{2}\text{ulp}$$

$$-\text{ulp} < E = (E_D + E_R) \leq \text{ulp}.$$

In the structure of FIR filter the truncated multipliers are placed the values estimation is different while compare with the other multipliers.

3.3 Truncated multiplier with transposed form of FIR

The transposed form of the FIR filters constructed by the multipliers, delay elements, different structural adders performed by the arithmetic operations. In these FIR forms we possess the different multipliers but we are using the truncated multipliers for the reduction of effective parameters like area, power and delay. the prior designs are used these transposed but the power parameters are not satisfied with the huge bit widths in the multipliers. But consideration of area the transposed form posses' nearly satisfied results.

3.4.Implementation of transposed FIR form with truncated multiplier
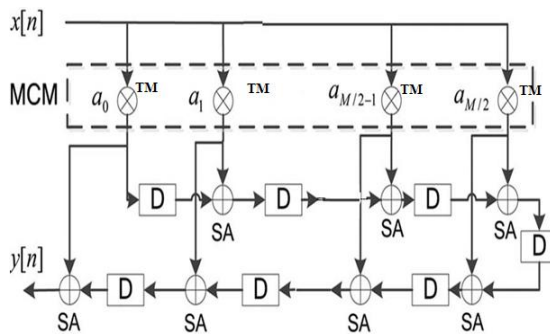
Fig 3.1: Implementation of transposed FIR form with truncated multiplier

The transposed forms are constructed with the truncated multipliers in the first stage. The filter input is directly forwarded to the multipliers and the multipliers process forwarded with the constant coefficients. by these inputs and coefficients the truncated multiplier values evaluated ,those values forwarded to the delay elements and after delay is introducing the next multiplier output also merged with the structural adder and it acts as one of the input and second input is taken from the delay elements. At the final structural adders we got the filter output in the transposed FIR structure.

3.5. Truncated multiplier with direct form of FIR

The direct form of the FIR filters constructed by the delay elements, adders performed by the arithmetic operations,at the middle of stage the multipliers performance is required as fast as possible. In these FIR forms we possess the different multipliers but we are using the truncated multipliers for the reduction of effective parameters like area, power and delay.

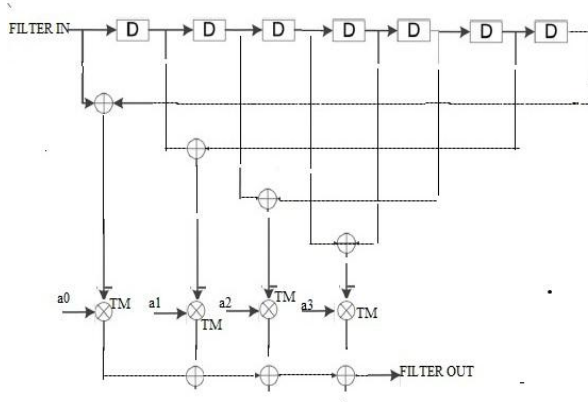### 3.6. Implementation of direct FIR form with truncated multiplier

Fig 3.2: Implementation of direct FIR form with truncated multiplier

In the implementation of direct form we used the delay elements, the filter input will gives to the delay elements for the introducing the delay in the operation of sequence. With the delay elements the input value just forwarded to the next delay elements.

3.7. Proposed system description

we invoke the proposed system in the place of the truncated multiplier which was existing and it is replaced by the Wallace tree multiplier nothing but modified wallace multiplier,then we proposed with the effective parameters comparision of area,delay,power in between proposed and existing system.

Process flow of Wallace multipliers

⮑ Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding N results. Reduce the

number of partial products to two layers of full adders.

⮑ where a multiplier plays a vital role include digital filtering, digital communications and spectral analysis.

⮑ Group the wires in two numbers, and add them with a conventional adder.
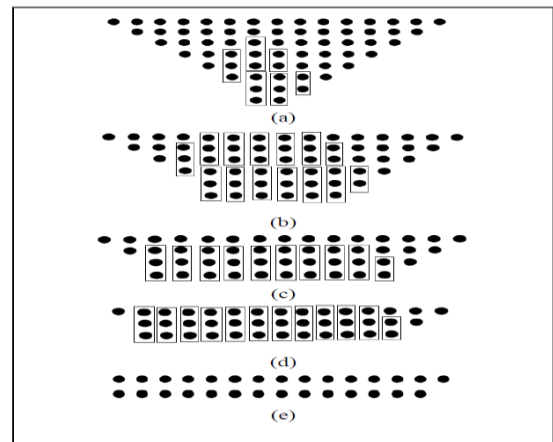

Fig 3.3:Process flow for 8x8 Wallace multiplier

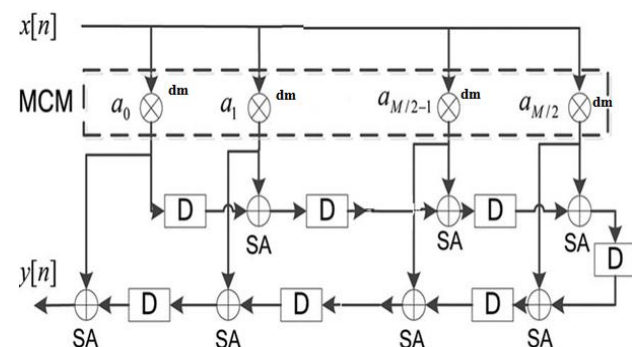### 3.8. Implementation of transposed FIR form with Wallace tree multiplier

Fig 3.4:Implementation of transposed FIR form with Wallace tree multiplier

The transposed forms are constructed with the Wallace tree multipliers in the first stage. The filter input is directly forwarded to this Wallace tree multipliers and the multipliers process will be forwarded with the constant coefficients. By these inputs and given coefficients to the Wallace tree multiplier values evaluated, those values forwarded to the delay elements and after delay is introducing the next multiplier elements.

3.9. Implementation of direct FIR form with Wallace tree multiplier
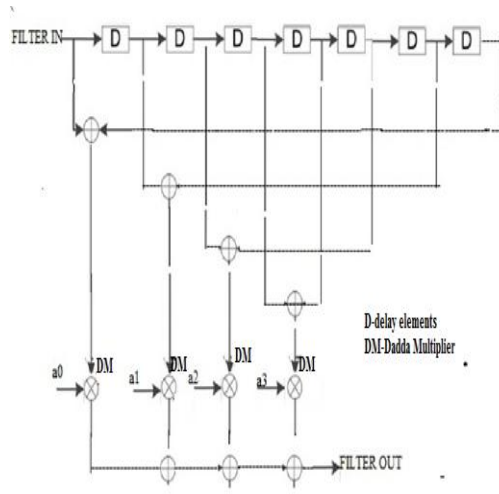
Fig 3.5: Implementation of direct FIR form with Wallace tree multiplier

In the implementation of direct form we used the delay elements, the filter input will gives to the delay elements for the introducing the delay in the operation of sequence. With the delay elements the input value just forwarded to the next delay elements. in these the input and the final delay elements are summed with the arithmetic operations ,this process will continues with the 1st delay element and 6th delay element,2nd delay element and 5th delay element, like these the operation process is completed with 3rd element and 4th element. After the addition operations it will moves to Wallace tree multipliers and the multiplication process will be starts here and the Wallace tree output will be evaluated with the perfect value.

4.SIMULATION RESULTS
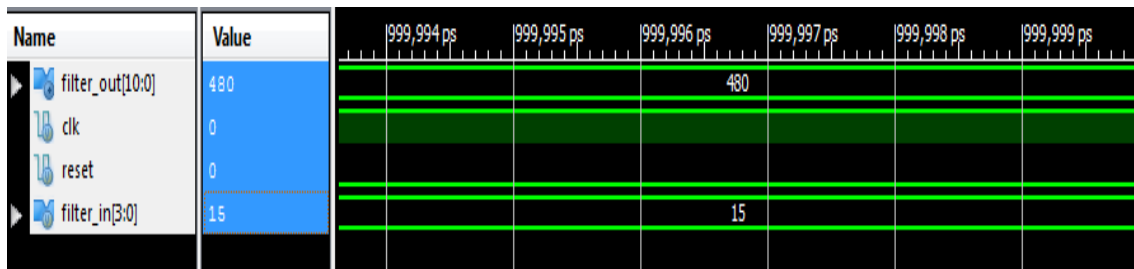
Direct form with truncated multiplier

4bit

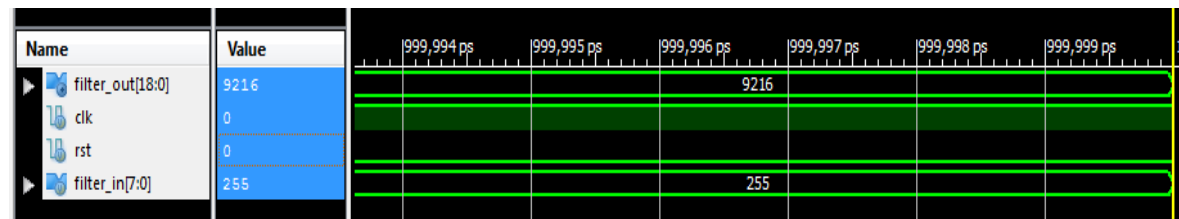**Fig 4.1: 4-Bit Direct form with truncated multiplier**

8bit

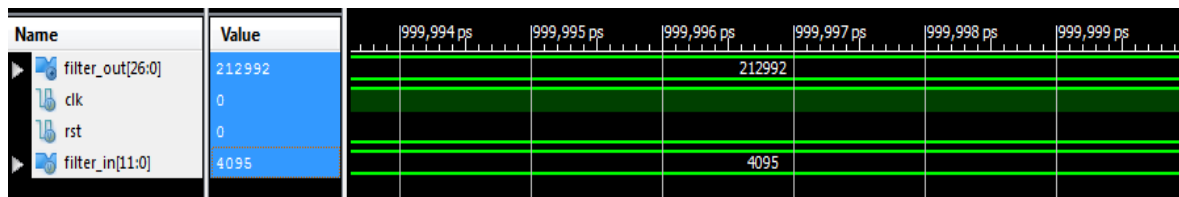**Fig 4.2: 8-Bit Direct form with truncated multiplier**

12bit

Fig 4.3: 12-Bit Direct form with truncated multiplier

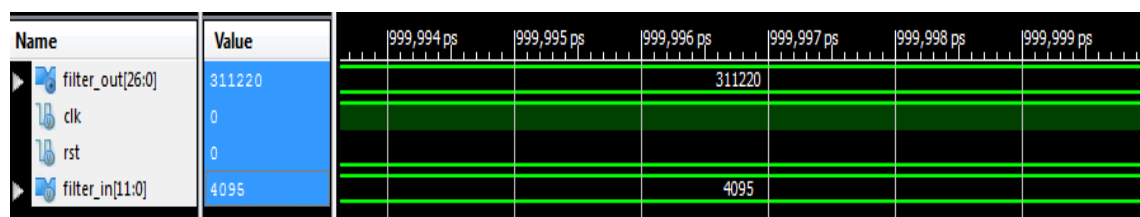**Direct form with Wallace tree multiplier**

12 bit

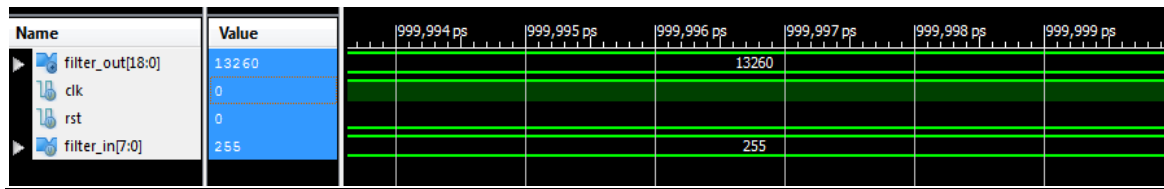**Fig 4.4: 12-Bit Direct form with Wallace tree multiplier**

8 bit

Fig 4.5: 8-Bit Direct form with Wallace tree multiplier
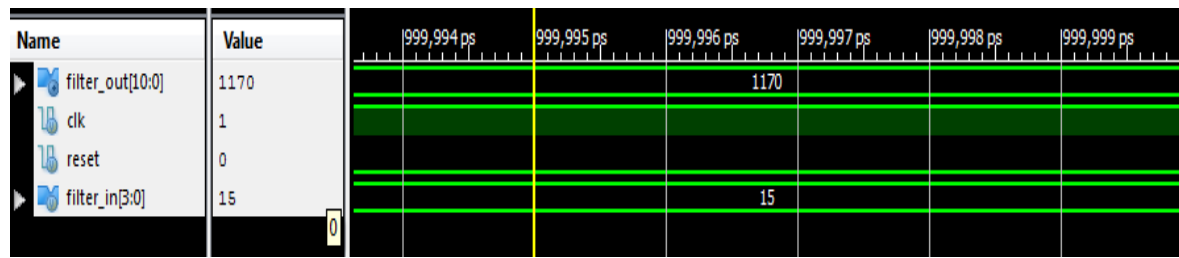
4-Bit

Fig 4.6: 4-Bit Direct form with Wallace tree multiplier

COMPARISON OF THE PARAMETERS BETWEEN THE FIR FORMS

Selected Device: spartan3E 3s500efg320-4

Truncated multiplier

Direct form	*Memory (in kbs)*	*Delay (in ns)*	*Power (in mw)*
12 bit	*259704*		**10.95mw**
8bit	*231416*	*4.13*	*5.633*
4bit	*187512*	*10.86*	*0.114*

Table 4.1: Truncated multiplier

Wallace tree multiplier

Direct form	*Memory (in kbs)*	*Delay(in ns)*	*Power (in mw)*
12bit	*216504*		**5.299mw**
8bit	*231992*	*4.04*	*2.144*
4bit	*187768*	*4.04*	*0.872*

Table 4.2: Wallace tree multiplier

When we do the analyzation of forms of FIR the transposed possess the memory results as low as possible. But coming to the power analyzation results the transposed form does not giving effective resuts to decrease the cost of FIR filters. In the anlyzation of parameters in the direct form we got the effective results in terms of power and delay based results. The area also in the satisifactable range value in the direct form.

CONCLUSION

We compare both the truncated multiplier and Wallace tree multiplier in the both forms of FIR forms by observing the parameters of the Wallace tree multiplier it realizes the area minimized factor.

While we compare the direct forms of the both multipliers the area factor is reducing in the Wallace tree multiplier than the truncated multiplier. Then we proposed Wallace tree multipliers under the area or memory utilization factor in the direct form than the truncated multiplier in this form.

Finally we conclude with the comparison of parameters in different forms of FIR structures with the different multipliers, for the low cost FIR filters this brief reference of parameter results in between the different forms with different multipliers has presented low-cost FIR filter designs by jointly considering the optimization of coefficient bit width and hardware resources in implementations without sacrificing the frequency response.

BIBILOGRAPHY

[1] M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. vol. 49, no. 3, pp. 196–203, Mar. 2002.

[2] C.-H. Chang, J. Chen, and A. P. Vinod, "Information theoretic approach to complexity reduction of FIR filter design," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 8, pp. 2310–2321, Sep. 2008

. [3] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution—A new approach to versatile subexpressions sharing in multiple constant multiplications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 2, pp. 559–571, Mar. 2008.

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

[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 Integer. Circuits Syst., vol. 21, no. 12, pp. 1525– 1529, Dec. 2002

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

. [7] O. Gustafsson, "Lower bounds for constant multiplication problems," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 11, pp. 974–978, Nov. 2007.

[8] Y. Voronenko and M. Puschel, "Multiplierless multiple constant multiplication," ACM Trans. Algorithms, vol. 3, no. 2, pp. 1–38, May 2007.

[9] D. Shi and Y. J. Yu, "Design of linear phase FIR filters with high probability of achieving minimum number of adders," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 1, pp. 126–136, Jan. 2011.