



Efficient LUT Based Filter Design Using Approximate Algorithm

Bathina Rajeswari*¹, G S S Prasad²

M.Tech Scholar, Department of ECE.¹

Asst.Prof, Department of ECE, Kakinada Institute of Engineering &Technology
East Godavari Dist., AP, India.²

Abstract— The filter design optimization (FDO) issue is described as finding a strategy of channel coefficients that yields a channel format with scarcest whimsies, fulfilling the channel destinations. It has gotten a mammoth energy by virtue of the regardless of what you look like at it usage of channels. Enduring that the coefficient duplications in the channel course of action are perceived under a move joins fabricating, the adaptable quality is for the most part depicted like the aggregate number of adders and subtractors. In this paper, i exhibit a privilege FDO is upgraded by utilizing Look UP Table approach. APC and OMS paired properties are utilized for actualizing LUT based FIR channel by utilizing APC And OMS decreased the quantity of capacity components right around half. This technique has better execution and it devours less power and possess less space contrast with FDO strategy.

Keywords: *Anti Symmetric Product Coding (APC), Odd multiple Storage (OMS), Look Up Table (LUT), Finite Impulse Response(FIR).*

I. INTRODUCTION

Filter is a frequency selective network. It passes a band of frequencies while constricting the others. Channels are delegated simple and computerized relying upon nature of sources of info and yields. Channels are additionally delegated limited drive reaction and unbounded motivation reaction channels relying upon motivation reaction. This part gives a brief about the sorts of channels.

Computerized channels are utilized broadly in every aspect of electronic industry. This is by virtue of mechanized channels can finish much best banner to upheaval extents over basic channels and at each most of the way arrange the straightforward channel adds more uproar to the banner, the propelled channel performs calm logical operations at each direct walk in the change. The propelled channels have ascended as a strong decision for clearing bustle, shaping

reach, and restricting between picture deterrent in correspondence structures. These channels have turned out to be prevalent in light of the fact that their exact reproducibility permits configuration designers to accomplish execution levels that are hard to get with simple channels.

Digital Filters can be constructed from 3 fundamental mathematical operations.

- Addition (or subtraction)
- Multiplication (normally of a signal by a constant)
- Time Delay i.e. delaying a digital signal by one or more sample periods

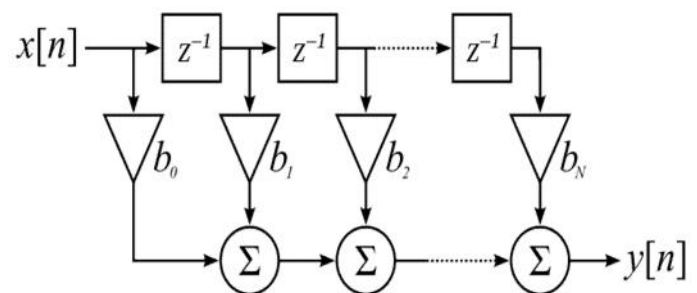


Figure 1: Block diagram of a Simple Digital Filter

Figure 1 shows a graphical means of describing a digital filter whereby the behavior of the filter is described by using the mathematical operations mentioned above.

The Impulse Response of a digital filter, $h(n)$ is the response of the filter to an input consisting of the unit impulse function, (n) . If the impulse response of a system is known, it is possible to calculate the system response for any input sequence $x(n)$. By definition, the unit impulse is applied to a system at sample index $n=0$. So, the impulse response is non-zero only for values of n greater than or equal to zero i.e. $h(n)$ is zero for $n < 0$. This impulse response is said to be causal otherwise the system would be producing a response before an input has been applied. It is known from the time-invariance property of a Linear Time Invariant System that the response of a system to a delayed unit impulse $(n-k)$ will be a delayed version of the unit impulse, i.e. $h(n-k)$. It is also known from

the linearity property that the response of a system to a weighted sum of inputs will be a weighted sum of responses of the system to each of the individual inputs. Therefore, the response of a system to an arbitrary input $x(n)$ can be written as follows:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

Fast multipliers are fundamental parts of cutting edge signal getting ready systems. The speed of copy operation is of uncommon essentialness in cutting edge hail planning and furthermore in the extensively helpful processors today, especially since the media taking care of took off. In the past growth was generally executed by methods for a course of action of extension, Subtraction, and move operations. Increment can be considered as a movement of repeated increases. The number to be incorporated is the multiplicand, the amount of times that it is incorporated is the multiplier, and the result is the thing. Every movement of extension makes a fragmented thing. In numerous PCs, the operand generally contains a comparable number of bits. Right when the operands are deciphered as entire numbers, the thing is generally twofold the length of operands with a particular ultimate objective to spare the information content. This repeated extension procedure that is proposed by the math definition is direct that it is regularly supplanted by an estimation that makes usage of positional depiction. It is possible to break down multipliers into two segments. The underlying fragment is dedicated to the period of fragmentary things, and the second one accumulates and incorporates them.

The basic duplication rule is twofold, i.e. evaluation of fragmentary things and gathering of the moved inadequate things. It is performed by the dynamic Addition's of the fragments of the moved inadequate thing framework. The "multiplier" is adequately moved and entryways the best possible bit of the 'multiplicand'. The delayed, gated event of the multiplicand should all be in a comparative fragment of the moved midway thing structure. They are then added to outline the thing bit for the particular shape. Increase is along these lines a multi operand operation. To extend the growth to both checked and unsigned numbers, an accommodating number structure would be the depiction of numbers in two's supplement arrange. Furthermore, it is generally the most domain eating up. Consequently, streamlining the speed and region of the multiplier is a vital arrangement issue. Regardless, region and speed are typically conflicting

objectives with the objective that upgrading speed comes to fruition for the most part in greater areas. Consequently, whole scopes of multipliers with different region speed necessities are made with totally parallel planning. Amidst are digit serial multipliers where single digits including a couple of bits are dealt with. These multipliers have coordinate execution in both speed and locale. In any case, existing digit serial multipliers have been tormented by confounded trading structures or possibly variations from the norm in design. Radix 2^n multipliers which chip away at digits in a parallel plan instead of bits pass on the pipelining to the digit level and keep up a vital separation from most of the above issues. They were introduced by M. K. Ibrahim in 1993. These structures are iterative and isolates. The pipelining done at the digit level brings the benefit of reliable operation speed paying little mind to the measure of the multiplier. The clock speed is recently controlled by the digit evaluate which is starting at now settled before the layout is executed.

$$Y[n]=b_0x[n]+b_1x[n-1]+b_2x[n-2]+b_3x[n-3]$$

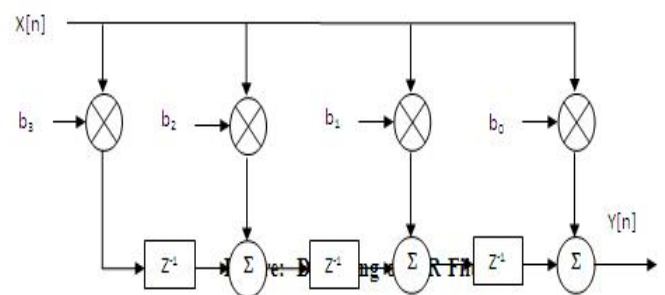


Figure 2: FIR multiplier

II. MULTIPLIERLESS DESIGN:

ECHO-A AND ECHO-D:

To realize the MCM block of the transposed form with the minimum number of adder-steps, in SIREN and NAIAD, we respectively used the modified versions of the approximate algorithms of [9] and [3] that can handle the delay constraint. Whenever a set of fixed point filter coefficients is determined in SIREN and NAIAD, the minimum adder-steps of coefficients is computed as given in Section II-B-1 and it is given to the algorithms of and as a delay constraint. In order to target the direct form of the FIR filter, in SIREN and NAIAD, ECHO-A is used to compute the smallest number of operations in the CAVM block and ECHOD is used for the design of the CAVM block with a

small number of adder-steps. Note that in direct form filters, the total number of operations in the filter, i.e., TA, is determined by the solution of ECHO-A or ECHO-D on the set of filter coefficients. The proposed methods can target different filter constraints. For example, when the lower and upper bounds of ϵ and δ in (4) are set to 1, the filter constraints of ϵ and δ are aimed. Setting ϵ and δ respectively to 0.7 and 1.4 corresponds to the 3 Db gain tolerance in the filter design. The proposed algorithms can also target asymmetric filters taking into account the related filter constraints. The proposed algorithms can target the optimization of the gate-level area of the filter design. In this case, whenever a set of coefficients is found, an algorithm, that can find the shift-adds design of the multiplier block of the filter occupying minimum area, should be used. In the transposed form filter, the size of registers and adders in the register add block should also be considered.

The APC approach, disregarding the way that giving a decreasing in LUT measure by a factor of two, unites significant overhead of range and time to play out the two's supplement operation of LUT yield for sign change and that of the data operand for input mapping. Regardless, it is found that when the APC approach is merged with the OMS system the two's supplement operations could be particularly unraveled since the data address and LUT yield could basically be changed into odd integers.¹ However, the OMS technique can't be joined with the APC plot, since the APC words delivered by are odd numbers. Likewise, the OMS plot in does not give a capable utilization when joined with the APC system. In this brief, a substitute kind of APC is given joined a changed sort of the OMS plot for compelling memory based duplication.

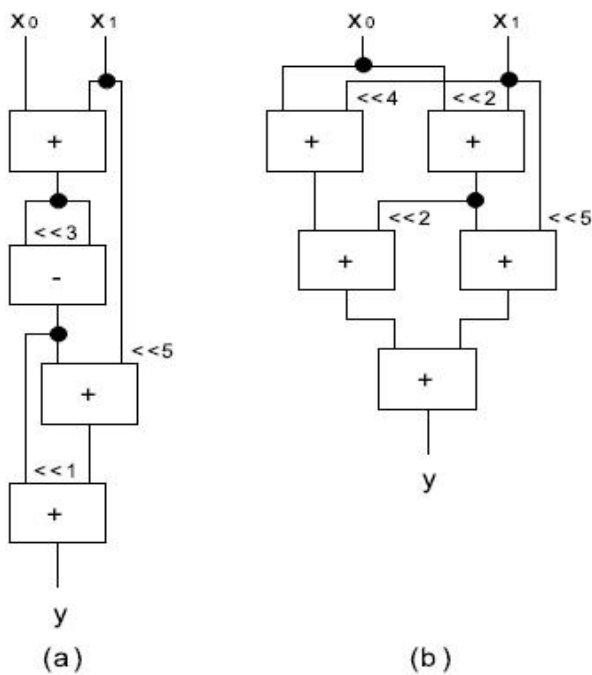


Figure 3: Multiplier less realization of constant multiplications using the DBR technique

III. LUT BASED FILTERS DESIGN:

Another approach to manage LUT arrangement is shown, where quite recently the odd results of the settled coefficient are required to be secured, which is suggested as the OMS. In like manner, by the APC approach, the LUT size can moreover be diminished to half, where the thing words are recoded as antagonistic to symmetric sets.

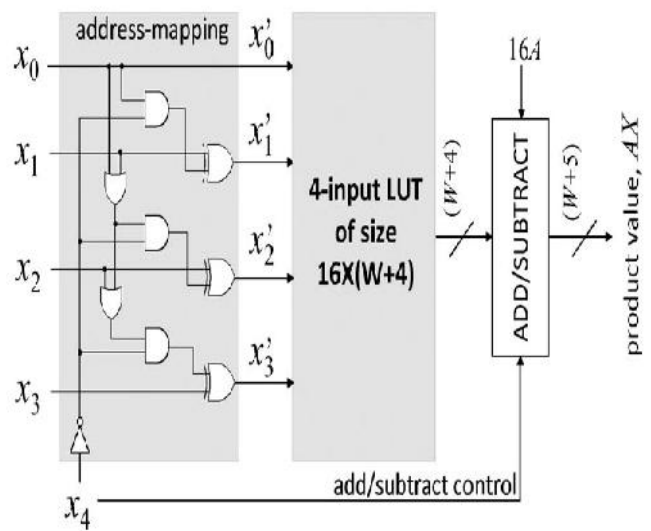


Figure 4: Proposed APC Part

APC WORDS FOR DIFFERENT INPUT VALUES FOR $L = 5$

Input, X	product values	Input, X	product values	address $x'_3x'_2x'_1x'_0$	APC words
00001	A	11111	31A	1111	15A
00010	2A	11110	30A	1110	14A
00011	3A	11101	29A	1101	13A
00100	4A	11100	28A	1100	12A
00101	5A	11011	27A	1011	11A
00110	6A	11010	26A	1010	10A
00111	7A	11001	25A	1001	9A
01000	8A	11000	24A	1000	8A
01001	9A	10111	23A	0111	7A
01010	10A	10110	22A	0110	6A
01011	11A	10101	21A	0101	5A
01100	12A	10100	20A	0100	4A
01101	13A	10011	19A	0011	3A
01110	14A	10010	18A	0010	2A
01111	15A	10001	17A	0001	A
10000	16A	10000	16A	0000	0

For $X = (0000)$, the encoded word to be stored is 16A

Figure 5: Stored APC Words

$16 \times (W+4) \rightarrow 16$ Locations and each location having $(W+4)$ bits.

. Let the product values on the second and fourth columns of a row be u and v , respectively. Since one can write $u = [(u + v)/2 - (v - u)/2]$ and $v = [(u + v)/2 + (v - u)/2]$, for $(u + v) = 32A$, $U=16A+ [(V-U)/2]$
 $V=16A-[(V-U)/2]$

The product values on the second and fourth columns of above figure therefore have a negative mirror symmetry.

This behavior of the product words can be used to reduce the LUT size, where, instead of storing u and v , only $[(v - u)/2]$ is stored for a pair of input on a given row. The 4-bit LUT addresses and corresponding coded words are listed on the fifth and sixth columns of the table, respectively. Since the representation of the product is derived from the antisymmetric behavior of the products, we can name it as antisymmetric product code. The 4-bit address $X_ = (x_3, x_2, x_1, x_0)$ of the APC word is given by

$$X'_i = \begin{cases} X_i, & \text{if } x_i = 1 \\ X'_i, & \text{if } x_i = 0 \end{cases}$$

IV. RESULTS

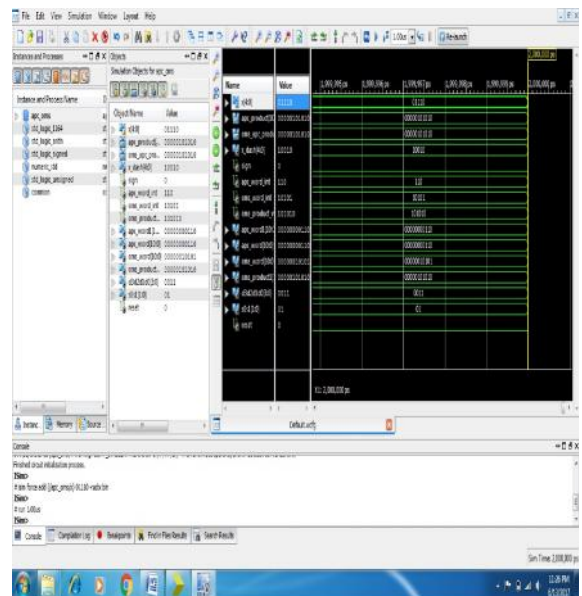


Figure 6: Experimental Result

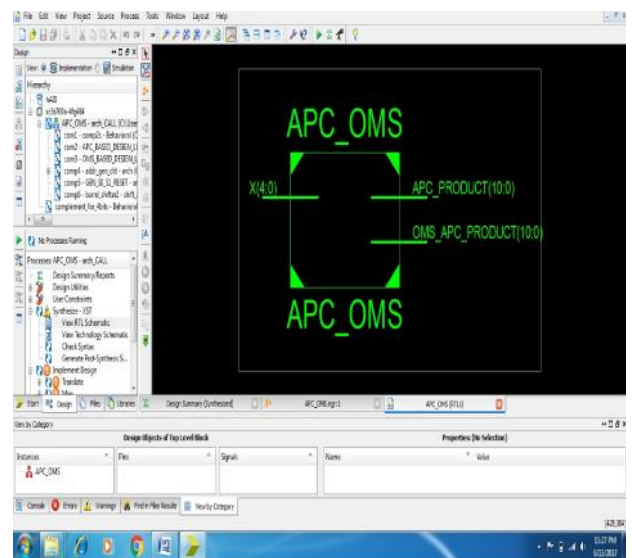


Figure 7: RTL Schematic

V. CONCLUSION

This article addressed the problem of optimizing the number of operations in the FIR filter design while satisfying the filter constraints, generally known as the FDO problem. It presented exact and approximate FDO algorithms, all of which are equipped with efficient methods to find the fewest operations in the shift-adds design of the coefficient multiplications. Moreover, it showed how these algorithms can be modified to target different filter constraints and filter

forms and to handle a delay constraint in the multiplier blocks of filters. It was observed that the exact FDO method can handle filters with a small number of coefficients, on which approximate FDO methods can find solutions very close to the minimum. It was also shown that heuristic methods are indispensable for filters with a large number of coefficients, on which the proposed approximate method can find better solutions in terms of the number of operations than prominent FDO algorithms. It was indicated that the total number of operations, EWL value, filter length, quantization value, and filter form have a significant impact on the gate-level area, delay, and power dissipation results of filter designs. Finally, area reduction using LUT approach is proposed for further improvement.

VI. REFERENCES

- [1] L. Wanhammar, *DSP Integrated Circuits*. New York, NY, USA: Academic, 1999.
- [2] H. Nguyen and A. Chatterjee, "Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 4, pp. 419–424, 2000.
- [3] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication," *ACM Trans. Algorithms*, vol. 3, no. 2, 2007, doi: 10.1145/1240233.1240234.
- [4] P. Cappello and K. Steiglitz, "Some complexity issues in digital signal processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 5, pp. 1037–1041, 1984.
- [5] R. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Circuits Syst. II*, vol. 43, no. 10, pp. 677–688, 1996.
- [6] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on minimal signed digit representation," in *Proc. Design Autom. Conf.*, 2001, pp. 468–473.
- [7] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Exact and approximate algorithms for the optimization of area and delay in multiple constant Multiplications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 6, pp. 1013–1026, 2008.
- [8] A. Dempster and M. Macleod, "Use of minimum adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II*, vol. 42, no. 9, pp. 569–577, 1995.
- [9] L. Aksoy, E. Gunes, and P. Flores, "Search algorithms for the multiple constant multiplications problem: Exact and approximate," *Elsevier J. Microprocessors Microsyst.*, vol. 34, no. 5, pp. 151–162, 2010.
- [10] K. Johansson, O. Gustafsson, and L. Wanhammar, "A detailed complexity model for multiple constant multiplication and an algorithm to minimize the complexity," in *Proc. IEEE Eur. Conf. Circuit Theory Design*, 2005, pp. 465–468.
- [11] H.-J. Kang and I.-C. Park, "FIR Filter Synthesis Algorithms for Minimizing the Delay and the Number of Adders," *IEEE Tran. on Circuits and Systems II*, vol. 48, no. 8, pp. 770–777, 2001.
- [12] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Finding the Optimal Tradeoff Between Area and Delay in Multiple Constant Multiplications," *Elsevier Journal on Microprocessors and Microsystems*, vol. 35, no. 8, pp. 729–741, 2011.