

Different Distributed Arithmetic Multiplication Schemes used in FIR Filter

P.Sritha, R.S.Valarmathi, P.Ramya

Abstract: Filters are the most necessary elements of the DSP application systems. Multiply Accumulate Unit (MAC) is the major block in the FIR filter, because of its operation. The complexities of the MAC unit can be reduced by several reduction techniques such as Constant Multiplications (CM) and Distributed Arithmetic (DA) methods. Distributed Arithmetic method replaces the MAC operation by Pre-computed results stored in Look Up Tables. Many DA based techniques have implemented in FIR filter for the reduction of area. This paper provides detail about various techniques of Distributed arithmetic methods implemented in FIR filter.

Keywords: Multiply Accumulate Unit, Distributed Arithmetic, and Look Up table.

INTRODUCTION

Finite Impulse response filter is a digital filter, whose impulse response is of finite duration [1]. FIR filters does not require any feedback, so the filter is known as non-recursive filters. The output function of the FIR filter is given by

$$Y(n) = X(n) * H(n)$$

For N order filter the each response of output sample is the summation of current values of the input samples.[1]

$$Y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) = \sum_{k=0}^{M-1} x(k)h(n-k)$$

Here, $x(n)$ is the output sequence, $h(n)$ represents the coefficients samples of filter and $Y(n)$ is the output response [1]. FIR filters are the essential structural blocks in the signal processing for removing noise from the original sound and it has various advantages like stability, linear phase response, and regular structure[2]. Optimization of area, power, and delay plays a major role in FIR filters, since it is used in VLSI signal processing applications and communication [2]. Multipliers are the major blocks in the FIR filter because major operations and delay took place in this particular block. The execution speed of the FIR filter is decided by the employed multipliers [3]. Thus the performance of the filter is concluded by the types of multiplication schemes employed in the FIR filter. The output of the multiplication response of filter is determined by the product of inputs and the coefficient of the filter. This Paper Examines the

Various proposed Distributed Arithmetic method for reduction of area, delay and power. The detailed description of DA method and modified DA is given in section I. Section II describes the Offset Binary Coding method. The Adaptive filter Based DA technique is described in section III.

I. DISTRIBUTED ARITHMETIC METHOD-OVERVIEW

Multipliers are the complex and time consuming process in the FIR filter. In order to remove the redundancy of multiplication process, several multiplier-less method have evaluated. This multiplier less method is classified into two types namely methods based on conversion and methods based on memory. In conversion based method, the coefficients of the FIR filter are converted into other numerical forms other than the binary forms for effective hardware implementation and reduced delay. Canonical Signed Digit [4] is the numerical form, in which the coefficients are written in terms of powers of two is used to reduce the complexity of the multiplication process. On the other hand Look up Tables is used for storing the pre calculated co products in memory based techniques. DA (Distributed Arithmetic) based on memory based techniques is trending architecture in recent years because of its high performance.

Distributed arithmetic method uses look up table for implementation of multiplication of fixed point digital filters [5]. The DA technique produces the output in lesser clock cycles and the elimination of multiplier reduces its hardware complexity [6]. The Expression of Distributed Arithmetic section is given as

$$Y = \sum_{k=1}^K A_k x_k \quad (1)$$

Where the fixed coefficients are represented as A_k [7], the input signal is represented as x_k and K is the number of input words. x_k can be modeled as

$$x_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \quad (2)$$

Where $x_k = \{b_{k0}, \dots, b_{k(N-1)}\}$, the output equations can be expressed as

Revised Manuscript Received on December 22, 2018.

P.Sritha, Department of Electrical and Electronics Engineering, Bannari Amman Institute of Technology, Sathyamanglam, India 638 401. (srithap@bitsathy.ac.in)

R.S.Valarmathi, Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamanglam, India 638 401. (atrmathy@gmail.com)

P.Ramya, Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamanglam, India 638 401. (ramyap@bitsathy.ac.in)

$$y = \sum_{n=1}^{n-1} \left[\sum_{k=1}^K A_k b_{kn} \right] 2^{-n} + \sum_{k=1}^K A_k (-b_{k0}) \quad (3)$$

The equation (3) is the finalized form of Distributed Arithmetic Technique. The value of b_{kn} is either 0 or 1 and has 2^k possible values. The pre-computed results are store in the ROM. The Size of the ROM is 2^*2^k , since it has to store both positive and negative values of b_{kn} .hence the size of the memory increases with the word size. The Diagrammatic representation of the Binary based Distributed Arithmetic is shown in the figure 1.X1, X2, X3, X4 are the input lines which are carried as single bit with Least Significant Bit of b_{kN-1} and the sign bit of b_{k0} as the Most Significant bit[7]. S is the sign bit timing signal. If S is set at position1 then S is set as 1 for sign bits and set as 0 for others. During the Clock cycle, the switch will be in position 2 [7].

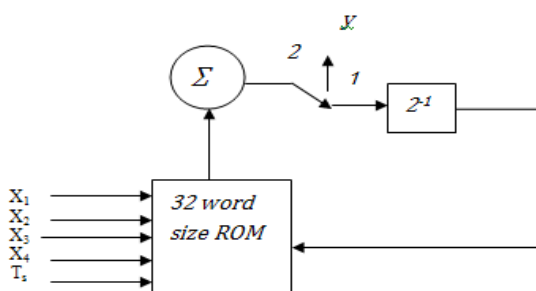


Figure1: Binary Based DA Technique

I.A. MODIFIED DA METHOD

The memory requirement is reduced into half by using adder and subtractor block The upper half of coefficients are the mirror image of the lower half of coefficients in LUT .If S=1, addition operation is performed and if S=0, subtraction operation is performed. The modified distributed arithmetic section is shown in the figure. [8]

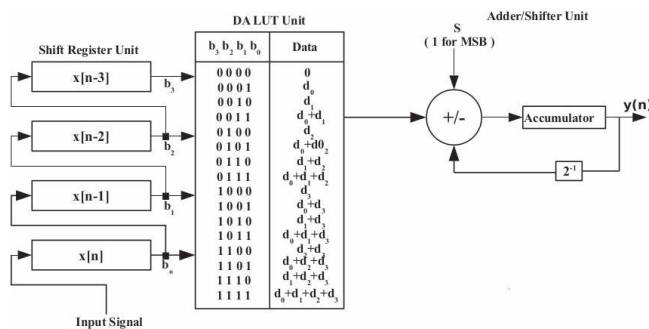


Figure 2 Modified DA technique

C. Cowan et al has developed the DA based technique by using new adaptive algorithm. The algorithm eliminates the need of multiplication by using LUTS, add and scaling operations. But this algorithm is not suitable for recursive type filters. [9].

II. OFFSET BINARY CODING METHOD

The Offset binary coding (OBC) is used to reduce the area of look up tables by the scale factor of 2 to 2^{N-1} . However the size of the look up table increases with the size of the FIR filter. [10]. the signed bits (-1, 1) are used as

input in place of binary bits. The structural design of Offset Binary Coding-DA FIR filter implementation is shown in Fig. 5 [10].

The final Expression for OBC DA technique is given by

$$y = \frac{1}{2} \sum_{k=1}^K d_k \left[\sum_{n=0}^{N-1} s_{kn} 2^{-n} - 2^{-(N-1)} \right] = \sum_{n=0}^{N-1} P(b_n) 2^{-n} + 2^{-(N-1)} P(0)$$

Rearranging the equation, we get

$$P(b_n) = \sum_{k=1}^K \frac{d_k}{2} s_{kn} P(0) = - \sum_{k=1}^K \frac{d_k}{2}$$

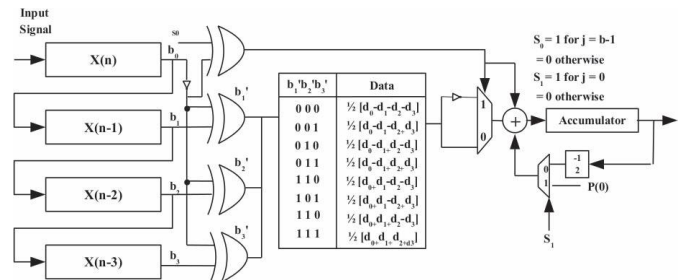


Figure3: offset Binary Coding DA Method

In the figure, the bit values in the left side b_{0n} are exored with the remaining input signals for the formation of address to the memory ROM[10]. The control signal for add and subtraction operation is given by the exored operation of b_{0n} and S_0 , so the OBC DA technique eventually reduces the memory size with additional requirement of EXOR Gate[10]

III. A.LUT LESS OBC DA METHOD

The LUT less OBC method has reduced the area requirement of DA technique. J. Choi etal has implemented the partial sum method as the DA reduction technique with a global shifter and adder implementation[10]. This method is needed area reduction because of the XOR and Multiplexer unit. The area overhead need is eliminated by Heejong Yoo et al. The proposed architecture is shown in the figure 4 [11].The single adder/shifter unit is used without the control signals from input or coefficients of the filter. The new DA based technique developed by Heejong Yoo et al for memory reduction for higher order filters. The developed algorithm uses recursive iteration methods for reduction of transistors, logic elements and hardware usage at the additional requirement of adders than the original technique [11]. The computational overhead of OBC can be reduced by the Sliding block OBC DA which implements the encoding technique in the memory. In Sliding Block DA technique, the input data of the filter is stored in a block are windowed at appropriate samples and convolved with the coefficients of FIR filter [12].The combination of Sliding Block with the offset binary has reduced the memory requirement of filter, but this technique does not parallel



updating the contents of coefficient filter operations leads to the more time execution. [12].The coefficient distributive method implements adder and multiplexer structures for changing the time varying coefficients has increased the area of the hardware. [13].

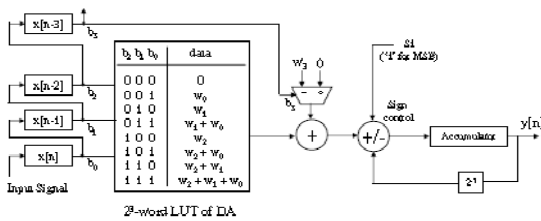


Figure 4: LUT less OBC Method

IV. DA TECHNIQUES IN ADAPTIVE FILTERS

Adaptive filters are widely used in the applications of Echo cancellation, communication, networking and signal processing [14].The main difficulty of implementing DA in adaptive filter design is the updating of coefficients at every clock cycles. D.J. Allred et al implemented a new DA based Least Mean Square technique in Fir filter. This technique has reduced power dissipation by decreasing the clock speed with increase in area by implementing the auxiliary LUT. This auxiliary LUT is mainly used to updating the contents of LUT used in adaptive FIR filters. The proposed architecture is composed of three modules namely DA filter module, auxiliary LUT and the controller module. [15].

Rui Guo et al have implemented two Distributed Arithmetic based techniques in which the Adaptive FIR filter is employed for the reduction of the hardware complexity. The first method uses the commutative property of convolution process for filtering operation and for address of the LUTS [16]. The second method uses object binary coded method for the decrease of LUT entries [16]. This two methods has eradicated the need of auxiliary LUT for updating the contents, hence the area requirement for

implementation of this algorithm is reduced than the previous methods. The process of weight adaptation is a complex process leads to the difficult mathematical procedures, because no partial products of the filter coefficients are stored. The pipeline Architecture method has replaced the adder/ shift accumulation block of DA algorithm by Carry save accumulation unit [17]. Two different Clock periods are used in this design. One clock period is used for carry save accumulation block and another clock period is utilized by remaining all other devices in the circuit. The architecture proposed by Park. et al has reduced both power and area requirement. The Look up table sharing technique for the computation of outputs and weight increment terms has reduced the usage of adders in the algorithm [18]. The updating of values in each iteration will lead to increase in time and power. The salient features of the Various DA based Adaptive filter algorithm is presented in the table 1. The area and power requirement of various algorithms is also analyzed. The algorithm proposed by Park et al has lowest power requirement, whereas the algorithm proposed by Mohanthy et al has lowest area requirement.

Harpreet Singh et al have implemented the DA technique in Adaptive filter. Memory elements are not used for updation of filter coefficients. The one scaling accumulator is used for storing and shifting the sum of all DA base units.[19]

Suganthi Venkatachalam et al have proposed the three approximate sum of products using Distributed Arithmetic technique [20]. The first model affords a notable power reduction and other two models provide better accuracy with increased area and power than the first model [20]. They implemented this DA based ASOP in Image signal processing applications, where the accuracy of the output plays a major role.

V. COMPARISON RESULT OF ADAPTIVE FILTER BASED DA TECHNIQUE

S.No	DA based techniques	Salient features	Challenges	Area in μm	Power in mw
1	D.J. Allred et al (2005) [15]	An Architecture was developed to reduce the memory requirements of DA implemented in higher order Filters. This leads to 50% reduction than the original LUT based DA in higher order filters. The presence of Auxiliary LUT reduces the Power Dissipation	The auxiliary LUT leads to increases in complexity in terms of area and time.	23854	24.39

2	Rui Guo et al [16] (2011)	Two coefficients based DA techniques were proposed for FIR filter. This method attains good results in speed and area	This method needs additional circuit and time for revising the data stored in the LUT.	19987	21.24
3	Park et al [17](2012)	The parallel look up table updation and weight updating operation has increased the throughput rate of the fir filter. operations	Two clock cycles implemented in this technique has needs more time for execution	182657	9.41
4	Mohanthly et al [18] (2013)	The Intra iteration sharing method reduces the word size of the LUT	This method reduced the hardware requirement on increasing Power requirement.	68687	17.4

Table: 1- Comparison of Adaptive Filter based DA technique

VI. CONCLUSION

This paper has presented a various Distributed Arithmetic based FIR filters architectures. The original DA techniques requires higher area requirement. The modified DA technique has reduced the area requirement further. The Offset binary coding DA technique has reduced the memory requirement of LUT with additional requirement of EXOR operation. The LUT less DA based technique is also effective in reduction of area requirement. The DA based Adaptive filter algorithms are also evaluated. The Area and power requirement of Various DA based algorithm is also presented.

VII. REFERENCES

1. M.S. Prakash, R.A.Shaik“High Performance Architecture for LMS Based Adaptive Filter Using Distributed Arithmetic”, IPCSIT vol. 24 IACSIT Press-2012, Singapore pp18-22
2. S.F. Hsiao, JH ZhangJia, M-C Chen “ Low cost FIR filter designs based on faithfully Rounded truncated multiple constant multiplications”, IEEE Trans. Circuits Syst.-II-2013 Expression Briefs, 60,Page no: 287–291
3. F Nekoei, Y.S Kavian. “Some schemes of realization digital FIR filters on FPGA for communication applications”. IEEE Crimean Conference. on Microwave and Telecommunication Technology. September- 2010, Page no. 616–619
4. R.Hartley, “Subexpression sharing in filters using canonic signed digit multipliers”, IEEE Transcations-1996.
5. Peled A, B. Liu, "A new hardware realization of digital filters", IEEE Transcations on. Acoustic. Speech Signal Processing, volume. ASSP-22 Page no :456-462, 1974.
6. White S. A., "Applications of distributed arithmetic to digital signal processing: A tutorial review," IEEE Transactions - ASSP Mag., volume. 6, page no: 4-19, Jul. 1989.
7. G. N. Jyothi and Sri Devi Sriadibhatla “Distributed Arithmetic Architectures for FIR Filters-A Comparative review” IEEE Wi -SPNET -2017 conference- page no: 2684- 2690
8. Ghamkhari S. F.,Ghaznavi-Ghouschi M. B., “Low-power low-area architecture design for distributed arithmetic (DA)

- unit” 20th Iranian Conference on. IEEE, May, 2012 page no : 15–17
9. C.F. N. Cowan, S.G. Smith, and J.H. Elliott, “A Digital Adaptive Filter Using a Memory Accumulator Architecture: Theory and Realization” IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING VOL. ASSP-31, NO. 3, JUNE 1983.Pp 541-549
10. B.Hong, Haibin Y, Xi.Wang, and Ying Xi, “Implementation of FIR filter on FPGA using DAOBC algorithm”, IEEE -2010.
11. J. Choi, S. Shin, and J. Chung, “Efficient ROM size reduction for distributed arithmetic” in Proceedings of the IEEE ISCAS, Geneva,May 2000, vol. 2, page no :61–64.
12. Yoo and David. V. Anderson, “Hardware-efficient distributed arithmetic architecture for high-order digital filters,” in Proceedings. IEEE International Conference - Acoustics, Speech, Signal Processing (ICASSP), March. 2005, vol. 5, Page no: v/125–v/128.
13. Huang .W and Anderson D. V., “Modified sliding-block distributed arithmetic with offset binary coding for adaptive filters,” Journal of Signal Process Systems., volume. 63, Page no:153–163, 2010.
14. S. Haykin, “Adaptive Filter Theory”. Upper Saddle River, New Jersey: Prentice-Hall-1996.
15. Allred, D. J., H. Yoo, Krishnan, Weing Huang, and Anderson D. V., “LMS adaptive filters using distributed arithmetic for high throughput,” IEEE Transcations on Circuits Systems-Jul. 2005. Volume 52, Page no: 1327–1337.
16. Rio Guo, and De Brunner, "Two high-performance adaptive filter implementation schemes using distributed arithmetic." IEEE Transcations on CAS-II volume. 58, Sep. 2011 Page no: 600-604.
17. Park S. Y, Meher P. K., “Low power high throughput and low area adaptive FIR filter based on distributed arithmetic,” IEEE Transcations. on Circuits and Systems-II: 2013 Express Briefs, volume. 60.
18. Mohanty, P. Meher: “A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm”, IEEE Transcations on. Signal Processing-2013, 61 Page no : 921–932.



19. H.Singh, G.Singh,T.Singh “*Performance analysis of DA based adaptive FIR filter using FPGA*” Annual IEEE India Conference (INDICON) Year: 2014 Pages: 1 – 4
20. Suganthi .V, Seok-Bum K “*Approximate Sum-of-Products Designs Based on Distributed Arithmetic*” IEEE Transactions on Very Large Scale Integration Systems Year: 2018, Volume: 26, Pages: 1604 – 1608.