RESEARCH ARTICLE OPEN ACCESS

# Implementation of Adaptive FIR Filter for the Wavelet Transforms Using Distributed Arithmetic Technique

G. Sathiyavani[1], (M.E AE), J.Amali[2], (M.E AE), S. Indumadhi[3], M.E
[1]Student/Dept of Applied Electronics
[2]Student/Dept of Applied Electronics
[3]Assistant Professor/ECE Dept
Jayam College of Engineering and Technology, Dharmapuri DT, Tamilnadu, India

***Abstract***
When computational resources are limited especially multipliers, Distributed Arithmetic (DA) is used in place of the typical multiplier based filtering structures. However, DA is not well suited for the adaptive applications. The bottleneck is updating the memory table. Several attempts have been done to accelerate updating the memory, but at the expense of additional memory usage and of convergence speed. To develop an adaptive DA filter with an uncompromised convergence rate, the memory table must be fully updated. An efficient method of fully updating a DA memory table is involved in this paper. The update method is based on exploiting the temporal locality of the stored data and subexpression sharing. This updation method reduces the computational workload and requires no additional memory resources. The parallelism inherent in DSP may be well exploited to implement the computation intensive discrete wavelet transform and making maximal utilization of the look-up table architecture by reformulating the wavelet computation in accordance with the parallel Distributed Arithmetic algorithm.

***Index Terms—*** Adaptive FIR Filter, Distributed Arithmetic (DA), Look up Table (LUT), LMS Algorithm, Offset Binary Coding (OBC).

## I. INTRODUCTION

Most of the signal processing algorithms involve DA, whereby computing the inner product of two vectors comprises most of the computational workload. As because of this computation, DA is widely used in signal processing algorithms. The inner product is evaluated using multipliers and adders. Instead of using multipliers DA can be used, so that the computational workload is reduced and attaining better performance.

This reduction is due to the storage of precomputed partial sum of the filter coefficients in the LUT. Due to the above case, DA involves with fewer arithmetic resources and no multipliers. Thus, this makes DA as one for computing with limited resources. Some of the applications of adaptive filtering are acoustic echo cancellation, signal de-noising and channel equalization. Due to the following issues, DA is not well suited for the above applications. The above said issue is, the computational workload for the adaptive filtering is more compared to the non adaptive filtering. The amount of resource required for adaptive filter is also increased. These additional resources are required for updating the memory content of DA.

In order to overcome the above issues, a new type of DA is involved here. That is, by updating the contents of the memory table without comprimising the convergence speed and requiring no additional memory resources and also by modifying the computational flow, the computational workload is said to be reduced. The goal of this paper is to update the memory table of the Distributed Arithmetic without comprimising the convergence rate and requiring no additional memory resources. With this updation, the Discrete Wavelet Transform is computed with the maximum utilization of look up table architecture.

## II. RELATED WORKS

In paper [2], the author proposed the DA formulation of Block LMS algorithm. This paper identifies that both convolution and correlation are performed using a common LUT for the computation of filter outputs and weight increment terms. With the previously derived DA, a parallel architecture for the implementation of BLMS Adaptive Digital Filter is made. This method is used to reduce the number of adders involved and also number of LUT words, while compared to the existing method by using one DA module, one error bit slice generator and one weight update cum bit slice generator. The number of flip flops is increased by this method and the area is also increased.

In paper [3], a common way is proposed for implementing the constant multiplication is by a series of add and shift operation. A previously derived common subexpression is briefly reviewed while the problem of minimizing the number of additions in filter designs by identification of suitable subexpressions is presented. This subexpression sharing for the single multiplier is limited, but for digital filters much more to be gained from this sharing. An algorithm is presented for converting two's complement to canonic signed digit. This sharing method reduces number of additions, but due to this algorithm the latch delay and the routing cost is increased.

## III. PROPOSED SCHEME

Updating the memory table of the DA with an uncomprimised convergence rate for the adaptive FIR Filter design. The update method is based on exploiting the temporal locality of the stored data and sub-expression sharing, Whereby the computational workload and additional memory resources required are reduced and increasing the performance of the adaptive FIR Filter. With this filter design the Discrete Wavelet Transform is computed with the maximum utilization of LUT.

### 3.1 Distributed Arithmetic

DA is used for calculating the MAC operations which is common in DSP algorithms like convolution and correlation. DA is a slow process as it is bit-serial in nature. It is said to be fast, if the vector elements are same as the wordsize. The process involved here is, precomputing the values and storing the result in the LUT with the input as address. By reducing the LUT size, the area is saved and also the system performance is said to be increased.

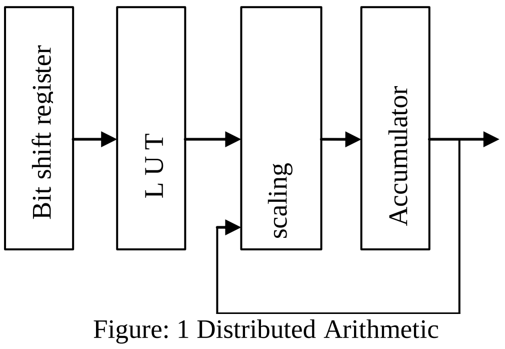

Figure: 1 Distributed Arithmetic

Two common optimizations are involved in reducing the LUT size i.e.) unreasonable amount of memory is reduced by this method. Thus, the two types are breaking up filter into smaller units and offset binary coding.

3.1.1 Breaking up the Filter

The memory requirement of the above said DA is increased by increasing the size of the filter. In order to say, a 64-tap DA FIR filter requires $2^{64}$ entries in the DA LUT. So, it is overcome by breaking breaking up the filter into smaller base DA filtering units that utilize tractable memory sizes and then summing up the outputs of these units.

Thus the diagram shows that output are summed and then it is given to the scaling process. Next to the scaling process, accumulation is carried out whereby the feedback is involved after this process.The output is feedbacked to the scaling process.

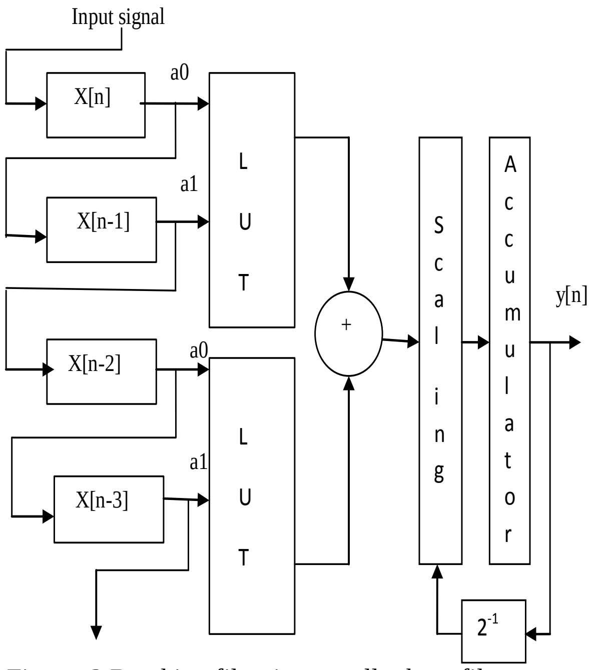

Figure: 2 Breaking filter into smaller base filter

3.1.2 Offset Binary Coding

Offset binary coding which can be used to halve the size of the memory tables in DA. Thus offset binary coding is used to reduce the memory size, that is the ROM size is reduced to half. From the diagram it shows the first half is the inverse symmetry of the next half.

Considering the output as the following,

$$y = \sum_{k=1}^{K} w_k x_k \quad (1)$$

By deriving the equation(1) and substituting the values we found that the 16 rows is reduced to 8 rows in the LUT by taking k=4.Rather than updating each entry of a memory table, some adaptive DA filters only update a few entries per sample period. The advantage of this approach is that it reduces the computational workload and this is given by the figure 3.

### 3.2. Updation of memory table

The implementation of DA is involved by storing all combinational sum of the filter coefficients in the memory table, whereas the proposed DA

| $b_{1n}$ | $b_{2n}$ | $b_{3n}$ | $b_{4n}$ | $c_{1n}$ | $c_{2n}$ | $c_{3n}$ | $c_{4n}$ | contents |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | -0.74 |
| 0 | 0 | 0 | 1 | -1 | -1 | -1 | 1 | 0.63 |
| 0 | 0 | 1 | 0 | -1 | -1 | 1 | -1 | 0.21 |
| 0 | 0 | 1 | 1 | -1 | -1 | 1 | 1 | 0.32 |
| 0 | 1 | 0 | 0 | -1 | 1 | -1 | -1 | 1.04 |
| 0 | 1 | 0 | 1 | -1 | 1 | -1 | 1 | -0.93 |
| 0 | 1 | 1 | 0 | -1 | 1 | 1 | -1 | 0.09 |
| 0 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | 0.02 |
| 1 | 0 | 0 | 0 | 1 | -1 | -1 | -1 | 0.02 |
| 1 | 0 | 0 | 1 | 1 | -1 | -1 | 1 | 0.09 |
| 1 | 0 | 1 | 0 | 1 | -1 | 1 | -1 | -0.93 |
| 1 | 0 | 1 | 1 | 1 | -1 | 1 | 1 | 1.04 |
| 1 | 1 | 0 | 0 | 1 | 1 | -1 | -1 | -0.32 |
| 1 | 1 | 0 | 1 | 1 | 1 | -1 | 1 | -0.21 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | -1 | 0.63 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.74 |

Figure:3 Offset Binary Coding

involves the above and as well as it is updated at the arrival of samples through updating method. In an adaptive Filter(LMS), the weights $w_i$ are updated according to the equation

$W_i[n+1]=w_i[n]+\mu e[n]x[n-i]$ (2)

Where µ is the step size and e is the error signal between the desired and filtered output. The memory table at a sample time instance n be denoted MEM[n].From the observation, the even addressed locations in the contents of MEM(n) are the lower half contents of MEM(n-1).Similarly, the odd addressed location of MEM(n) is given by the even addressed location of MEM(n) according to the equation(3)

$MEM_{(2l+1)}[n]=MEM_{(2l)}[n]+x[n],l=0,..,2^{k-1}-1$ (3)

The fig.4 shows that the contents in the lower half of MEM(n-1) is re-mapped to the even addressed locations of MEM(n).

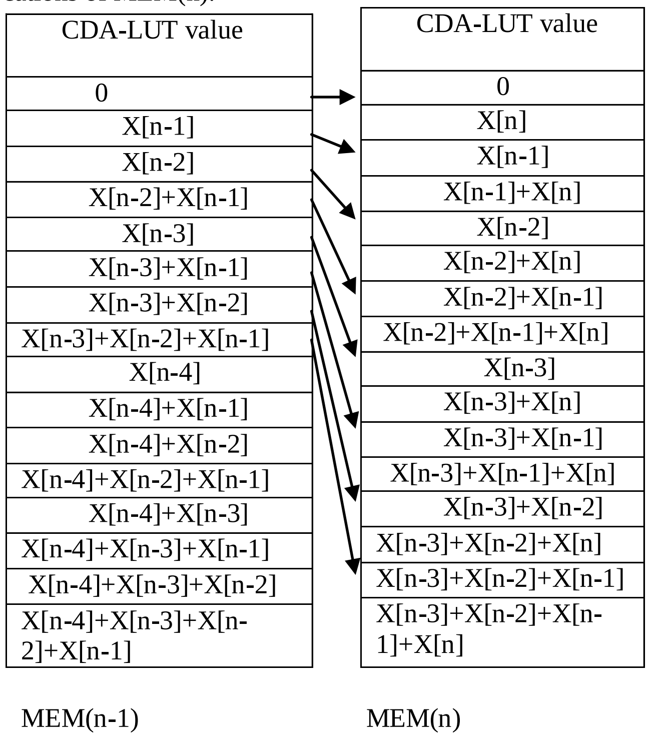

Figure: 4 Memory Update of DA

It is obtained by just left rotating the address lines of the memory table. This address rotation makes the LUT contents to be same eventhough the external logic sees the contents of the LUT is remapped. Thus the above diagram shows the mapping of address from MEM(n-1) to MEM(n). The contents of the odd addressed locations in the MEM(n) are overwritten by the values obtained according to the equation above. In other words, the contents of the odd addressed locations of the MEM[n] are obtained by reading the contents of the corresponding preceding even addressed locations, adding the newest sample x[n] and then storing the result back at the odd addressed locations.

### 3.3 Discrete Wavelet Transform

The discrete wavelet transform is said to be a fast computation of wavelet transform,which is easier to implement. The computational time and the resource required is reduced. In DWT, using digital filtering techniques a time-scale representation of the digital signal is obtained. The signal to be analysed are passed through the filter. Thus, the signal transmission and the reconstruction of the signal takes place with processing of the signal.Thus, the

Discrete Wavelet Transform is computed with the maximum utilization of Look Up Table Architecture.

## IV. SIMULATION RESULT

Thus, by updating the memory table of the Distributed Arithmetic we obtain high performance of Adaptive FIR Filter. By this method throughput is increased, power consumption is better compared to the existing method. Thus the simulation result shows that the Discrete Wavelet Transform is computed with the maximum Utilization of LUT.

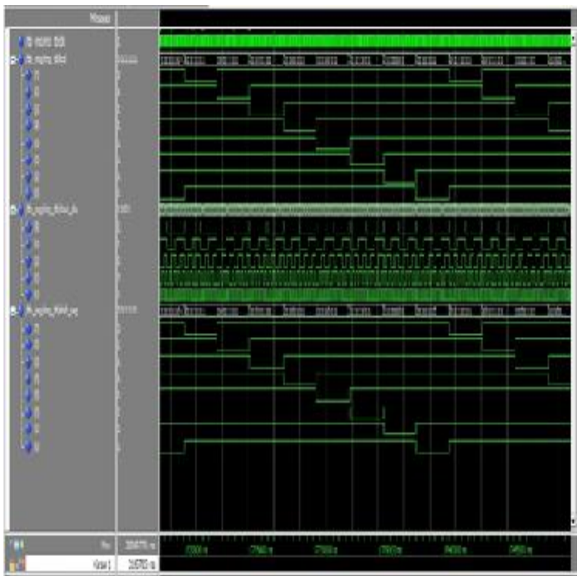

Figure: 5 LUT utilization of DWT

## V. CONCLUSION

This paper presented the updation of the memory table of Distributed Arithmetic using the temporal locality of the stored data and subexpression sharing. It is well implemented for Adaptive Filtering applications. Thus, the FIR Filter is updated without compromising the convergence rate and requiring no additional memory resources so that the computational workload is reduced. Here, the LUT size is reduced by splitting the filter and by using OBC.With this updation the DWT is computed with maximum utilization of LUT architecture. Thus, this paper reduces the computational workload of LUT access and increases the performance of Adaptive Filter.

## REFERENCE

[1] Sang Yoon Park And Pramod Kumar Meher "Low Power,High Throughput And Low Area Adaptive Fir Filter Based On Distributed Arithmetic" Ieee Trans.On Signal Processing,Vol. 60, No. 6, Jun, 2013.

[2] Basant K. Mohanty and Pramod Kumar Meher, "A High-Performance Energy-Efficient Architecture for FIR Adaptive Filter Based on New Distributed Arithmetic Formulation of Block LMS Algorithm,"IEEE Trans.on signal processing,vol. 61, no. 4, february15, 2013.

[3] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. vol. 43, no. 10, pp. 677–688, Oct. 1999.

[4] R. Guo and L. S. DeBrunner, "Two high-performance adaptive filter implementation schemes using distributed arithmetic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 9, pp. 600–604, Sep. 2011.

[5] Erhan Özalevli,Walter Huang,Paul E. Hasler and David V. Anderson, "A Reconfigurable Mixed-Signal VLSI Implementation of Distributed Arithmetic Used for Finite-Impulse Response Filtering,"IEEE Trans.on Circuits and Systems-I: Regular papers,vol. 55, no. 2, March 2008.

[6] Chip-Hong chang, Jiajiachen, and A.P.Vinod(2008)"Information Theoretic Approach to Complexity Reduction of FIR Filter Design," IEEE Trans.on Circuits and Systems-I: Regular papers,vol. 55, no. 8, september 2008.

[7] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 7, pp. 1327–1337, Jul. 2005.

[8] Emilio Soria, Javier Calpe, Jonathon Chambers, Marcelino Martínez, Gustavo Camps and José David Martín Guerrero, "A Novel Approach to Introducing Adaptive Filters Based on the LMS Algorithm and Its Variants," IEEE Trans.on education, vol. 47, no. 1, February 2004.

[9] Marcos Martinez-Peiro,Eduardo I.Boemo, and Lars Wanhammer(2002) "Design of High-Speed Multiplier less Filters Using a No recursive Signed Common Sub expression Algorithm," IEEE Trans.on Circuits and Systems-II: Analog and digital signal processing,vol. 49, no. 3, march 2002.