



# Design and Performance Verification of Reconfigurable FIR Filter Based on Split-Up Distributed Arithmetic

G.Pavithra<sup>1</sup>, R.Kousalya<sup>2</sup>

PG Scholar, Department of ECE, SVCE, Sriperumbudur, India<sup>1</sup>

Assistant Professor, Department of ECE, SVCE, Sriperumbudur, India<sup>2</sup>

**ABSTRACT:** This proposal presents a novel split-up architecture for low-power and low-area implementation of reconfigurable FIR filter based on distributed arithmetic (DA). The memory size of the proposed design is significantly reduced by split-up look up tables (LUT) and by reconfiguring the order of filter length. The conventional shift/add accumulation unit for DA-based partial product coefficient computation is replaced by a carry-save accumulation unit to reduce the area complexity. Reduction of power consumption is achieved in the proposed design by reconfiguring the higher order bits into lower order. It involves the reduced number of LUT's and registers compared to the existing DA-based design. From Cadence EDA tool synthesis results, it is found that the proposed design consumes 54% less power and 69% less area over the previous DA-based FIR filter in average for filter lengths  $N = 4$  and  $8$ . The simulation results thus obtained confirms the superiority of the proposed topology over the existing topologies.

**KEYWORDS:** FIR, Distributed Arithmetic (DA), Split-Up DA, Look Up Tables (LUT), Carry Save Accumulation (CSA), Shift Register.

## I. INTRODUCTION

Recent advances in multimedia and mobile computing applications demand low-power and high-performance VLSI Digital Signal Processing (DSP) systems. One of the most widely used filtering operations in DSP is Finite-Impulse Response (FIR) filtering. Multiplication is the strongest operation because it is repeated addition. It requires large portion of chip area to design. Consumption of power is more. Memory-oriented structures are more compact compared with the multiply and accumulate based structures; and have many other added advantages, e.g., greater potential for decreased-latency and higher throughput implementation and are expected to consume less power due to less switching activities for memory-read operations compared to the conventional multipliers. Memory oriented structures are very well-suited for many digital signal processing (DSP) applications, which involve multiplication with a fixed set of coefficients. For this Distributed Arithmetic kind of architecture used in FIR filter. In recent years, the multiplier-less distributed arithmetic (DA) algorithm-based technique has acquired substantial popularity for its higher regularity and high-throughput processing capability, which results in cost-effective and area-time efficient computing structure. Distributed arithmetic, is one of the best ways to implement convolution operation with multiplier less structure, where the MAC operations are performed by a series of LUT access and summations using adder. Distributed Arithmetic is a different kind of approach for implementing digital filters. The key idea is to replace all additions and multiplications by a table (LUT) and a shift- add accumulator unit. Basically, each look up table (LUT) is a group of single bit memory cells storing individual bit values in each of the memory cells. Distributed Arithmetic provides efficient area-time and cost-effective computing structures.

The DA implementation of an FIR filter realization is very particularly attractive for lower filter order cases due to LUT address space limitations. The collection of outputs of a low-order filters can be added together to define the output of a high-order FIR. This brief proposes a DA-based architecture for low-power and low-area, implementation of FIR filter. The contribution of this brief are as follows.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Special Issue 3, April 2017

- 1) Memory consumption is reduced by memory split-up.
- 2) Conventional shift-add based accumulation is replaced by a carry-save accumulation of signed partial inner product computation to reduce the sampling period and helps to reduce the area complexity of the proposed design.[2]
- 3) Reduction of power consumption is achieved by reconfiguring the higher order bits into lower order bits.

In the next section, we present a brief review of the Distributed Arithmetic (DA) algorithm, followed by the description of the proposed DA-based technique for FIR filter in Section III. Simulation and synthesis results are discussed in Section IV. Conclusions and future work are given in Section V.

## II. DISTRIBUTED ARITHMETIC ALGORITHM

Distributed Arithmetic (DA) technique is basically a bit-serial in nature. It is a kind of bit-level rearrangement of the multiply and accumulation (MAC) operation. The DA is a computational algorithm that provides an effective implementation of the dot product or sum of product (SOP). DA is a bit-serial operation used to compute the inner (dot) product of a variable input vector and a constant coefficient vector in a single direct step and is given by,[2]

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

Where,

y- Output response

$A_k$  -Constant filter coefficients

$x_k$  - Input data

Let  $x_k$  be N-bits scaled two's complement number and it can be represented as,

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

Where  $b_{k0}$  is the sign bit and by substituting (2) in (1), get the expression as

$$y = -\sum_{k=1}^K (b_{k0} \bullet A_k) + \sum_{k=1}^K \sum_{n=1}^{N-1} (A_k \bullet b_{kn}) 2^{-n} \quad (3)$$

Rearranging the summation based on power terms and then grouping the sum of the products, the final reformulation is given by

$$y = -\sum_{k=1}^K A_k \bullet (b_{k0}) + \sum_{n=1}^{N-1} \left[ \sum_{k=1}^K A_k \bullet b_{kn} \right] 2^{-n} \quad (4)$$

The DA of FIR filter consists of Look Up Table (LUT), Shift registers and scaling accumulator is shown in the below figure 1. In DA, all the cumulative partial product outcomes are pre-computed and stored in a Look up Table (LUT) which is addressed by the multiplier bits. A filter with K coefficients the LUT has  $2^K$  values. In the above equation, each term inside the bracket indicates a binary AND operation involving all the bits of the constant and a bit of the input variable. The exponential factor denotes the scaled parts of the bracketed pairs to the total sum. We can store these in a look-up table of  $2^K$  words addressed by K bits. For e.g., if the number of inputs is 4, then the LUT will have 16 memory words.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Special Issue 3, April 2017

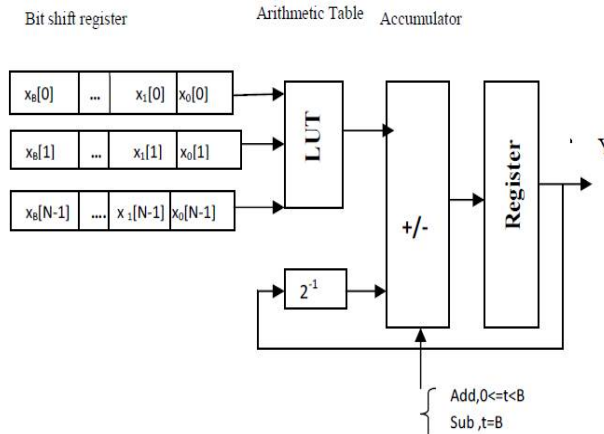


Figure 1. Block diagram of Distributed Arithmetic

The complete dot product computation takes  $X$  clocks where  $X$  is the number of input variable bits, and is independent of the number of input variables. During the first iteration, the LSB bits  $x_0[0], x_0[1], x_0[N - 1], \dots$ , of the  $K$  input samples forms an address of  $K$ -bit to the look up table (LUT) for  $fn(x, 0)$ , and that LUT's output becomes the initial value of the accumulator. During the second iteration, the next-to-LSB bits  $x_1[0], x_1[1], x_1[N - 1], \dots$ , of the  $K$  input samples forms another address of  $K$ -bit to the lookup table for  $fn(x, 1)$ , and the adder sums the LUT output to the contents of the accumulator shifted by one bit position. This process continues until the final iteration, where the MSB bits  $x_B[0], x_B[1], x_B[N - 1], \dots$ , of the  $K$  input samples forms an address of  $K$ -bit to the LUT for  $fn(x, B)$  and the adder sums the LUT output to the contents of the accumulator after shifting it to the corresponding position.

## A. DA Technique for 4-tap FIR Filter

Consider the design of 4 tap FIR filter, where

No of coefficients=4

No of inputs =4

LUT size = $2^4$ =16 memory location

In this method, possible outputs of partial product co-efficient values are pre-computed and stored in the LUT. LUT addressed through input of the filter. For 4 tap FIR filter, 4 tap indicates the number of inputs to the filter and address bit, as well as it indicates the number of co-efficient of the given filter. Each location has different output for the corresponding input values. The possible combination of inputs for this filter is 0(0000) - 15(1111) and the contents of LUT is shown in the below table 1.

TABLE 1

LUT content for 4 tap FIR filter

No	Address	Data
0	0000	0
1	0001	h0
2	0010	h1
3	0011	h0 + h1
4	0100	h2
5	0101	h2 + h0
6	0110	h2 + h1

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Special Issue 3, April 2017

7	0111	$h_2 + h_1 + h_0$
8	1000	$h_3$
9	1001	$h_3 + h_0$
10	1010	$h_3 + h_1$
11	1011	$h_3 + h_0 + h_1$
12	1100	$h_3 + h_2$
13	1101	$h_3 + h_2 + h_0$
14	1110	$h_3 + h_2 + h_1$
15	1111	$h_0 + h_1 + h_2 + h_3$

## B. Disadvantage

A filter with K coefficients the LUT has  $2^K$  values. For higher order filter LUT size will increase, it requires more memory space. So, for higher order filter applications this DA will not suit, to overcome those drawbacks this paper proposed an algorithm called split-up DA.

## III. PROPOSED SPLIT-UP DA ALGORITHM

The above mentioned technique holds good only when we go for lower order filters. For higher order filters, the size of the LUT increases exponentially with the filter order. For a filter with K coefficients, the LUT have  $2^K$  values. This in turn reduces the performance of the filter design. Therefore for higher order filters, the size of the LUT to be reduced to reasonable levels. To reduce the size, the LUT can be splitted into several LUT's, called Split-Up LUT's. Each Split-up LUT operates on a different set of filter taps. The results obtained from the split-up LUT's are summed and it is shown in the below figure 2.

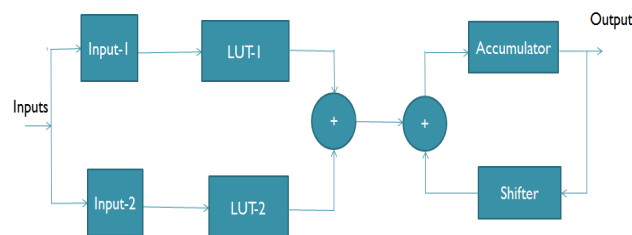


Figure 2. Block diagram of Split-Up DA algorithm

### A.4-Tap FIR Filter with Split-Up DA Method

LUT is decomposed into LUT 1&LUT 2. Each LUT has 2 inputs and 4 memory locations. It is shown in figure 3. Consider, Input = 1011 means ,first 2 bits are address bit of LUT 1, output becomes 10=  $h_0$ , Remaining 2 bits are address bit of LUT 2, output becomes 11 =  $h_2 + h_3$ . Final Output = output of LUT1 + output of LUT2 =  $h_0 + h_2 + h_3$ .

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Special Issue 3, April 2017

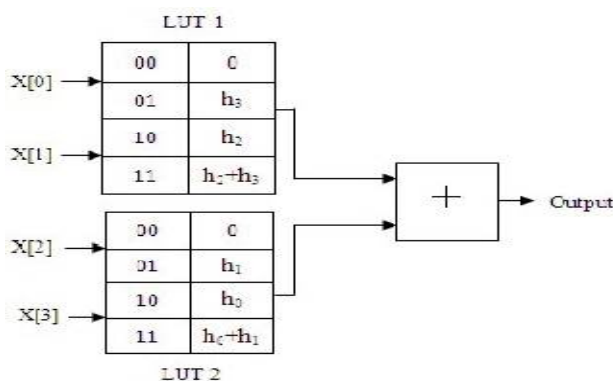


Figure 3. 4-tap FIR filter with Split-up DA method

In this method, the memory has been reduced to 8 locations only. Previous method 16 memory locations required for producing the same output. Thus the re-configurability of the filter order takes place by splitting the higher order filter lengths into lower order filter lengths thereby performing filtering operation parallelly and summing up all the lower order filters output and finally the filter output is carried out with lower order whereas actually the inputs are in higher order using carry-save accumulation unit (CSA) is shown in the below figure 4.

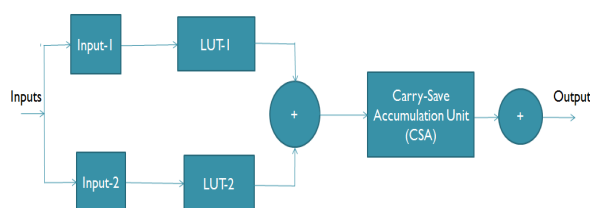


Figure 4. Block Diagram of Proposed System

## B. Carry save Accumulation Unit (CSA)

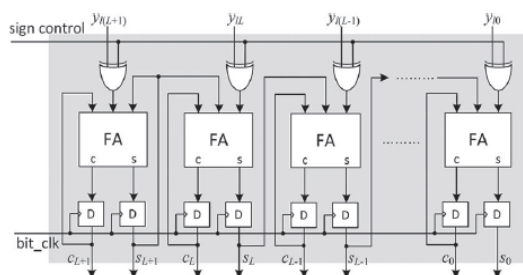


Figure 5. Carry-save implementation of shift accumulation

Since the shift accumulation in Figure 1 & 2 involves worst critical path, hence perform the shift accumulation using carry-save accumulator, as shown in Figure 5, [10]. The bit slices of vector are given one after the next in the least Significant bit (LSB) fashion to the most significant bit (MSB) order to the carry-save accumulator unit. However, the negative (2's complement) of the LUT output needs to be accumulated in case of MSB address slices. Therefore, all the bits of LUT output are fed through XOR gates with a sign-control input which is set to 1 only when the MSB slice appears as address. The XOR gate output thus produces the 1's complement of the LUT output corresponding to the MSB address slice, but do not affect the output for other bit slices. Finally, the sum and carry words obtained after clock cycles are required to be added by a final adder given in the figure 4 after the CSA unit and the input carry of the final adder is required to be set to 1 to account for the 2's complement operation of the LUT output corresponding to the MSB slice.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Special Issue 3, April 2017

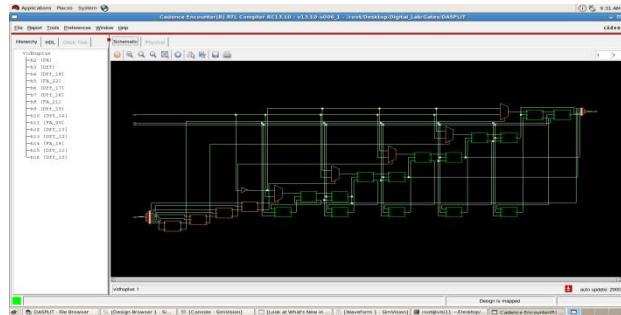


Figure 9. RTL of 8-tap FIR Filter Using Split-Up DA

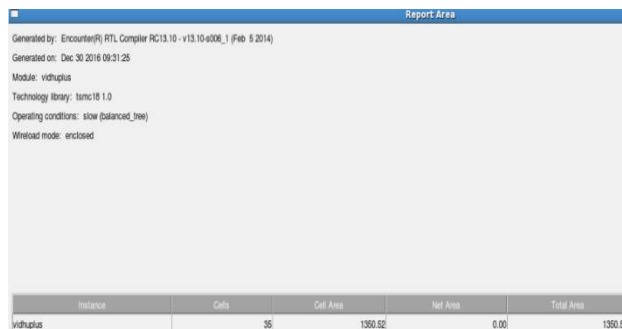
### C. Performance Evaluation Using Cadence EDA Tool

Cadence EDA is a tool that provides an accurate measurement of area, power and delay parameters over the Xilinx ISE software tool. The observed synthesized results for both existing and proposed FIR filter realization using DA is shown below.



Instance	Cells	Cell Area	Net Area	Total Area
pitamuganman	219	4307.69	0.00	4307.69

Figure 10. Area Utilization of 8-tap FIR Filter Using DA



Instance	Cells	Cell Area	Net Area	Total Area
vidhuplus	35	1350.52	0.00	1350.52

Figure 11. Area Utilization of 8-tap FIR Filter Using Split-Up DA

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Special Issue 3, April 2017



Figure 12. Power Consumption of 8-tap FIR Filter Using DA

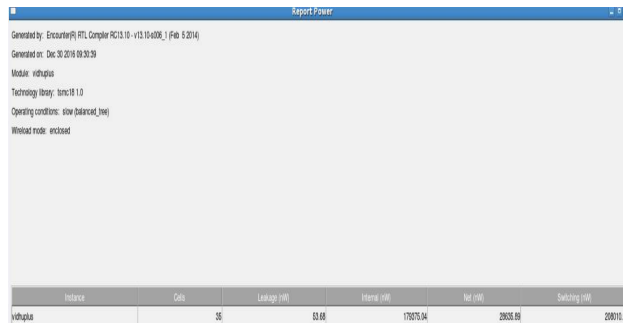


Figure 13. Power Consumption of 8-tap FIR Filter Using Split-up DA

## D. Comparison

TABLE 2

Comparison between DA and Split-up DA Methods

Algorithm	Area [ $\mu\text{m}^2$ ]	Power [nW]
Existing DA Algorithm	4307.69	61745.37
Proposed Split-up DA	1350.52	28635.89

From the above comparison Table 2, it clearly shows that the area and power consumption for the proposed method has been reduced to more than 50% than the existing FIR filter realization using DA. By using Cadence EDA tool, the proposed design has been synthesized accurately and the exact values of area and power has been generated and plotted.



# International Journal of Innovative Research in Computer and Communication Engineering

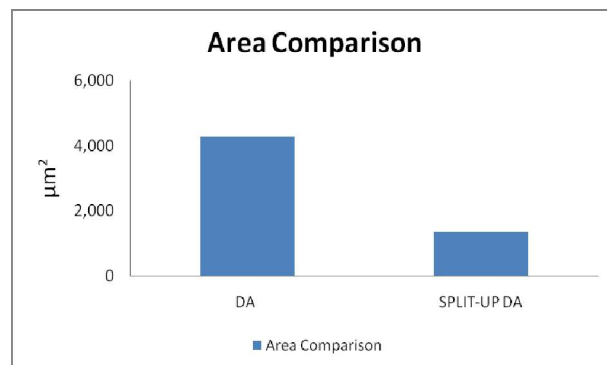
(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

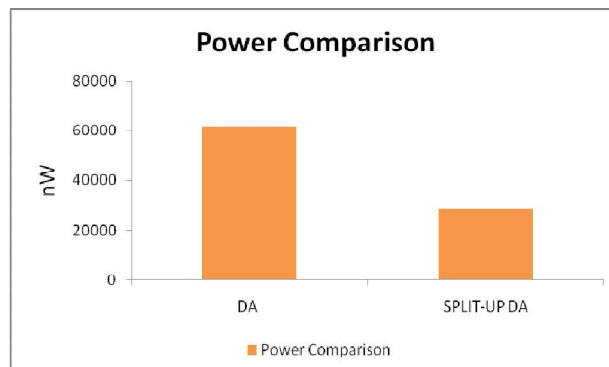
Vol. 5, Special Issue 3, April 2017

## E. Graphical Representation

Area: DA Vs Split-Up DA



Power: DA Vs Split-Up DA



## V. CONCLUSION AND FUTURE WORK

The proposed work suggested an efficient split-up DA architecture for low-power and low-area implementation of DA-based FIR filter. Memory size is reduced by splitting the LUT's and by reconfiguring the filter order. Also proposed a carry-save accumulation scheme of partial inner product coefficient computation to reduce area complexity. From the Cadence synthesis results, the proposed design consumes 54% less power and 69% less area over our previous DA-based FIR filter in average for filter lengths  $N = 4$  and 8.

The future work focuses on implementing the pipelined architecture for improving the speed, since in this proposed design only one sequence of bits can be processed at a time, whereas in the pipelining system more than one sequence of bits can be processed at the cost of improved throughput rate. And the same is to be synthesized using Cadence EDA tool.

## REFERENCES

- [1] Basant Kumar Mohanty, Pramod Kumar Meher and Subodh Kumar Singhal (Jan 2016) "A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic," Integration, the VLSI journal, (ELSEVIER).
- [2] Pranav J. Mankara, Ajinkya M. Pundb, Kunal P. Ambhorec, Shubham C and Anjankard (2016) "Design and Verification of low power DA-Adaptive digital FIR filter," 7th International Conference on Communication, Computing and Virtualization. (ELSEVIER)
- [3] Indranil Hatai, Indrajit Chakrabarti, Member, IEEE, and Swapna Banerjee, Senior Member, IEEE (April 2015) "An Efficient Constant Multiplier Architecture Based on Vertical-Horizontal Binary Common Sub-Expression Elimination Algorithm for Reconfigurable FIR Filter



ISSN(Online): 2320-9801  
ISSN (Print): 2320-9798

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Special Issue 3, April 2017

- Synthesis, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 62, NO. 4.
- [4] Basant Kumar Mohanty and Pramod Kumar Meher, Senior Member, IEEE (Feb 2015) "A high-performance FIR Filter Architecture for Fixed and Reconfigurable Applications," IEEE TRANSACTIONS ON VLSI SYSTEMS.
  - [5] S. Padmapriya and V. Lakshmi Prabha (2015) "Design of an efficient dual mode reconfigurable FIR filter architecture in speech signal processing," SCIENCE DIRECT, (ELSEVIER).
  - [6] J. Chen, C. Chang, F. Feng, W. Ding, J. Ding (Jan 2015) "Novel design algorithm for low complexity Programmable FIR filters based on extended double base number system," IEEE Trans. Circuits Syst I , Regul.Pap.62.
  - [7] S.Y. Park and P.K. Meher (Jul. 2014) "Efficient FPGA and ASIC realizations of DA-based reconfigurable FIR digital filter," IEEE Trans. Circuits Syst II, Express. Brief, 61,511–515.
  - [8] S. J. Darak, S. K. P. Gopi, V. A. Prasad, and E. Lai (May 2014) "Low-complexity reconfigurable fast filter bank for multi-standard wireless receivers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 5, pp.1202–1206.
  - [9] I. Hatai, I. Chakrabarti, and S. Banerjee (May 2014) "An efficient VLSI architecture of a reconfigurable pulse-shaping FIR interpolation filter for multi-standard DUC," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.
  - [10] S.Y. Park and P.K. Meher (Jun 2013) "Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic," IEEE Trans. Circuits Syst. II, Express Br. 60346–350.
  - [11] J. Chen and C. Chang (Dec 2009) "High-level synthesis algorithm for the design of reconfigurable constant multiplier," IEEE Trans Computer Aided des Integrated Circuits Syst 1844–1856.