

Area and Speed Efficient Implementation of Symmetric FIR Digital Filter through Reduced **Parallel LUT Decomposed DA Approach**

S. C. Prasanna¹, S. P. Joy Vasantha Rani²

¹Department of EIE, Valliammai Engineering College, Anna University, Chennai, India ²Department of Electronics, MIT Campus, Anna University, Chennai, India Email: mail2scprasanna@yahoo.co.in, joy mit@annauniv.edu

Received 25 March 2016; accepted 22 April 2016; published 9 June 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY). http://creativecommons.org/licenses/by/4.0/ **(c)**

Open Access

Abstract

This brief proposes an area and speed efficient implementation of symmetric finite impulse response (FIR) digital filter using reduced parallel look-up table (LUT) distributed arithmetic (DA) based approach. The complexity lying in the realization of FIR filter is dominated by the multiplier structure. This complexity grows further with filter order, which results in increased area, power, and reduced speed of operation. The speed of operation is improved over multiply-accumulate approach using multiplier less conventional DA based design and decomposed DA based design. Both the structure requires B clock cycles to get the filter output for the input width of B, which limits the speed of DA structure. This limitation is addressed using parallel LUTs, called high speed DA FIR, at the expense of additional hardware cost. With large number of taps, the number of LUTs and its size also becomes large. In the proposed method, by exploiting coefficient symmetry property, the number of LUTs in the decomposed DA form is reduced by a factor of about 2. This proposed approach is applied in high speed DA based FIR design, to obtain area and speed efficient structure. The proposed design offers around 40% less area and 53.98% less slice-delay product (SDP) than the high throughput DA based structure when it's implemented over Xilinx Virtex-5 FPGA device-XC5VSX95T-1FF1136 for 16-tap symmetric FIR filter. The proposed design on the same FPGA device, supports up to 607 MHz input sampling frequency, and offers 60.5% more speed and 67.71% less SDP than the systolic DA based design.

Keywords

Distributed Arithmetic, Field Programmable Gate Array (FPGA), Finite-Impulse Response (FIR) Filter, High Speed, Reduced Look-Up Table (LUT)

How to cite this paper: Prasanna, S.C. and Rani, S.P.J.V. (2016) Area and Speed Efficient Implementation of Symmetric FIR Digital Filter through Reduced Parallel LUT Decomposed DA Approach. Circuits and Systems, 7, 1379-1391. http://dx.doi.org/10.4236/cs.2016.78121

1. Introduction

These Finite impulse response (FIR) digital filters are extensively used in many digital signal processing (DSP) applications and communication systems [1] [2]. Due to the advancement in very large scale integration (VLSI) technology, DSP has become increasingly popular over the years, and demands the realization of FIR filters with high speed, less area and less power consumption.

The general form of FIR filter is represented by the equation,

$$y(n) = \sum_{k=0}^{N-1} h(k) x(n-k).$$
 (1)

where y(n) is the output; x(n-k) is the delayed input; h(k) is the coefficient; and N is the number of taps of the filter. This representation shows that one of the major issue or complexity lying in the realization of FIR filter is dominated by the complexity in the implementation of multipliers. In performing multiplication operation, the number of partial products generated increases with the increase in width of filter input and filter coefficient. This in turn increases the number of adder units and logic levels needed and hence logic depth of the structure, which consequently decreases the speed of operation of filter structure [3]. Since the complexity of implementation grows further with the filter order, which maximizes area and power consumption, real-time realization of these filters with desired level of accuracy is a challenging task. Such compute-intensive applications can be implemented efficiently over field-programmable gate arrays (FPGA) platform than application specific integrated circuits (ASICs) [4] [5] platform due to its speed, flexibility, and price performance over ASIC. Thus several researchers have contributed towards designing a low-power, low-area, and high speed dedicated and reconfigurable architectures for realization of FIR filters in FPGA platforms.

Several multiplier less approaches are proposed for implementing cost, area and time efficient computing structures for realizing FIR filters. Multiplier less DA based technique [6] stores the precomputed partial results of inner product, which are read and shift – accumulated to get the filter output. It yields faster output compared with the multiplier-accumulator-based designs The high-throughput processing capability, and increased regularity make this a popular approach for FIR filter implementation. DA was first introduced by Croisier *et al.* [7] and was further developed by *Peled and Liu* [8] for efficient implementation of digital filters. DA based design suggested for adaptive filter presented in [9] [10] cannot support high sampling frequency, as it requires several clock cycles for processing each input signal.

The DA based design for adaptive filter suggested in [11] offers high throughput at the expense of hardware cost. The memory requirement for DA-based implementation of FIR filters, however, exponentially increases with the filter order. To eliminate the problem of such a large memory requirement, Meher *et al.* [12] suggested systolic decomposition techniques for DA-based implementation, which was found to involve less area-delay complexity. *Park and Meher* [13] present high speed implementation of DA based reconfigurable FIR filter, which involves flexible frequency of operation, however, lesser the frequency, area utilized is less, and higher the frequency, area utilized is more. The structure in [13] employs parallel LUTs to speed up the computation similar to the proposed structure. Area optimization is done in the proposed design when compared to [13], by using the proposed reduced LUT decomposed DA algorithm for symmetric FIR filter.

This paper proposes reduced LUT decomposed DA approach to reduce the area in high speed implementation of DA based filter using parallel LUTs, to achieve area as well as speed optimization in symmetric FIR filter realization.

The rest of the paper is organized as follows. Section 2 presents the formulation of algorithm for conventional DA based scheme, and decomposed DA based scheme. The derivation of algorithm for the proposed structure for symmetric FIR filter is described in Section 3. The architectural details of conventional and proposed scheme are described in Section 4. In Section 5, implementation results and discussion on the comparison of proposed design with the earlier reported result are presented. Finally the proposed work is concluded in Section 6.

2. Formulation of Algorithm for Conventional and Decomposed DA Based FIR Filter

This section briefly outlines the formulation of algorithm for conventional DA based realization, and for the decomposed DA based realization of FIR filters [14].

2.1. Conventional DA Algorithm for FIR Filter Realization

The general form of representation of FIR filter given in (1) shows that the output of an FIR is the sum of prod-

uct of coefficient (impulse response) vector h(k) and the input vector x(k) To simplify the derivation, the N-tap FIR filter represented by (1), is written again in its compact form without time index n as,

$$y = \sum_{k=0}^{N-1} h(k) x(k).$$
 (2)

where the coefficients h(k) { $h(0), h(1), h(2), \dots, h(N-1)$ } are constants, and the input vector x(k)

 $\{x(0), x(1), x(2), \dots, x(N-1)\}\$ is a variable. Assuming B to be the word length of x(k), and also assume that the signal samples x(k) are unsigned, then x(k) can be represented as,

$$x(k) = \sum_{b=0}^{B-1} x_b(k) \cdot 2^b, \text{ with } x_b(k) \in [0,1].$$
(3)

where $x_b(k)$ denotes the b_{th} bit of x(k). By applying the expression in (3) into the expression in (2) the expanded form of inner product is represented as,

$$y = \sum_{k=0}^{N-1} h(k) \cdot \sum_{b=0}^{B-1} x_b(k) \cdot 2^b.$$
(4)

To get the distributed structure the order of summation over the indexes k and b are interchanged, and this results in

$$y = \sum_{b=0}^{B-1} 2^{b} \cdot \sum_{k=0}^{N-1} h(k) x_{b}(k).$$
(5)

Expressing it in simpler form

$$y = \sum_{b=0}^{B-1} 2^{b} \cdot F_{b}(k).$$
 (6a)

where

$$F_{b}(k) = \sum_{k=0}^{N-1} h(k) x_{b}(k).$$
(6b)

This shows that the filter output is the shifted accumulation of $F_b(k)$ for B bits. Ultimately the implementation of function $F_b(k)$ requires special attention. Here h(k) is a constant vector and x(k) is a variable of length B, which can take either 0 or 1 for all the N samples. Since h(k) is constant, all the possible 2^N values of product $h(k)x_b(k)$ is precomputed and stored in LUT. Now the input vector,

 $x_b(k) = \{x_b(0), x_b(1), \dots, x_b(N-1)\}$ forming the address lines for accessing the LUT to get the desired inner product $F_b(k)$. Thus inner product computation is performed using multiplier less DA based LUT. Finally shifted accumulation of B number of $F_b(k)$ provides the filter output.

Therefore the conventional DA algorithm represented by (5) or (6) shows that, the inner product is computed using (6b), which requires LUT of size 2^{N} words, and B cycles of memory (LUT) read operation for an input word length of B bits, followed by B number of shift—accumulation to get the filter output (6b). The structure used for implementing this conventional DA based FIR is shown in Figure 1.

2.2. Decomposed DA Algorithm for FIR Filter Realization

In conventional DA based FIR implementation, the size of LUT grows exponentially with number of coefficients (taps) N. For large values of *N*, however, the LUT size becomes too large, and the LUT access time also becomes large. The conventional DA-based implementation is, therefore, not suitable for large filter orders. This complexity can be resolved by decomposing single LUT into multiple LUTs, at the expense of additional adders as explained below.

When N is a composite number given by N = LM (L and M may be any two positive integers), then expression

in (2) becomes,

$$y = \sum_{k=0}^{LM-1} h(k) x(k).$$
 (7)

Now mapping the index k into (m + lM) for $m = 0, 1, \dots, M - 1$ and $l = 0, 1, \dots, L - 1$, the sum can be partitioned into L independent M^{th} parallel DA LUTs resulting in

$$y = \sum_{l=0}^{L-1M-1} h(m+lM) x(m+lM).$$
(8)

Using the representation of x(k) given in (3), into (8), and re-distributing the summation we get,

$$y = \sum_{b=0}^{B-1} 2^{b} \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} h(m+lM) x_{b}(m+lM).$$
(9)

Expressing it in simpler form

$$y = \sum_{b=0}^{B-1} 2^{b} \sum_{l=0}^{L-1} DF_{b}(m).$$
(10a)

where

$$DF_{b}(m) = \sum_{m=0}^{M-1} h(m+lM) x_{b}(m+lM).$$
(10b)

 $DF_b(m)$ is the inner product of decomposed form of DA FIR. These inner products can be computed using LUTs of size 2^M words rather than 2^N words in conventional DA approach. According to (10), in the decomposed form of DA FIR, L number of LUTs of size 2^M words are accessed in parallel, then these L outputs are added (the 2nd summation) to get the inner product, finally this sum is shift-accumulated (the 1st summation). This process is repeated for B cycles to get the filter output. Hence the size of the LUT can be greatly reduced using decomposed form of DA FIR, at the expense of additional adders. This structure requires B clock cycles (for the input word width of B) to get the filter output, as it has to fetch the LUT sequentially for B bit positions. In the proposed structure in order to speed up the computation process, LUTs corresponding to each L, is duplicated B times, so that the read operation from LUT, corresponding to each bit position is made in parallel, hence speeds up the computation, at the expense of additional (B-1)L LUTs. The number of LUTs is reduced by a factor of 2 by employing the proposed algorithm as explained in the next section.

3. Derivation of Algorithm for the Proposed Structure

This section describes the derivation of algorithm for implementing the proposed structure, which reduces the number of LUTs in the decomposed DA based symmetric FIR filter. Then this algorithm is explained with an example. The result of application of this algorithm to high speed DA FIR realized using parallel LUTs in decomposed form of DA FIR is discussed.

3.1. Derivation of Proposed Algorithm for Symmetric FIR Filter Realization

As explained in Section 2 the number of LUTs needed for realizing FIR filter using decomposed DA algorithm is L. However when the value of N is very large that would result in the use of large number of LUTs, that is larger L. This complexity for symmetric FIR filter is reduced in the proposed structure. In the proposed structure, the coefficient symmetry property of FIR filter, $\{h(n) = h(N-1-n)\}$ is exploited to reduce the number of LUTs needed for storing the inner products, by a factor of about 2, that is L/2 for L even and $\lfloor L/2 \rfloor + 1$ for L odd. This is possible by computing the inner product outputs for lower half of LUTs $\{\lfloor L/2 \rfloor + 1$ to L} from the corresponding and respective equivalent upper half LUTs itself by generating appropriate address signal as is discussed below.

To derive this algorithm let us first express the filter output in (9) as a function of inner product, IP(l,m) as,

$$y = \sum_{b=0}^{B-1} 2^{b} \cdot IP(l,m).$$
(11a)

where

$$IP(l,m) = \sum_{l=0}^{L-1M-1} h(m+lM) x_b(m+lM).$$
(11b)

Now splitting the first summation in the inner product function in (11b) as first half and as second half with reference to summation index l, with the assumption that L is even. Then

$$IP_{FH}(l,m) = \sum_{l=0}^{\left(\frac{L}{2}\right)^{-1}} \sum_{m=0}^{M-1} h(m+lM) x_b(m+lM), \qquad (12a)$$

is the expression for computing the inner product corresponding to first half of LUTs $(IP_{FH}(l,m))$, and

$$IP_{SH}(l,m) = \sum_{l=L/2}^{L-1} \sum_{m=0}^{M-1} h(m+lM) x_b(m+lM),$$
(12b)

is the expression for computing the inner product corresponding to second half of LUTs $(IP_{SH}(l,m))$. Now applying coefficient symmetry property, h(m+lM) = h(N-1-(m+lM)) for realizing second half of LUTs, in (12b), we get

$$IP_{SH}(l,m) = \sum_{l=L/2}^{L-1} \sum_{m=0}^{M-1} h(N-1-(m+lM)) x_b(m+lM).$$
(13)

When we compare the pre-computation values to be stored in the LUTs, computed using (12a) for the first half LUTs and (13) for the second half of the LUTs, coefficient values considered for the respective equivalent LUTs (L = 0 and L - 1, L = 1 and L - 2, etc.) are the same, but it is in the reversed order for second half when compared to the first half. Therefore the required inner product corresponding to the second half of LUTs is obtained using first half of LUTs itself, by reversing the order of address bits generated for the second half of LUTs, and using these reversed bits for accessing respective first half of LUTs to get the required inner product. Then the algorithm for realizing the second half of LUTs becomes,

$$IP_{SH}(l,m) = \sum_{l=0}^{\left(\frac{L}{2}\right)^{-1}} \sum_{m=0; j=M-1}^{m=M-1; j=0} h(m+lM) x_b (j+lM+(N/2)-1).$$
(14)

This equation shows that it utilizes first half of LUTs $(l = 0, 1, \dots, (L/2) - 1)$, hence the first half of coefficients, and the address bits are generated for second half of LUTs in the bit reversed order (since $i = M - 1, M - 2, \dots, 0$). Therefore the algorithm for the proposed reduced LUT decomposed DA FIR is ob-

$$j = M - 1, M - 2, \dots, 0$$
). Therefore the algorithm for the proposed reduced LUT decomposed DA FIR, is obtained by combining Equations (12a) and (14) and applying it in (11a),

$$y = \sum_{b=0}^{B-1} 2^{b} \cdot \left\{ IP_{FH}(l,m) + IP_{SH}(l,m) \right\}.$$
 (15)

where

$$IP_{FH}(l,m) = \sum_{l=0}^{\left(\frac{L}{2}\right)^{-1}} \sum_{m=0}^{M-1} h(m+lM) x_b(m+lM).$$
$$IP_{SH}(l,m) = \sum_{l=0}^{\left(\frac{L}{2}\right)^{-1}} \sum_{m=0; j=M-1}^{m=M-1; j=0} h(m+lM) x_b(j+lM+(N/2)-1).$$

Let LUT L shares with the LUT 1, and a3a2a1a0 be the address bits of LUT 1, b3b2b1b0 be the address bits of LUT L. Then the inner product corresponding to LUT L(b3b2b1b0) is accessed using LUT 1 with the address b0b1b2b3, that is LUT L(b3b2b1b0) = LUT 1(b0b1b2b3). Hence according to proposed method in (15), the number of LUTs needed, is reduced by a factor of about 2, that is L/2 for L even and |L/2|+1 for L odd.

3.2. Illustrative Example

Consider for example a symmetric 6-tap FIR filter. Let us chose L = 2 and M = 3 for N = 6. The decomposed

form as per (9), requires 2 LUTs of size 2^3 words. Let h(0), h(1), h(2), h(3), h(4), and h(5) be the symmetric coefficients, that is h(0) = h(5), h(1) = h(4), and h(2) = h(3). The precomputed values stored in the LUT 1 and LUT 2 with its corresponding address is shown in **Table 1**. Then by applying coefficient symmetry property, the row wise equivalent precomputed value for LUT 2 (column 4) in LUT 1 is given in the column 5. The corresponding address for fetching the LUT 1 for these equivalent values is shown in column 6 of the same table.

From this table it is understood that all the precomputed values of LUT 2 is available in LUT 1 and it is possible to realize inner product computation using LUT 2 by LUT 1 itself. Now row wise comparison of address bits of LUT 2 in column 3 with the corresponding address for LUT 1 in column 6 reveals that, the address bits in column 6 are in the bit reversed form of address bits in column 3. Therefore the inner product computation using LUT 2 can be performed using its equivalent LUT, LUT 1 itself, by reversing the address bits generated for LUT 2, and using this to access LUT 1 for the generation of inner product as explained in proposed reduced LUT decomposed DA based symmetric FIR filter implementation.

Therefore for an N-tap FIR filter with symmetric coefficients, when realized using the proposed reduced LUT DA algorithm (15) with N = LM, the number of LUTs are reduced from L to L/2 for L even and $\lfloor L/2 \rfloor + 1$ for L odd. In general the equivalent LUTs for L even and for L odd are tabulated in Table 2.

3.3. Result of Application of Proposed Algorithm to High Speed Decomposed DA FIR Filter

Application of this proposed algorithm to high speed decomposed DA FIR filter, that employs parallel LUTs, results in the reduction of LUTs from *BL* to B(L/2) for *L* even and $B(\lfloor L/2 \rfloor + 1)$ for *L* odd, thus area as well as speed optimization is done in proposed structure. Consequently the resulting structure would give area optimized result for high speed DA FIR and also speed optimization over conventional and decomposed DA FIR.

4. Proposed Structure for Symmetric FIR Filter

This section first describes about the conventional and decomposed DA form of FIR filter. Then describes about reduced LUT decomposed DA form of FIR filter using the proposed algorithm, followed by this, high speed DA FIR structure and the proposed modified form of address generation logic and LUT structure of this high speed

LUT 1		LUT 2		Equivalent value and corresponding address of LUT 1 for LUT 2		
Address	Precomputed stored value	Address	Precomputed stored value	Equivalent value	Corresponding address	
000	0	000	0	0	000	
001	<i>h</i> (0)	001	<i>h</i> (3)	<i>h</i> (2)	100	
010	<i>h</i> (1)	010	<i>h</i> (4)	<i>h</i> (1)	010	
011	h(1) + h(0)	011	h(4) + h(3)	h(2) + h(1)	110	
100	<i>h</i> (2)	100	<i>h</i> (5)	<i>h</i> (0)	001	
101	h(2) + h(0)	101	h(5) + h(3)	h(2) + h(0)	101	
110	h(2) + h(1)	110	h(5) + h(4)	h(1) + h(0)	011	
111	h(2) + h(1) + h(0)	111	h(5) + h(4) + h(3)	h(2) + h(1) + h(0)	000	

Table 2. Equivalent LUTs for symmetric FIR filter.	

Equivalent LUTs for L even	Equivalent LUTs for L odd			
LUT L equivalent to LUT 1 LUT L-1 equivalent to LUT 2 LUT L-2 equivalent to LUT 3	LUT L equivalent to LUT 1 LUT L-1 equivalent to LUT 2 LUT L-2 equivalent to LUT 3 LUT L-3 equivalent to LUT 4			
LUT L-4 equivalent to LUT 4 LUT L-4 equivalent to LUT 5	LUT L-4 equivalent to LUT 5 			
LUT (L/2)+1 equivalent to LUT (L/2)	LUT $\lfloor L/2 \rfloor + 1$			

DA FIR filter is described. The direct form of FIR filter is used for all the DA based implementation.

• Conventional DA FIR filter

In general an N-tap FIR filter requires, N registers (shift registers) of B bits wide for an input width of B for storing the input, and the delayed form of inputs. The least significant bit of each register is considered for forming the address bits ($x_b(0)$ as least significant bit (LSB) and $x_b(N-1)$ as most significant bit (MSB)) for accessing the LUTs as shown in Figure 1. In conventional DA based realization of an N-tap FIR filter, N-bit address is formed from input and its delayed form, and requires single LUT of size 2^N words for generating inner products. Then these are simply shift-accumulated for B bits to get the filter output. But this implementation becomes impractical for larger values of N as LUT size grows exponentially with N. This lead into the development of decomposed DA algorithm as explained in Section 2.

• Decomposed DA FIR filter and reduced LUT decomposed DA FIR filter

The general block diagram of the decomposed DA based N-tap FIR filter according to (9) and the general block diagram of the reduced LUT decomposed DA based symmetric N-tap FIR filter according to the proposed algorithm in (15) is shown in **Figure 2** and **Figure 3** respectively. Comparison of these two implementation shows that the number of LUTs in the proposed structure is reduced from *L* to L/2 for *L* even and $\lfloor L/2 \rfloor + 1$ (shown for *L* – even in **Figure 4**) for *L* odd, at the cost of additional address mapping logic circuit and dual port LUTs. However these additional logics do not affect the performance of the filter.

The major blocks of the decomposed DA FIR using proposed algorithm are address generation logic and address mapping logic, inner product generation unit using LUTs, pipelined adder array and shift – accumulator.

In addition it requires clock divider block to generate frequency clk/B from frequency clk, as this structure requires two different clock frequency signals for its operation.

Address generation logic is implemented using one parallel-in-serial-out shift register (PISOSR) and N-1 serial-in-serial-out shift registers (SISOSR) of B bits wide. The filter input signal (filter_in) of width B is loaded in parallel to PISOSR in synchronization with the clock signal clk/B. The same register performs serial-out operation in synchronization with the clock signal clk. Similarly all the SISOSRs operating in synchronization with clock signal clk. Therefore output bit of these shift registers forming the address bits for accessing LUTs. Address mapping logic is needed for lower half of address bits, which just performs bit reversal task to get the required address to make use of upper half of LUTs for realizing lower half of LUTs. Each LUT is stored with the all the possible combinations precomputed values of corresponding decomposed coefficients for inner product generation. The output of LUTs is added using pipelined adder array to get the inner product corresponding to particular bit position. Finally these inner products are shift – accumulated for all B bit positions to get the filter output and shift-accumulator is reset once in B cycles. Therefore the filter output is made available once in B cycles only, which limits the speed of operation, especially when B becomes larger.

High speed DA FIR filter and its proposed modified structure

The frequency of operation of the decomposed DA FIR and reduced LUT decomposed DA FIR is improved by using parallel LUTs as stated in previous section, resulting in high speed DA FIR filter. First the structure of



Figure 1. Conventional DA FIR filter.

S. C. Prasanna, S. P. J. V. Rani



Figure 3. The general block diagram of reduced LUT decomposed DA FIR filter using proposed algorithm.



Figure 4. High speed decomposed decomposed DA FIR filter.

the high speed decomposed DA FIR filter for an N-tap FIR filter with N = LM derived from Figure 2 is shown in Figure 4.

The entire structure operates at single clock frequency (not explicitly shown in figure) and the output is computed in single clock period. Here the input and its delayed form are stored in parallel- in parallel-out shift registers (PIPOSR), it is represented as x(0), x(1), ..., x(LM-1) [x(0) = x(n), x(1) = x(n-1), ..., x(LM-1) = x(n-(LM-1))] in figure. The B bit output of these registers, grouped into form L number of M bit address for accessing respective LUTs.

Each LUT is duplicated B-1 times. That is LUT_{1_1} , LUT_{1_2} , ..., $LUT_{1_2}B-1$ are the duplication of LUT_{1_0} . The LUT_{1_0} is accessed using the address bits formed from the least significant bit (LSB) of x(0), x(1), ..., x(M-1). The LUT_{1_1} is accessed using the address bits formed from the second LSB (bit position 1) of x(0), x(1), ..., x(M-1) and so on. Similarly $LUT_{1_2}B-1$ is accessed using the address bits formed from the most significant bit (MSB) (bit position B-1) of x(0), x(1), ..., x(M-1).

In a similar manner in the second set of LUTs, LUT_2_1 , LUT_2_2 , ..., LUT_2_B-1 are the duplication of LUT_2_0 , and are accessed using the respective bits of x(M), x(M+1), ..., x(2M-1) and so on. Finally LUT_{L_1} , LUT_{L_2} , ..., $LUT_{L_2}B-1$ are the duplication of $LUT_{L_0}0$ and are accessed using the respective bits of x(LM-M), x(LM-M+1), ..., x(LM-1).

The output of all the LUTs (LUT₁_0, LUT₂_0, ..., LUT_L_0) corresponding to bit position 0 are added using adder array_0, the output of all the LUTs (LUT₁_1, LUT₂_1, ..., LUT_L_1) corresponding to bit position 1 are added using adder array_1, and so on. Finally the output of all the LUTs (LUT₁_B-1, LUT₂_B-1, ..., LUT_L_B-1) corresponding to bit position B-1 are added using adder array_B-1. Next the output of adder array_1 is shifted left by one bit position, the output of adder array_2 is shifted left by two bit positions and so on. Finally the output of adder array_B-1 is shifted left by B-1 bit positions. Then all these shifted outputs' and the output of adder array_0 are added using another adder array to get the filter output as shown in Figure 4.

Therefore it is understood that the speed of operation in the decomposed DA FIR is improved by employing parallel access using multiple duplicate LUTs, and combining their outputs using multiple adder arrays to yield the output in single clock period, which eliminates the need of shift-accumulation unit as in conventional and decomposed DA FIR. However this speed improvement is achieved at the expense of additional hardware cost. This hardware cost is reduced in the proposed structure by applying reduced LUT decomposed DA approach according to (15) over the high speed decomposed DA FIR shown in **Figure 4**. The modification is done over address generation logic and LUT input-output structure, and the remaining circuitry is the same in the proposed design. The proposed modified address generation logic and LUT structure is shown in **Figure 5**.

The comparison of Figure 5 with Figure 4 shows that the number of LUTs in the proposed structure is reduced from BL to B(L/2). But the proposed structure requires dual port LUTs, whereas the high speed decomposed DA requires single port LUT. The address generated for accessing second half of LUTs are bit reversed using address mapping logic, which are then used to access respective equivalent LUTs as shown in Figure 5. Let i and j be the integers, then LUT_{i_outj} corresponds to output of i-th LUT for the address bits generated from bit position j. Similarly rLUT_{i_outj} corresponds to output of i-th LUT for reversed form of address bits generated from bit position j for upper half of equivalent LUT.

The precomputed values stored in LUT₁ j and LUT₂ j corresponding to all bit positions, for N = 16, L = M = 4, and coefficient and input width of W and B bits respectively are shown in Figure 6. The inputs A₁ and rA₁ to LUT₁ j and the inputs A₂ and rA₂ to LUT₂ j are the address bits generated for upper half and reversed address bits corresponding to lower half of LUTs respectively.

The LUT outputs shown in **Figure 5**, $LUT_{i_out}0$, and $rLUT_{i_out}0$ for i = 1, 2, 3, ..., L/2 are given to adder array_0, the outputs $LUT_{i_out}1$, and $rLUT_{i_out}1$ for i = 1, 2, 3, ..., L/2 are given to adder array_1, and so on. Similarly the outputs LUT_{i_outB-1} , and $rLUT_{i_outB-1}$ for i = 1, 2, 3, ..., L/2 are given to adder array_B-1. The rest of the process is similar to high speed decomposed DA FIR as explained before.

5. Results and Discussion

The proposed reduced parallel LUT DA based structure for symmetric FIR filter for N = 16, L = M = 4, and with coefficient and input word length of 8 bits is implemented on Xilinx Virtex-5 XC5VSX95T-1FF1136 field-programmable gate array device, and the result is tabulated in **Table 3**. For the purpose of performance comparison, number of slice registers (NSR), number of slice LUTs (NSL), number of slices (NS), delay, frequency and slice-delay product (SDP) improvement percentage of the proposed design is compared with the existing high throughput DA based structure in [13] and DA based systolic structure in [12]. The structure in [13] also employs parallel LUTs to speed up the computation similar to the proposed structure. Area optimization is done in the proposed design when compared to [13], by using the proposed reduced LUT decomposed DA algorithm for symmetric FIR filter. From **Table 3**, it is seen that the proposed structure provides area as well as speed improvement over earlier designs. The proposed structure requires 60%, 34.3%, and 27.4% less NSR,

NSL, and NS respectively compared to [13], resulting in SDP improvement of 53.98% for the proposed design. Similarly comparison with sequential access LUT design in [12], shows that the proposed structure offers 60.62% rise in speed of operation over [12]. The area utilization metrics such as NSR, NSL, and NOS also less for the proposed structure compared to systolic structure.

6. Conclusions

The FIR digital filters are the core unit in many digital signal processing (DSP) applications and communication systems. The implementation of FIR filter through one of the multiplier less DA based approach is considered in



Figure 5. Proposed Modified address generation logic and LUT structure for high speed decomposed DA FIR filter.



Figure 6. LUTs with precomputed stored values for N = 16, L = M = 4.

 Table 3. Performance comparison of proposed design with existing design implemented on Virtex-5 FPGA (XC5VSX95T-1FF1136).

Design Method	NSR	NSL	NS	Delay (ns)	Frequency (MHz)	SDP
Proposed	448	496	225	1.646	607	370
High throughput DA based $(R = 1)$ [13]	1120	755	310	2.27	440	804
Systolic DA based [12]	688	833	275	4.17	239	1146

this work. The algorithm for conventional DA based implementation is described. The limitation of this algorithm is exponential increase of LUT size with filter taps. Then the algorithm, which overcomes this limitation, called decomposed DA based implementation is discussed, which partitions single LUT into many LUTs of smaller size at the cost of additional adder array. We proposed and derived algorithm to optimize the area further, called reduced LUT decomposed DA based implementation for symmetric FIR filter, in which the number of LUTs were further reduced by a factor of about 2. This approach is implemented over high speed DA based FIR filter, which employs parallel LUTs for each decomposed group L, to speed up the computation in the decomposed DA based structure. Thus the resulting proposed structure is an area and speed efficient structure for the implementation of symmetric FIR filter.

The 16-tap FIR filter with L = M = 4, and input and coefficient widths of 8 bits is considered for implementation to analyze the performance with existing high throughput DA based design and with systolic DA based design, implemented over Xilinx Virtex-5, XC5VSX95T-1FF1136 FPGA device. The performance comparison of area utilization indices, NSR, NSL, and NS of the proposed structure with high throughput DA based structure, implies that the proposed structure requires 60%, 34.3%, and 27.4% less NSR, NSL, and NS respectively, resulting in an average area improvement of around 40%. The proposed design also requires lesser clock period than the high throughput DA based design. It is also found that the proposed design offers 60.5% less delay and requires less area than the systolic DA based design, and can support up to the maximum operating frequency of 607 MHz.

References

- [1] Proakis, J.G. and Manolakis, D.G. (1996) Digital Signal Processing: Principles, Algorithms and Applications. Prentice-Hall, Upper Saddle River.
- [2] Antoniou, A. (1993) Digital Filters: Analysis, Design, and Applications. McGraw-Hill, New York.
- [3] Ashour, M.A. and Saleh, H.I. (2000) An FPGA Implementation Guide for Some Different Types of Serial-Parallel Multiplier Structures. *Microelectronics Journal*, **31**, 161-168. <u>http://dx.doi.org/10.1016/S0026-2692(99)00110-X</u>
- [4] Qasim, S.M., Telba, A.A. and AlMazroo, A.Y. (2010) FPGA Design and Implementation of Matrix Multiplier Architectures for Image and Signal Processing Applications. *International Journal of Computer Science and Network Security*, **10**, 168-176.
- [5] Obeid, A.M., Qasim, S.M., BenSaleh, M.S., Marrakchi, Z., Mehrez, H., Ghariani, H. and Abid, M. (2014) Flexible Reconfigurable Architecture for DSP Applications. *Proceedings of 27th IEEE International System-on-Chip Confe*rence (SOCC), Qasim, September 2014, 204-209.
- [6] White, S.A. (1989) Applications of the Distributed Arithmetic to Digital Signal Processing: A Tutorial Review. IEEE ASSP Magazine, 6, 5-19. <u>http://dx.doi.org/10.1109/53.29648</u>
- [7] Croisier, A., Esteban, D.J., Levilion, M.E. and Rizo, V. (1973) Digital Filter for PCM Encoded Signals. US Patent 3 777 130, 4 December 1973.
- [8] Peled, A. and Liu, B. (1974) A New Hardware Realization of Digital Filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 22, 456-462. <u>http://dx.doi.org/10.1109/TASSP.1974.1162619</u>
- [9] Allred, D., Yoo, H., Krishnan, V., Huang, W. and Anderson, D. (2004) A Novel High Performance Distributed Arithmetic Adaptive Filter Implementation on an FPGA. *Proceedings of IEEE International Conference on Acoustics*, *Speech, and Signal Processing (ICASSP*'04), 2004, 161-164. <u>http://dx.doi.org/10.1109/icassp.2004.1327072</u>
- [10] Allred, D.J., Yoo, H., Krishnan, V., Huang, W. and Anderson, D.V. (2005) LMS Adaptive Filters Using Distributed Arithmetic for High Throughput. *IEEE Transactions on Circuits and Systems—I: Regular Papers*, 52, 1327-1337. http://dx.doi.org/10.1109/TCSI.2005.851731
- [11] Meher, P.K. and Park, S.Y. (2011) High-Throughput Pipelined Realization of Adaptive FIR Filter Based on Distributed Arithmetic. *IEEE/IFIP* 19th International Conference on VLSI and System-on-Chip, Hong Kong, 3-5 October 2011, 428-433. <u>http://dx.doi.org/10.1109/vlsisoc.2011.6081621</u>
- [12] Meher, P.K., Chandrasekaran, S. and Amira, A. (2008) FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic. *IEEE Transactions on Signal Processing*, 56, 3009-3017. http://dx.doi.org/10.1109/TSP.2007.914926
- [13] Park, S.Y. and Meher, P.K. (2014) Efficient FPGA and ASIC Realizations of a DA-Based Reconfigurable FIR Digital Filter. *IEEE Transactions on Circuits and Systems—II: Express Briefs*, **61**, 511-515.
- [14] Meyer-Baese, U. (2003) Digital Signal Processing with Field Programmable Gate Arrays. Springer Pvt. Ltd., India.