
Delay Efficient Fir Filter Architecture for Fixed and Re-Configurable Applications

G.Shabnum Sultana & R. Shobha Rani

- 1) **PG Scholar**, VLSI System Design, Ellenki Institute Of Engineering & Technology.
- 2) Associate Professor, Ellenki Institute Of Engineering & Technology, Patalguda, Patancheru, Medak (Dist), India.

Abstract—Transpose form finite-impulse response (FIR) filters are inherently pipelined and support multiple constant multiplications (MCM) technique that results in significant saving of computation. However, transpose form configuration does not directly support the block processing unlike direct form configuration. In this paper, we explore the possibility of realization of block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, we have derived a flow graph for transpose form block FIR filter with optimized register complexity. A generalized block formulation is presented for transpose form FIR filter. We have derived a general multiplier-based architecture for the proposed transpose form block filter for reconfigurable applications. A low-complexity design using the MCM scheme is also presented for the block implementation of fixed FIR filters. The proposed structure involves significantly less areadelay product (ADP) than the existing block implementation of direct-form structure for medium or large filter lengths, while for the short-length filters, the block implementation of direct-form FIR structure has less ADP than the proposed structure. Application specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involves 42% less ADP than the best available FIR filter structure proposed for reconfigurable

applications. For the same filter length and the same block size, the proposed structure involves 13% less ADP than that of the existing direct-form block FIR structure.

Keywords— Block processing, finite-impulse response (FIR) filter, reconfigurable architecture, VLSI.

I.INTRODUCTION

FINITE-IMPULSE response (FIR) digital filter is widely used in several digital signal processing applications, such as speech processing, loud speaker equalization, echo cancellation, adaptive noise cancellation, and various communication applications, including software-defined radio (SDR) and so on [1]. Many of these applications require FIR filters of large order to meet the stringent frequency specifications [2]–[4]. Very often these filters need to support high sampling rate for high-speed digital communication [5]. The number of multiplications and additions required for each filter output, however, increases linearly with the filter order. Since there is no redundant computation available in the FIR filter algorithm, real-time implementation of a large order FIR filter in a resource constrained environment is a challenging task. Filter coefficients very often remain constant and known a priori in signal processing applications. This feature has been utilized to reduce the complexity of realization of multiplications. Several designs have been suggested by various researchers for

efficient realization of FIR filters (having fixed coefficients) using distributed arithmetic (DA) [18] and multiple constant multiplication (MCM) methods [7], [11]–[13]. DA-based designs use lookup tables (LUTs) to store precomputed results to reduce the computational complexity. The MCM method on the other hand reduces the number of additions required for the realization of multiplications by common sub expression sharing, when a given input is multiplied with a set of constants. The MCM scheme is more effective, when a common operand is multiplied with more number of constants. Therefore, the MCM scheme is suitable for the implementation of large order FIR filters with fixed coefficients. But, MCM blocks can be formed only in the transpose form configuration of FIR filters.

Block-processing method is popularly used to derive high-throughput hardware structures. It not only provides throughput-scalable design but also improves the area-delay efficiency. The derivation of block-based FIR structure is straightforward when direct-form configuration is used [16], whereas the transpose form configuration does not directly support block processing. But, to take the computational advantage of the MCM, FIR filter is required to be realized by transpose form configuration. Apart from that, transpose form structures are inherently pipelined and supposed to offer higher operating frequency to support higher sampling rate.

There are some applications, such as SDR channelizer, where FIR filters need to be implemented in a reconfigurable hardware to support multi standard wireless communication [6]. Several designs have been suggested during the last decade for efficient realization of reconfigurable FIR (RFIR) using general multipliers and constant multiplication schemes [7]–[10]. A RFIR filter architecture using computation sharing vector-scaling

technique has been proposed in [7]. Chen and Chiueh [8] have proposed a canonic sign digit (CSD)-based RFIR filter, where the nonzero CSD values are modified to reduce the precision of filter coefficients without significant impact on filter behavior. But, the reconfiguration overhead is significantly large and does not provide an area-delay efficient structure. The architectures in [7] and [8] are more appropriate for lower order filters and not suitable for channel filters due to their large area complexity. Constant shift method (CSM) and programmable shift method have been proposed in [9] for RFIR filters, specifically for SDR channelizer. Recently, Park and Meher [10] have proposed an interesting DA-based architecture for RFIR filter. The existing multiplier-based structures use either direct form configuration or transpose form configuration. But, the multiplier-less structures of [9] use transpose form configuration, whereas the DA-based structure of [10] uses direct-form configuration. But, we do not find any specific block-based design for RFIR filter in the literature. A block-based RFIR structure can easily be derived using the scheme proposed in [15] and [16]. But, we find that the block structure obtained from [15] and [16] is not efficient for large filter lengths and variable filter coefficients, such as SDR channelizer. Therefore, the design methods proposed in [15] and [16] are more suitable for 2-D FIR filters and block least mean square adaptive filters. In this paper, we explore the possibility of realization of block FIR filter in transpose form configuration in order to take advantage of the MCM schemes and the inherent pipelining for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. The main contributions of this paper are as follows.

- 1) Computational analysis of transpose form configuration of FIR filter and derivation of flow graph for transpose form block FIR filter with reduced register complexity.

- 2) Block formulation for transpose form FIR filter.
- 3) Design of transpose form block filter for reconfigurable applications.
- 4) A low-complexity design method using MCM scheme for the block implementation of fixed FIR filters.

II. COMPUTATIONAL ANALYSIS AND MATHEMATICAL FORMULATION OF BLOCK TRANSPOSE FORM FIR FILTER

The output of an FIR filter of length N can be computed using the relation

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

The computation of (1) can be expressed by the recurrence relation

$$Y(z) = [z^{-1}(\dots(z^{-1}(z^{-1}h(N-1) + h(N-2)) + h(N-3)) \dots + h(1)) + h(0)]X(z). \quad (2)$$

A. Computational Analysis The data-flow graphs (DFG-1 and DFG-2) of transpose form FIR filter for filter length N = 6, as shown in Fig. 1, for

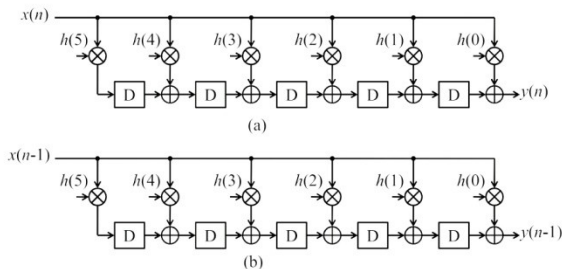


Fig.1. DFG of transpose form structure for N = 6. (a) DFG-1 for output y(n). (b) DFG-2 for output y(n - 1).

a block of two successive outputs {y(n), y(n - 1)} that are derived from (2). The product values and their accumulation paths in DFG-1 and DFG-2 of Fig. 1 are shown in data-flow tables (DFT-1 and DFT-2) of Flow table.1

ecs	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆
1	x(n-5)h(5)	x(n-5)h(4)	x(n-5)h(3)	x(n-5)h(2)	x(n-5)h(1)	x(n-5)h(0)
2	x(n-4)h(5)	x(n-4)h(4)	x(n-4)h(3)	x(n-4)h(2)	x(n-4)h(1)	x(n-4)h(0)
3	x(n-3)h(5)	x(n-3)h(4)	x(n-3)h(3)	x(n-3)h(2)	x(n-3)h(1)	x(n-3)h(0)
4	x(n-2)h(5)	x(n-2)h(4)	x(n-2)h(3)	x(n-2)h(2)	x(n-2)h(1)	x(n-2)h(0)
5	x(n-1)h(5)	x(n-1)h(4)	x(n-1)h(3)	x(n-1)h(2)	x(n-1)h(1)	x(n-1)h(0)
6	x(n)h(5)	x(n)h(4)	x(n)h(3)	x(n)h(2)	x(n)h(1)	x(n)h(0)

(a)

ecs	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆
1	x(n-6)h(5)	x(n-6)h(4)	x(n-6)h(3)	x(n-6)h(2)	x(n-6)h(1)	x(n-6)h(0)
2	x(n-5)h(5)	x(n-5)h(4)	x(n-5)h(3)	x(n-5)h(2)	x(n-5)h(1)	x(n-5)h(0)
3	x(n-4)h(5)	x(n-4)h(4)	x(n-4)h(3)	x(n-4)h(2)	x(n-4)h(1)	x(n-4)h(0)
4	x(n-3)h(5)	x(n-3)h(4)	x(n-3)h(3)	x(n-3)h(2)	x(n-3)h(1)	x(n-3)h(0)
5	x(n-2)h(5)	x(n-2)h(4)	x(n-2)h(3)	x(n-2)h(2)	x(n-2)h(1)	x(n-2)h(0)
6	x(n-1)h(5)	x(n-1)h(4)	x(n-1)h(3)	x(n-1)h(2)	x(n-1)h(1)	x(n-1)h(0)

(b)

Flow table.1. (a) DFT of multipliers of DFG shown in Fig. 1(a) corresponding to output y(n). (b) DFT of multipliers of DFG shown in Fig. 1(b) corresponding to output y(n - 1). Arrow: accumulation path of the products.

The arrows in DFT-1 and DFT-2 of Flow table.1 represent the accumulation path of the products. We find that five values of each column of DFT-1 are same as those of DFT-2 (shown in gray color in Fig. 2). These redundant computation of DFG-1 and DFG-2 can be avoided using nonoverlapped sequence of input blocks, as shown in Flow table. 2. DFT-3 and DFT-4 of DFG-1 and DFG-2 for non overlapping input blocks are, respectively, shown in Fig. 3(a) and (b). As shown in Flow table. 2(a) and (b), DFT-3 and

DFT-4 do not involve redundant computation. It is easy to find that the entries in gray cells in DFT-3 and DFT-4 of Flow table. 2(a) and (b) correspond to the output $y(n)$, whereas the other entries of DFT-3 and DFT-4 correspond to $y(n-1)$. The DFG of Fig. 1 needs to be transformed appropriately to obtain the computations according to DFT-3 and DFT-4.

ccs	M_1	M_2	M_3	M_4	M_5	M_6
1	$x(n-10)h(5)$	$x(n-10)h(4)$	$x(n-10)h(3)$	$x(n-10)h(2)$	$x(n-10)h(1)$	$x(n-10)h(0)$
2	$x(n-8)h(5)$	$x(n-8)h(4)$	$x(n-8)h(3)$	$x(n-8)h(2)$	$x(n-8)h(1)$	$x(n-8)h(0)$
3	$x(n-6)h(5)$	$x(n-6)h(4)$	$x(n-6)h(3)$	$x(n-6)h(2)$	$x(n-6)h(1)$	$x(n-6)h(0)$
4	$x(n-4)h(5)$	$x(n-4)h(4)$	$x(n-4)h(3)$	$x(n-4)h(2)$	$x(n-4)h(1)$	$x(n-4)h(0)$
5	$x(n-2)h(5)$	$x(n-2)h(4)$	$x(n-2)h(3)$	$x(n-2)h(2)$	$x(n-2)h(1)$	$x(n-2)h(0)$
6	$x(n)h(5)$	$x(n)h(4)$	$x(n)h(3)$	$x(n)h(2)$	$x(n)h(1)$	$x(n)h(0)$

(a)

ccs	M_1	M_2	M_3	M_4	M_5	M_6
1	$x(n-11)h(5)$	$x(n-11)h(4)$	$x(n-11)h(3)$	$x(n-11)h(2)$	$x(n-11)h(1)$	$x(n-11)h(0)$
2	$x(n-9)h(5)$	$x(n-9)h(4)$	$x(n-9)h(3)$	$x(n-9)h(2)$	$x(n-9)h(1)$	$x(n-9)h(0)$
3	$x(n-7)h(5)$	$x(n-7)h(4)$	$x(n-7)h(3)$	$x(n-7)h(2)$	$x(n-7)h(1)$	$x(n-7)h(0)$
4	$x(n-5)h(5)$	$x(n-5)h(4)$	$x(n-5)h(3)$	$x(n-5)h(2)$	$x(n-5)h(1)$	$x(n-5)h(0)$
5	$x(n-3)h(5)$	$x(n-3)h(4)$	$x(n-3)h(3)$	$x(n-3)h(2)$	$x(n-3)h(1)$	$x(n-3)h(0)$
6	$x(n-1)h(5)$	$x(n-1)h(4)$	$x(n-1)h(3)$	$x(n-1)h(2)$	$x(n-1)h(1)$	$x(n-1)h(0)$

(b)

Flow table.2. DFT of DFG-1 and DFG-2 for three non overlapped input blocks $[x(n), x(n-1)]$, $[x(n-2), x(n-3)]$, and $[x(n-4), x(n-5)]$. (a) DFT-3 for computation of output $y(n)$. (b) DFT-4 for computation of output $y(n-1)$.

B. DFG Transformation

The computation of DFT-3 and DFT-4 can be realized by DFG-3 of non overlapping blocks, as shown in Fig. 2. We refer it to block transpose form type-I

configuration of block FIRfilter.

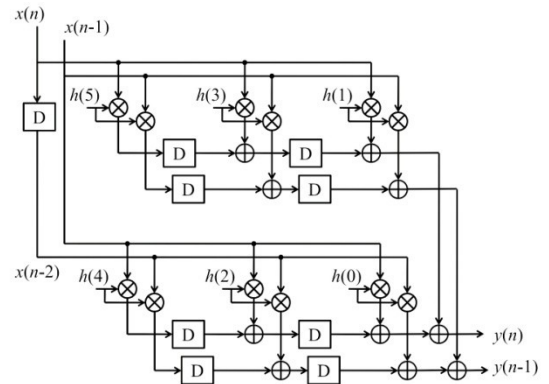


Fig.2. Merged DFG (DFG-3: transpose form type-I configuration for block FIR structure).

The DFG-3 can be retimed to obtain the DFG-4 of Fig. 3, which is referred to block transpose form type-II configuration. Note that both type-I and type-II configurations involve the same number of multipliers and adders, but type-II configuration involves nearly L times less delay elements than those of

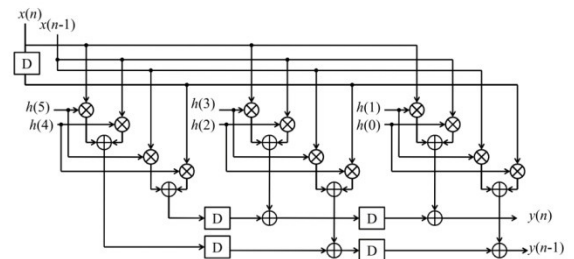


Fig.3. DFG-4 (retimed DFG-3) transpose form type-II configuration for block FIR structure.

type-I configuration. We have, therefore, used block transpose form type-II configuration to derive the proposed structure. In Section II-C, we present mathematical formulation of block transpose form type-II FIR filter for a generalized formulation of the concept of block-based computation of transpose form FIR filters.

III. PROPOSED STRUCTURES

There are several applications where the coefficients of FIR filters remain fixed, while in some other applications, like SDR channelizer that requires separate FIR filters of different specifications to extract one of the desired narrowband channels from the wideband RF front end. These FIR filters need to be implemented in a RFIR structure to support multi standard wireless communication [6]. In this section, we present a structure of block FIR filter for such reconfigurable applications. In this section, we discuss the implementation of block FIR filter for fixed filters as well using MCM scheme.

A. Proposed Structure for Transpose Form Block FIR Filter for Reconfigurable Applications

The proposed structure for block FIR filter is [based on the recurrence relation of (12)] shown in Fig. 4 for the block size $L = 4$. It consists of one coefficient selection unit (CSU), one register unit (RU), M number of inner product units (IPUs), and one pipeline adder unit (PAU). The CSU stores coefficients of all the filters to be used for the reconfigurable application.

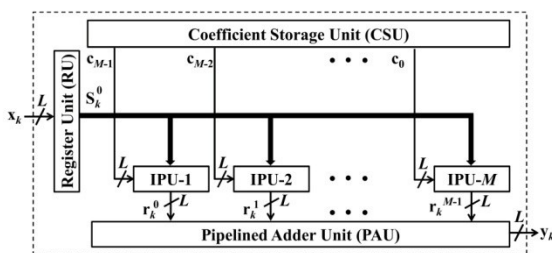


Fig.4. Proposed structure for block FIR filter.

It is implemented using N ROM LUTs, such that filter coefficients of any particular channel filter are obtained in one clock cycle, where N is the filter

length. The RU [shown in Fig. 5(a)] receives x_k during the k th cycle and produces L rows of S_0^k in parallel. L rows of S_0^k are transmitted to M IPUs of the proposed structure. The M IPUs also receive M short-weight vectors from the CSU such that during the k th cycle, the $(m + 1)$ th IPU receives the weight vector c_{M-m-1} from the CSU and L rows of S_0^k form the RU. Each IPU performs matrix-vector product of S_0^k with the short-weight vector c_m , and computes a block of L partial filter outputs (r_m^k). Therefore, each IPU performs L inner-product computations of L rows of S_0^k with a common weight vector c_m . The structure of the $(m + 1)$ th IPU is shown in Fig. 5(b). It consists of L number of L -point inner-product cells (IPCs).

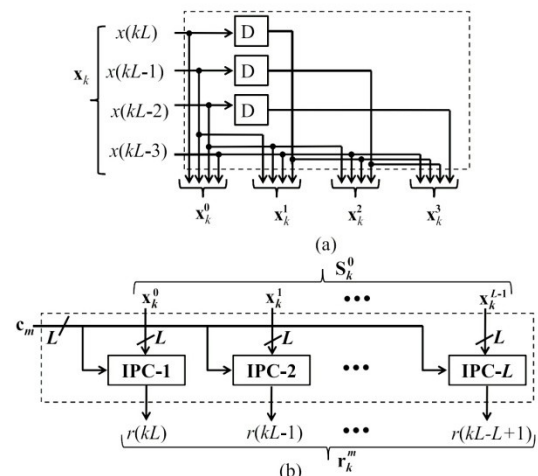


Fig.5. (a) Internal structure of RU for block size $L = 4$. (b) Structure of $(m + 1)$ th IPU.

The $(l + 1)$ th IPC receives the $(l + 1)$ th row of S_0^k and the coefficient vector c_m , and computes a partial result of inner product $r^k(l)$, for $0 \leq l \leq L - 1$. Internal structure of $(l + 1)$ th IPC for $L = 4$ is shown in Fig. 6(a). All the M IPUs work in parallel and produce M blocks of result (r_m^k). These partial inner products

are added in the PAU [shown in Fig. 6(b)] to obtain a block of L filter outputs. In each cycle, the proposed structure receives a block of L inputs and produces a block of L filter outputs, where the duration of each cycle is $T = T_M + T_A + T_{FA} \log_2 L$, T_M is one multiplier delay, T_A is one adder delay, and T_{FA} is one full-adder delay.

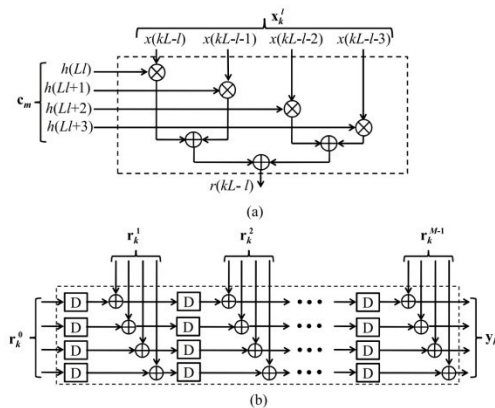


Fig.6. (a) Internal structure of $(l + 1)$ th IPC for $L = 4$.
(b) Structure of PAU for block size $L = 4$.

B. MCM-Based Implementation of Fixed-Coefficient FIR Filter

We discuss the derivation of MCM units for transpose form block FIR filter, and the design of proposed structure for fixed filters. For fixed-coefficient implementation, the CSU of Fig. 4 is no longer required, since the structure is to be tailored for only one given filter. Similarly, IPU's are not required. The multiplications are required to be mapped to the MCM units for a low-complexity realization. In the following, we show that the proposed formulation for MCM-based implementation of block FIR filter makes use of the symmetry in input matrix S_0^k to perform horizontal and vertical common sub expression elimination [17] and to minimize the number of shift-

add operations in the MCM blocks. The recurrence relation of (12) can alternatively be expressed as

$$Y(z) = z^{-1} \cdots z^{-1} (z^{-1} r_{M-1} + r_{M-2} + r_{M-3}) + \cdots + r_1 + r_0. \quad (13)$$

The M intermediate data vectors r_m , for $0 \leq m \leq M - 1$ can be computed using the relation.

$$R = S_k^0 \cdot C$$

The proposed MCM-based structure for FIR filters for block size $L = 4$ is shown in Fig. 7 for the purpose of illustration. The MCM-based structure (shown in Fig. 7) involves six MCM blocks corresponding to six input samples. Each MCM block produces the necessary product terms as listed in Table I. The sub expressions of the MCM blocks are shift added in the adder network to produce the inner-product values $(r_{l,m})$, for $0 \leq l \leq L - 1$ and $0 \leq m \leq (N/L) - 1$ corresponding to the matrix product of (14). The inner-product values are finally added in the PAU of Fig. 6(b) to obtain a block of filter output.

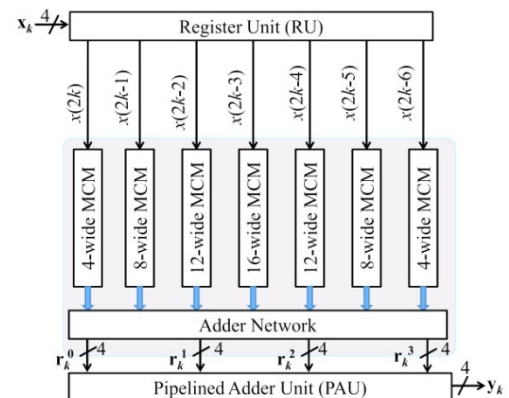
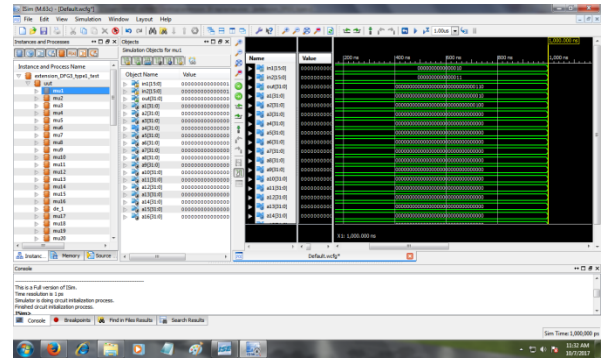
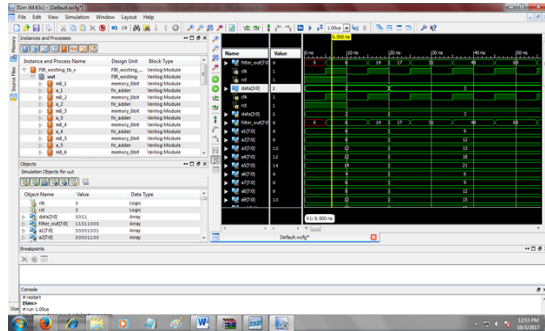


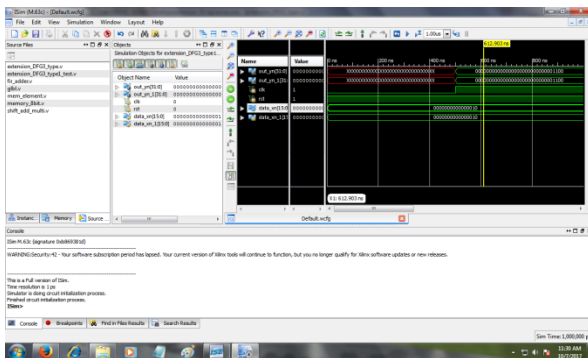
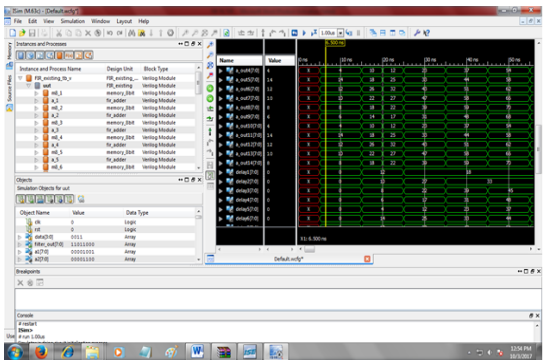
Fig.7. Proposed MCM-based structure for fixed FIR filter of block size $L = 4$ and filter length $N = 16$.

IV. SIMULATION RESULTS



V. CONCLUSION

In this paper, we have explored the possibility of realization of block FIR filters in transpose form configuration for AREA, DELAY efficient realization of both fixed and reconfigurable applications. A generalized block formulation is presented for transpose form block FIR filter, and based on that we have derived transpose form block filter for reconfigurable applications. We have presented a scheme to identify the MCM blocks for horizontal and vertical sub expression elimination in the proposed block FIR filter for fixed coefficients to reduce the computational complexity. Performance comparison shows that the proposed structure involves significantly less ADP and less EPS than the existing block direct-form structure for medium or large filter lengths while for the short-length filters, the existing block direct-form structure has less ADP and less EPS than the proposed structure. Application-specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involve 42% less ADP and 40% less EPS than the best available FIR filter structure of [10] for reconfigurable applications. For the same filter length and the same block size, the proposed structure involves 13% less ADP and 12.8% less EPS than that of the existing direct-form block FIR structure of [15].



VI. FUTURE SCOPE

FIR filters are widely used in digital signal processing and can be implemented using programmable digital processors. But in the realization of large order filters the speed, cost, and flexibility is affected because of complex computations.

The future scope of this work includes the following:

- The A/D and D/A converter can be interfaced within the fpga.
- The optimization of the design can be done in terms of area occupied on the chip.

REFERENCES

- [1] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.
- [2] T. Hentschel and G. Fettweis, "Software radio receivers," in *CDMA Techniques for Third Generation Mobile Systems*. Dordrecht, The Netherlands: Kluwer, 1999, pp. 257–283.
- [3] E. Mirchandani, R. L. Zinser, Jr., and J. B. Evans, "A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals]," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 10, pp. 681–694, Oct. 1995.
- [4] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system," in *Proc. IEEE Southeastcon*, Apr. 1993, p. 1–6.
- [5] J. Mitola, *Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering*. New York, NY, USA: Wiley, 2000.
- [6] A. P. Vinod and E. M. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1669–1675, Jul. 2006.
- [7] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," *IEEE J. Solid State Circuits*, vol. 39, no. 2, pp. 348–357, Feb. 2004. [8] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 617–621, Aug. 2006.
- [9] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 2, pp. 275–288, Feb. 2010.
- [10] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 7, pp. 511–515, Jul. 2014.
- [11] P. K. Meher, "Hardware-efficient systolization of DA-based calculation of finite digital convolution," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 707–711, Aug. 2006.
- [12] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3009–3017, Jul. 2008.