A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications

Basant Kumar Mohanty, Senior Member, IEEE and Pramod K. Meher, Senior Member, IEEE

Abstract—Transpose-form finite impulse response (FIR) structures are inherently pipelined and support multiple constant multiplication (MCM) 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-from configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on detail computational analysis of transpose-form configuration of FIR filter we have derived a flow-graph for transpose-from block FIR filter with optimized register complexity. A generalized block formulation is presented for transpose-from 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 MCM scheme is also presented for the block implementation of fixed FIR filters. Performance comparison shows that the proposed structure involves significantly less area-delay product (ADP) and less energy per sample (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 FIR structure has less ADP and less EPS than the proposed structure. ASIC 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 structure proposed 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-from block FIR structure. Based on these findings, we present a scheme for the selection of direct-form and transpose-form configuration based on the filter lengths and block-length for obtaining area-delay and energy efficient block FIR structures.

Index Terms—Reconfigurable Architecture, Block Processing, FIR Filter, 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) etc. [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) [22] and multiple constant multiplication (MCM) methods [10], [13]–[16]. DA-based designs use look-up-tables (LUTs) to store pre-computed 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, 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. Not only does it provide throughput-scalable design but also improves the area-delay efficiency. The derivation of block-based FIR structure is straight-forward when direct-from configuration is used [21] 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 suppose 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]–[12], [17], [18]. A programmable multiply-accumulator based processor is proposed in [7] for FIR filtering. The area and power requirement of these architectures are significantly large and, therefore, they are not suitable for SDR channelizer. The structure of [9] is multiplier-based and uses poly-phase decomposition scheme. In [10], a reconfigurable FIR filter architecture using computation sharing vector-scaling technique of [8] has been proposed. In [11], a programmable canonical signed digit (CSD) based architecture was proposed using Booth encoding to generate partial products and Wallace tree adder for

B. K. Mohanty with the Department of Electronics and Communication Engineering, Jayceee University of Engineering and Technology, Raghotharh, Guna, Madhya Pradesh, India-473226, (email: bk.mohanti@juet.ac.in.
P. K. Meher is with School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, 639798, email: aspkmeher@ntu.edu.sg, URL:http://www.ntu.edu.sg/homes/aspkmeher/.

Pramod K. Meher with School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, 639798, email: aspkmeher@ntu.edu.sg, URL:http://www.ntu.edu.sg/homes/aspkmeher/.
addition of partial products. Chen et al [12] have proposed a CSD-based reconfigurable FIR filter where the non-zero 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 of [8]–[12] are more appropriate for lower order filters and they are not suitable for channel filters due to their large area complexity. Constant shift method (CSM) and programmable shift method (PSM) have been proposed in [16], [17] for RFIR filters specifically for SDR channelizer. Recently, Park et al. [18] have proposed an interesting distributed arithmetic (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 [16], [17] use transpose-form configuration whereas the DA-based structure of [18] 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 [20], [21]. But, we find that the block structure obtained from [20], [21] is not efficient for large filter lengths and variable filter coefficients such as SDR channelizer. Therefore, the design methods proposed in [20], [21] are more suitable for 2-D FIR and BLMS adaptive filters.

In this paper we explore the possibility of realization of block FIR filter in transpose-from 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 architectures. The main contributions of this paper are as follows:

- Computational analysis of transpose-form configuration of FIR filter and derivation of flow-graph for transpose-form block FIR filter with reduced register complexity.
- Block formulation for transpose-from FIR filter.
- Design of transpose-form block filter for reconfigurable applications.
- A low-complexity design method using MCM scheme for the block implementation of fixed FIR filters.

The remainder of the paper is organized as follows: In Section II, computational analysis and mathematical formulation of block transpose-form FIR filter are presented. The proposed architectures for fixed and reconfigurable architectures are presented in Section III. Hardware and time complexities along with performance comparison are presented in Section IV. The paper ends with the conclusions in Section V.

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

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

(1)

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

$$Y(z) = \left[ z^{-1} \cdots (z^{-1} h(N-1) + h(N-2)) + h(N-3) \cdots + h(1) + h(0) \right] X(z)$$

(2)

![Fig. 1. Data-flow graph (DFG) of transpose-form structure for $N = 6$. (a) For output $y(n)$. (b) For output $y(n-1)$.](image1)

![Fig. 2. (a) Data-flow table (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 $y(n-1)$. The arrow shows accumulation path of the products](image2)

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 a block of two successive outputs $\{y(n), y(n-1)\}$ which 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 Fig.2. The arrows in DFT-1 and DFT-2 of Fig.2 represents the accumulation path of the products. We find that 5 values of each column of DFT-1 are same as those of DFT-2 (shown in gray colour in Fig.2). These redundant computation of DFG-1 and DFG-2 can be avoided by using non-overlapped sequence of input-blocks as shown in Fig.3. Data flow tables (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 Fig.3(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 Fig.3(a) and (b) correspond to the output $y(n)$ while the other
entries of DFT-3 and DFT-4 correspond to $y(n-1)$. The DFG of Fig.1 need to be transformed appropriately to obtain the computations according to DFT-3 and DFT-4.

**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.4. We refer it to block transpose-form type-I configuration of block FIR filter. The DFG-3 can be retimed to obtain the DFG-4 of Fig.5 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 type-I configuration. We have, therefore, used block transpose-form type-II configuration to derive the proposed structure. In the next section 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.

**C. Mathematical Formulation of the Transpose-form Block FIR Filter**

Suppose in every cycle the block FIR filter takes a block of $L$ new input samples, and processes those to produce a block of $L$ output samples. The $k$-th block of filter output $y_k$ is computed using the relation

$$ y_k = X_k \cdot h $$

where the weight-vector $h$ is defined as:

$$ h = [h(0), h(1), \ldots, h(N-1)]^T $$

The input matrix $X_k$ is defined as

$$ X_k =
\begin{bmatrix}
  x(kL) & x(kL-1) & \cdots & x(kL-N+1) \\
  x(kL-1) & x(kL-2) & \cdots & x(kL-N) \\
  \vdots & \vdots & \ddots & \vdots \\
  x(kL-L+1) & x(kL-L) & \cdots & x(kL-L-N+2)
\end{bmatrix} $$

The input matrix $X_k$ can otherwise be expressed as

$$ X_k = [x_k^0 \ x_k^1 \ \cdots \ x_k^{N-1}] $$

where $x_k^i$ is the $(i+1)$-th column of $X_k$ are defined as:

$$ x_k^i = [x(kL-i) \ x(kL-i-1) \ \cdots \ x(kL-i-L+1)]^T $$

Substituting (5) in (3), the matrix-vector product is expressed in the form of scalar-vector products (SVPs) as

$$ y_k = \sum_{i=0}^{N-1} x_k^i \cdot h(i) $$

Suppose, $N$ is a composite number and decomposed as $N = ML$, then index $i$ is expressed as $i = l + mL$, for $0 \leq l \leq L - 1$, and $0 \leq m \leq M - 1$. Substituting $i = l + mL$ in (6), we have

$$ x_k^{i+mL} = x_k^{i-m} $$
Substituting (8) in (5), we have

\[
X_k = \begin{bmatrix}
x_0^k & x_1^k & \cdots & x_{k-1}^k & x_k^0 & x_{k-1}^0 & \cdots & x_{k-M+1}^0 \\
0 & x_1^k & \cdots & x_{k-1}^k & x_k^0 & x_{k-1}^0 & \cdots & x_{k-M+1}^0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & x_1^k & x_{k-1}^k & \cdots & x_{k-M+1}^k \\
0 & 0 & \cdots & 0 & 0 & x_1^k & \cdots & x_{k-M+1}^k \\
\end{bmatrix}
\]

(9)

Substituting (9) in (3) we have

\[
y_k = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} x_{k-m}^l h(l + mL)
\]

(10)

The input-matrix \(X_k\) of (9) has an interesting feature. The data-block \(x_0^k\) is the current block while \(\{x_0^0, x_1^0, \cdots, x_{k-M+1}^0\}\) are blocks delayed by 1, 2, \(\cdots\), \((M-1)\) cycles. The overlapped blocks \(\{x_0^1, x_1^1, \cdots, x_{k-M+1}^1\}\) are, respectively, 1 clock cycle, 2 clock cycles, \(\cdots\), \((M-1)\) cycles delayed version of overlapped block \(x_1^1\). To take the advantage of this feature, the input-matrix \(X_k\) is decomposed into \(M\) small matrices \(S_k\) such that \(S_k\) contain \(L\) input-blocks \(\{x_0^k, x_1^k, \cdots, x_{k-M+1}^k\}\), and \(S_k^1\) contain input-blocks \(\{x_0^1, x_1^1, \cdots, x_{k-M+1}^1\}\). Similarly, the input-block \(\{x_0^{k-M+1}, x_1^{k-M+1}, \cdots, x_{k-M+1}^{k-M+1}\}\) constitute the matrix \(S_{k-M}^1\). The coefficient vector \(h\) is also decomposed into small weight-vectors \(c_m = \{h(mL), h(mL+1), \cdots, h(mL + L-1)\}\).

Interestingly, \(S_k^m\) is symmetric and satisfy the following identity

\[
S_k^m = S_{k-m}^0
\]

(11)

According to (11), \(S_k^m\) (for \(1 \leq m \leq M-1\)) are \(m\) clock cycle delayed with respect to \(S_k^0\). Computation of (10) can be expressed in matrix-vector product using \(S_{k-m}^0\) and \(c_m\) as

\[
y_k = \sum_{m=0}^{M-1} x_k^m
\]

(12a)

\[
r_k^m = S_{k-m}^0 \cdot c_m
\]

(12b)

The Computations of (12) may be expressed in a recurrence form

\[
Y(z) = S_0^0(z) \left[ (z^{-1})^M (z^{-1})^{M-1} c_{M-1} + c_{M-2} \right. \\
+ \left. (z^{-1})^{M-3} + \cdots + c_1 + c_0 \right]
\]

(13)

where, \(S_0^0(z)\) and \(Y(z)\) are the \(z\)-domain representation of \(S_k^0\) and \(y_k\), respectively.

The DFG-4 of block transpose-form type-II configuration (shown in Fig.5 for \(N = 6\) and \(L = 2\)) can be derived using the recurrence relation of (13). The delay operator \(\{z^{-1}\}\) of (13) represents a delay for a block of data in the transpose-form type-II structure which stores the product of \(S_k^0\) and \(c_m\). The proposed structure based on the recurrence relations of (13) is presented in the following Section.

III. PROPOSED STRUCTURES

There are several applications where the coefficients of FIR filters remain fixed, while in some other applications, like SDR channelizer which requires separate FIR filters of different specifications to extract one of the desired narrow-band channels from the wide-band RF front-end. These FIR filters need to be implemented in a reconfigurable FIR 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-from-block Block FIR Filter for Reconfigurable Applications

The proposed structure for block FIR filter is based on the recurrence relation of (13)) and shown in Fig.6 for block size \(L = 4\). It consists of one coefficient selection unit (CSU), one register unit (RU), \(M\) number of inner-product units (IPU) and one pipeline adder unit (PAU). The CSU stores coefficients of all the filters to be used for the reconfigurable application. It is implemented using \(N\) ROM LUTs such that filter coefficients of any particular channel filter is obtained in one clock cycle, where \(N\) is the filter length. The RU (shown in Fig.7) receives \(x_k\) during the \(k\)-th cycle and produces \(L\) rows of \(S_k^0\) in parallel. \(L\) rows of \(S_k^0\) 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 \(S_k^0\) form the RU. Each IPU performs matrix-vector product of \(S_k^0\) with the short-weight vector \(c_m\), and computes a block of \(L\) partial filter outputs \((r_k^m)\). Therefore, each IPU performs \(L\) inner-product computations of \(L\) rows of \(S_k^0\) with a common weight-vector \(c_m\). The structure of the \((m+1)\)-th IPU is shown in Fig.8. It consists of \(L\) number of \(K\)-point inner-product cells (IPCs). The \((l+1)\)-th IPC receives the \((l+1)\)-th row of \(S_0^0\) and the coefficient-vector \(c_m\), and computes a partial result of inner-product \(r(lK-l, l)\) for \(0 \leq l \leq L-1\). Internal structure of \((l+1)\)-th IPC for \(L = 4\) is shown in Fig.9. All the \(M\) IPUs work in parallel and produce \(M\) blocks of result \((r_k^m)\). These partial inner-products are added in the PAU (shown in Fig.10) 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 full-adder (FA) delay.

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.6 is no longer required since the structure is to be tailored for only one given filter. Similarly, IPUs are not required. The multiplications are required to be mapped to the the MCM units for a low-complexity realization. In the
Fig. 6. Proposed structure for block FIR filter

Fig. 7. Internal structure of register unit (RU) for block size $L = 4$.

Fig. 8. Structure of $(m + 1)$-th inner product unit (IPU).

The following, we show that proposed formulation for MCM-based implementation of block FIR filter which makes use of the symmetry in input matrix $S^0_k$ to perform horizontal and vertical common sub-expression elimination (CSE) [16] to minimize the number of shift-add operations in the MCM blocks.

The recurrence relation of (13) can alternately be expressed as

$$Y(z) = z^{-1}(z^{-1}r_{M-1} + \cdots + r_1 + r_0) + r_0$$  \hspace{3em} (14)

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

$$R = S^0_k \cdot C$$  \hspace{3em} (15)

where $R$ and $C$ are defined as

$$R = \begin{bmatrix} r_0^T & r_1^T & \cdots & r_{M-1}^T \end{bmatrix}$$  \hspace{3em} (16a)

$$C = \begin{bmatrix} c_0^T & c_1^T & \cdots & c_{M-1}^T \end{bmatrix}$$  \hspace{3em} (16b)

To illustrate the computation of (15) for $L = 4$ and $N = 16$, we write it as a matrix product given by (17). From (17) we can observe that the input matrix contains six input samples \{x(4k), x(4k - 1), x(4k - 2), x(4k - 3), x(4k - 4), x(4k - 5), x(4k - 6)\}, and multiplied with several constant coefficients as shown in Table I.

As shown in Table I, MCM can be applied in both horizontal and vertical direction of the coefficient matrix. The sample $x(4k - 3)$ appears in 4 rows or 4 columns of the input matrix of (17) whereas $x(4k)$ appears in only one row or one column. Therefore, all the 4 rows of coefficient matrix are involved in the MCM for the $x(4k - 3)$ whereas only the first row of coefficients are involved in the MCM for $x(4k)$. For larger values of $N$ or the smaller block-sizes, the row size of the coefficient matrix are larger which results in larger MCM size across all the samples which results into larger saving in computational complexity.

The proposed MCM-based structure for FIR filters for block-size $L = 4$ is shown in Fig.11 for the purpose of illustration. The MCM-based structure (shown in Fig.11) involves
to be implemented by the proposed reconfigurable structure. We have excluded complexity of CSU in the performance comparison since it is common in all the reconfigurable FIR structures. Each IPU is comprised of $L$ IP cells, where each IP cell involves $L$ multipliers and $(L - 1)$ adders. The RU involves $(L - 1)$ registers of $B$-bit width. The PAU involves $L(M - 1)$ adders and the same number of registers, where each register has a width of $(B + B')$, $B$ and $B'$ respectively, being the bit-width of input sample and filter-coefficients. Therefore, the proposed structure involves $LN$ multipliers, $L(N - 1)$ adders and $[B(N - 1) + B'(N - L)]$ FFs; and processes $L$ samples in every cycle where the duration of cycle period $T = [T_M + T_A + T_{FA}\log_2 L - 1)]$. We have estimated the hardware and time complexity of the block transpose-form type-I structure from DFG-3 of Fig.4 for comparison purpose. We do not find a multiplier-based direct-form block FIR structure on RFIR in the literature. However, direct-form multiplier-based block FIR structure can be derived from the block formulation of [20]. We have derived the direct-form block FIR structure using equation (4) of [20], and estimated its hardware and time complexities for comparison purpose.

B. Performance Comparison

The hardware and time complexities of proposed structure, the block transpose-form type-I structure and the extracted direct-form structure of [20] along with those of the existing reconfigurable FIR filter structures of [18] and [17] are listed in Table II for comparison. We have assumed fixed word-length $(B + B')$ for the adder-tree in case of direct-form structure as well as the pipeline-adder in case of transpose-form structure. As shown in Table II, the proposed block transpose form structures have the same number of multipliers and adders, and they differ only by the register complexity and cycle period. The direct-form structure of [20] and the proposed structures involve the same number of multipliers and adders, but the proposed one involves $\{\log_2 M - 1\}T_{FA}$ less cycle period, where $M = N/L$, at a marginal cost of $B'(N - L)$ FFs. The register complexity of the proposed structure is independent of block-size as in the case of direct-form structure. However, the cycle period of the proposed block transpose-form type-II structure depends on the input block-size whereas in case of existing direct-form block FIR structure of [20] it depends on the filter-length. Since, filter-length is usually higher than the block-length, the cycle period of the existing direct-form structure increase for large order filters. To compare with the DA-based structure of [18] and the proposed structure, we find that the proposed structure involves $(LN)$ multipliers in place of $(3NBB'/2)$ MUXes (bit-level), nearly $(2L/B)$ times more adders and $B'N$ more FFs, but offers nearly $L$ times higher throughput. Similarly, compared with the CSM-based structure of [17], the proposed structure involve $(LN)$ multipliers in place of $(\approx 7NBB'/3)$ MUXes (bit-level), $(\approx 3L/B')$ times more adders, $B'(L - 1)$ less FFs, and offers $L$ times higher throughput. The proposed block transpose-form type-II structure involves nearly $L$ times less FFs than those of block transpose-form type-I structure and both the structures have the same cycle period. Due to

![Fig. 11. Proposed MCM-based structure for fixed FIR filter of block-size $L = 4$ and filter length $N = 16$.](image)

### Table I

<table>
<thead>
<tr>
<th>Input sample</th>
<th>Coefficient Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x(4k)$</td>
<td>${h(0), h(4), h(8), h(12)}$</td>
</tr>
<tr>
<td>$x(4k - 1)$</td>
<td>${h(0), h(4), h(8), h(12)}$</td>
</tr>
<tr>
<td></td>
<td>${h(1), h(5), h(9), h(13)}$</td>
</tr>
<tr>
<td>$x(4k - 2)$</td>
<td>${h(0), h(4), h(8), h(12)}$</td>
</tr>
<tr>
<td></td>
<td>${h(1), h(5), h(9), h(13)}$</td>
</tr>
<tr>
<td></td>
<td>${h(2), h(6), h(10), h(14)}$</td>
</tr>
<tr>
<td>$x(4k - 3)$</td>
<td>${h(0), h(4), h(8), h(12)}$</td>
</tr>
<tr>
<td></td>
<td>${h(1), h(5), h(9), h(13)}$</td>
</tr>
<tr>
<td></td>
<td>${h(2), h(6), h(10), h(14)}$</td>
</tr>
<tr>
<td></td>
<td>${h(3), h(7), h(11), h(15)}$</td>
</tr>
<tr>
<td>$x(4k - 4)$</td>
<td>${h(1), h(5), h(9), h(13)}$</td>
</tr>
<tr>
<td></td>
<td>${h(2), h(6), h(10), h(14)}$</td>
</tr>
<tr>
<td></td>
<td>${h(3), h(7), h(11), h(15)}$</td>
</tr>
<tr>
<td>$x(4k - 5)$</td>
<td>${h(2), h(6), h(10), h(14)}$</td>
</tr>
<tr>
<td></td>
<td>${h(3), h(7), h(11), h(15)}$</td>
</tr>
<tr>
<td>$x(4k - 6)$</td>
<td>${h(3), h(7), h(11), h(15)}$</td>
</tr>
</tbody>
</table>
TABLE II
GENERAL COMPARISON OF HARDWARE AND TIME COMPLEXITIES

<table>
<thead>
<tr>
<th>Structures</th>
<th>Flip-Flop</th>
<th>Adder</th>
<th>Multiplier</th>
<th>MUX 2:1 (bit-level)</th>
<th>CP</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mahesh et al [17] (CSM)</td>
<td>$(B + B')(N - 1)$</td>
<td>$N(\alpha + 1) + 2$</td>
<td>0</td>
<td>$[7\alpha(B + 2) + B + B']N$</td>
<td>$4T_{MUX} + 4T_A + 3T_{FA}$</td>
<td>$1/CP$</td>
</tr>
<tr>
<td>Park et al [18] (One Level Pipeline)</td>
<td>$B(N + B' + \phi + 1)$</td>
<td>$[N(B + 1)/2] - 1$</td>
<td>0</td>
<td>$3NNB'/2$</td>
<td>$2T_{MUX} + 2T_A + 3T_{FA}$</td>
<td>$1/CP$</td>
</tr>
<tr>
<td>Direct-form structure of [20]</td>
<td>$B(N - 1)$</td>
<td>$L(N - 1)$</td>
<td>$NL$</td>
<td>0</td>
<td>$T_M + T_A + (\log_2 N - 1)T_{FA}$</td>
<td>$L/CP$</td>
</tr>
<tr>
<td>Transpose-form Type-I</td>
<td>$B[L(N - L + 1) - 1] + B'L(N - L)$</td>
<td>$L(N - 1)$</td>
<td>$NL$</td>
<td>0</td>
<td>$T_M + T_A + (\log_2 L)T_{FA}$</td>
<td>$L/CP$</td>
</tr>
<tr>
<td>Transpose-form Type-II</td>
<td>$B(N - 1) + B'(N - L)$</td>
<td>$L(N - 1)$</td>
<td>$NL$</td>
<td>0</td>
<td>$T_M + T_A + (\log_2 L)T_{FA}$</td>
<td>$L/CP$</td>
</tr>
</tbody>
</table>

$\alpha = \lceil B' / 3 \rceil$, $\beta = \lceil \log_2 \alpha - 1 \rceil$, $\phi = \log_2 (N/2) - 1$.

TABLE III
THEORETICALLY ESTIMATED HARDWARE AND TIME COMPLEXITIES OF PROPOSED AND EXISTING STRUCTURES FOR $B = 8$ AND $B' = 16$

<table>
<thead>
<tr>
<th>Structures</th>
<th>Filter-length</th>
<th>FF</th>
<th>Adder</th>
<th>Multipliers</th>
<th>MUX 2:1 bit-level</th>
<th>Cycle period (T)</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mahesh et al [17] (CSM)</td>
<td>16</td>
<td>360</td>
<td>114</td>
<td>0</td>
<td>7104</td>
<td>$4T_{MUX} + 4T_A + 3T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>744</td>
<td>226</td>
<td>0</td>
<td>14208</td>
<td>$4T_{MUX} + 4T_A + 3T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1512</td>
<td>450</td>
<td>0</td>
<td>28416</td>
<td>$4T_{MUX} + 4T_A + 3T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>3048</td>
<td>898</td>
<td>0</td>
<td>56832</td>
<td>$4T_{MUX} + 4T_A + 3T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td>Park et al [18] (One Level Pipeline)</td>
<td>16</td>
<td>296</td>
<td>71</td>
<td>0</td>
<td>3456</td>
<td>$2T_{MUX} + 2T_A + 2T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>432</td>
<td>143</td>
<td>0</td>
<td>6912</td>
<td>$2T_{MUX} + 2T_A + 2T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>696</td>
<td>287</td>
<td>0</td>
<td>13824</td>
<td>$2T_{MUX} + 2T_A + 2T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>1216</td>
<td>575</td>
<td>0</td>
<td>27648</td>
<td>$2T_{MUX} + 2T_A + 2T_{FA}$</td>
<td>$1/T$</td>
</tr>
<tr>
<td>Direct-form structure of [20]</td>
<td>$L = 4$</td>
<td>16</td>
<td>120</td>
<td>60</td>
<td>64</td>
<td>$T_M + T_A + 3T_{FA}$</td>
<td>$4/T$</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>248</td>
<td>124</td>
<td>128</td>
<td>0</td>
<td>$T_M + T_A + 4T_{FA}$</td>
<td>$4/T$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>504</td>
<td>252</td>
<td>256</td>
<td>0</td>
<td>$T_M + T_A + 5T_{FA}$</td>
<td>$4/T$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>1016</td>
<td>508</td>
<td>512</td>
<td>0</td>
<td>$T_M + T_A + 6T_{FA}$</td>
<td>$4/T$</td>
</tr>
<tr>
<td>Direct-form structure of [20]</td>
<td>$L = 8$</td>
<td>16</td>
<td>120</td>
<td>120</td>
<td>128</td>
<td>0</td>
<td>$T_M + T_A + 3T_{FA}$</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>248</td>
<td>248</td>
<td>256</td>
<td>0</td>
<td>$T_M + T_A + 4T_{FA}$</td>
<td>$8/T$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>504</td>
<td>504</td>
<td>512</td>
<td>0</td>
<td>$T_M + T_A + 5T_{FA}$</td>
<td>$8/T$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>1016</td>
<td>1016</td>
<td>1024</td>
<td>0</td>
<td>$T_M + T_A + 6T_{FA}$</td>
<td>$8/T$</td>
</tr>
<tr>
<td>Proposed structure</td>
<td>$L = 4$</td>
<td>16</td>
<td>312</td>
<td>60</td>
<td>64</td>
<td>0</td>
<td>$T_M + T_A + 2T_{FA}$</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>696</td>
<td>124</td>
<td>128</td>
<td>0</td>
<td>$T_M + T_A + 2T_{FA}$</td>
<td>$4/T$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1464</td>
<td>252</td>
<td>256</td>
<td>0</td>
<td>$T_M + T_A + 2T_{FA}$</td>
<td>$4/T$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>3000</td>
<td>508</td>
<td>512</td>
<td>0</td>
<td>$T_M + T_A + 2T_{FA}$</td>
<td>$4/T$</td>
</tr>
<tr>
<td>Proposed structure</td>
<td>$L = 8$</td>
<td>16</td>
<td>248</td>
<td>120</td>
<td>128</td>
<td>0</td>
<td>$T_M + T_A + 3T_{FA}$</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>632</td>
<td>248</td>
<td>256</td>
<td>0</td>
<td>$T_M + T_A + 3T_{FA}$</td>
<td>$8/T$</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1400</td>
<td>504</td>
<td>512</td>
<td>0</td>
<td>$T_M + T_A + 3T_{FA}$</td>
<td>$8/T$</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>2096</td>
<td>1016</td>
<td>1024</td>
<td>0</td>
<td>$T_M + T_A + 3T_{FA}$</td>
<td>$8/T$</td>
</tr>
</tbody>
</table>

less FF complexity, proposed type-II structure is more efficient than the type-I structure. In-spite of more FFs, the proposed structure may have less area-delay product (ADP) and less energy per sample (EPS) than the existing direct-form structure due to its small cycle period.

We have estimated hardware and time complexities of proposed structures for block sizes $L = 4, 8$, and filter-lengths $N = 32$ and 64. Also we have estimated the hardware and the time complexities of the direct-form structure extracted from [20] for the same block-size and for the filter-lengths, and those of [18] and [17] for the same filter-lengths. We have considered $B = 8$ (wordlength of input sample), $B' = 16$ (wordlength of filter coefficient) and 24-bit wordlength for the intermediate and output signals for all the designs. The estimated values are listed in Table III for comparison. We can find from Table III that, the multiplier and adder complexities of the proposed structure increases proportionately with block-size and filter-length as in the case of direct-form.
C. Synthesis Results

We have coded the proposed structure in VHDL for filter lengths 16, 32 and 64 and block-sizes 4 and 8. Also we have coded the direct-form block FIR structure extracted from [20] for the same filter-lengths and the same block-sizes, and the structures of [18] and [17] for the same filter-lengths. We have considered \( B = 8, B' = 16 \), and 24-bit wordlength for the intermediate and the output signals of all the designs. All the designs are synthesized using Synopsys Design Compiler TMSC 65nm CMOS library. The area, the minimum clock period (MCP), and power estimates obtained from the synthesis reports generated by the Design Compiler are listed in Table IV for comparison. As shown in Table IV, the proposed transpose-form type-II structure involves more area and consumes more power than the existing direct-form structure [20] due to extra FFs. But, it has less MCP (higher sampling frequency\(^1\), block-size=1 in case of [17], [18]) than the corresponding direct-form structure of [20] due to shorter critical path. We have estimated the increase in area \((\Delta A)\) and reduction in MCP \((\Delta T)\) of proposed transpose-form type-II structure over the direct-form structure of [20] for different block-sizes and different filter-lengths. Graphs are plotted using these estimated values and shown in Fig.12 and Fig.13. Note that the ADP varies directly with \((\Delta A)\) whereas it varies inversely with \((\Delta T)\). As shown in Figs.12 and 13, the intersection point of two curves \((\Delta A, \Delta T)\) gives a filter-length \( N_0 \) where the direct-form structure of [20] and transpose-form type-II structures have nearly the same ADP. For \( N < N_0 \), \((\Delta A)\) is higher than \((\Delta T)\) and transpose-form type-II structure has higher ADP than that of direct-form structure of [20]. Similarly, for \( N > N_0 \) the \((\Delta T)\) is higher than \((\Delta A)\) and the transpose-form type-II structure has less ADP than the direct-form structure of [20]. The \( N_0 \) shift marginally towards higher value for higher block-sizes due to increase in MCP of the transpose-form type-II structure.

We have estimated ADP\(^2\) and energy per sample (EPS\(^3\)) of proposed transpose-form type-II structure and direct-form structure of [20]. We have also estimated the reduction of ADP (RADP) and reduction of EPS (REPS) of the proposed transpose-form type-II structure over the direct-form structure of [20]. The estimated RADP and REPS values are shown in the bar-chart of Fig.13 for comparison. As shown in Fig.13, the proposed transpose-form type-II structure has higher ADP and EPS (negative RADP and REPS) than the existing direct-form structure of [20] for small filter-lengths while it has less ADP and less EPS (positive RADP and REPS) for filter-length

\(^1\)sampling frequency=block-size/minimum clock period

\(^2\)ADP=area/sampling frequency

\(^3\)EPS=power/sampling frequency
The proposed block transpose-form type-II structure offers higher saving of ADP and EPS saving than the existing direct-form structure [20] for higher filter-lengths. Based on the above observation, we suggest a selection rule (given in Table V) for the direct-form (DF) and transpose-form type-II (TF-II) configuration to derive area-delay and energy efficient block FIR filter structure of different block-sizes and filter-lengths.

We have estimated ADP and EPS of the existing structures of [17] and [18] and these estimated values along with those of the proposed structures and the direct-form structure of [20] are shown in bar-charts of Fig.13 for comparison. As shown in Fig.13, the proposed block transpose-form type-II structures has significantly less ADP and EPS than the existing multiplier-less structures of [17] and [18]. Compared with the existing direct-form structure of [20], the transpose-form type-II structure offers higher ADP and EPS saving for medium and large filter lengths while the existing direct-form structure has higher ADP and EPS saving than the proposed block transpose-form type-II structure for short-length filters. Compared with the structure of [18] which is the best amongst the existing designs, the block proposed transpose-form type-II structure for block-size 4 and filter-length 64 involve 42% less ADP and 40% less EPS respectively.

### V. Conclusion

Transpose-form structures are inherently pipelined and supports MCM which results significant saving in computation and increase in higher sampling rate. However, transpose-form configuration does not directly support the block processing. In this paper, we have explored the possibility of realization of block FIR filter in transpose-from configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. We have made computational analysis of transpose-form configuration of FIR filter and derived a flow-graph for transpose-from block FIR filter with optimized register complexity. A generalized block formulation is also presented for transpose-from block FIR filter. Based on that we have derived transpose-form block filter for reconfigurable applications. We have presented the scheme to identify the MCM blocks explored the horizontal and vertical sub-expression elimination for the implementation of the proposed block FIR structure for fixed coefficients to reduce the computational complexity. A low-complexity design method using MCM scheme is also presented for the block implementation of fixed FIR filters. Performance comparison
shows that the proposed structure involve 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 FIR structure has less ADP and less EPS than the proposed structure. ASIC 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 structure of [18] for reconfigurable applications. For the same filter length and block size, the proposed structure involves 13% less ADP and 12.8% less EPS than that of the existing direct-from block FIR structure of [20]. Based on these findings, selection of direct-form and transpose-form configuration based on the filter lengths and block-length is suggested for obtaining area-delay and energy efficient structures for block FIR filters.

REFERENCES


Basant K Mohanty (M’06, SM’11) received M.Sc degree in Physics from Sambalpur University, India, in 1989 and received Ph.D degree in the field of VLSI Digital Signal Processing from Berhampur University, Orissa in 2000. In 2001 he joined as Lecturer in IEEE Department, BITS Pilani, Rajasthan. Then he joined as an Assistant Professor in the Department of ECE, Mody Institute of Education Research (Deemed University), Rajasthan. In 2003 he joined Jaypee University of Engineering and Technology, Guna, Madhya Pradesh, where he becomes Associate Professor in 2005 and full Professor in 2007. His research interest includes design and implementation of low-power and high-performance systems for adaptive filters, image and video-processing applications, secured communication and reconfigurable architectures. He has published nearly 50 technical papers. Dr. Mohanty is serving as Associate Editor for the Journal of Circuits, Systems, and Signal Processing. He is a life time member of the Institution of Electronics and Telecommunication Engineering, New Delhi, India, and he was the recipient of the Rashtriyaa Gaurav Award conferred by India International friendship Society, New Delhi for the year 2012.

Pramod Kumar Meher (SM’93) received the B.Sc. (Honours) and M.Sc. degree in physics, and the Ph.D. degree in science from Sambalpur University, India, in 1976, 1978, and 1996, respectively. Currently, he is a Senior Research Scientist with Nanyang Technological University, Singapore. Previously, he was a Professor of Computer Applications with Utkal University, India, from 1997 to 2002, and a Reader in electronics with Berhampur University, India, from 1993 to 1997. His research interest includes design of dedicated and reconfigurable architectures for computation-intensive algorithms pertaining to signal, image and video processing, communication, bio-informatics and intelligent computing. He has contributed more than 200 technical papers to various reputed journals and conference proceedings.

Dr. Meher has served as a speaker for the Distinguished Lecturer Program (DLP) of IEEE Circuits Systems Society during 2011 and 2012 and Associate Editor of the IEEE Transactions on Circuits and Systems-II: Express Briefs during 2008 to 2011, and Associate Editor for the IEEE Transactions on Circuits and Systems-I: Regular Papers during 2012-2013. Currently, he is serving as Associate Editor for the IEEE Transactions on Very Large Scale Integration (VLSI) SYSTEMS, Journal of Circuits, Systems, and Signal Processing (CSSP), and Integration, the VLSI Journal. Dr. Meher is a Fellow of the Institution of Electronics and Telecommunication Engineers, India. He was the recipient of the Samanta Chandrasekhar Award for excellence in research in engineering and technology for 1999.