



International Journal for Innovative Engineering and Management Research

A Peer Reviewed Open Access International Journal

www.ijiemr.org

COPY RIGHT



ELSEVIER
SSRN

2021IJIEMR. Personal use of this material is permitted. Permission from IJIEMR must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. No Reprint should be done to this paper, all copy right is authenticated to Paper Authors

IJIEMR Transactions, online available on 12th Jun 2021. Link

[:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-06](http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-06)

DOI: 10.48047/IJIEMR/V10/I06/02

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

Volume 10, Issue 06, Pages: 8-16

Paper Authors

CHINTHA ANUSHA, Dr T Raghavendra Vishnu



USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per **UGC Guidelines** We Are Providing A Electronic Bar Code

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

CHINTHA ANUSHA

M-tech student Scholar Department of Electronics and Communication Engineering,
Priyadarshini Institute Of Technology & Science For Women, chintalapudi Village, Duggirala Mandal,
Near Tenali, Guntur, Andhra Pradesh, India.

Dr T Raghavendra Vishnu M.tech, Ph.D.,

Associate Professor & HOD Department of Electronics and Communication Engineering,
Priyadarshini Institute Of Technology & Science For Women, chintalapudi Village, Duggirala Mandal,
Near Tenali, Guntur, Andhra Pradesh, 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 directform 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 Area delay product (ADP) and less energy per sample (EPS) 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 and less EPS than the proposed structure. Applicationspecific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involves 42% less ADP and 40% less EPS 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 and 12.8% less EPS than that of the existing direct-form blocks FIR structure.

Key Words: Area delay product (ADP), Finite-impulse response (FIR), multiple constant multiplications (MCM), energy per sample (EPS)

(I) Introduction

In this paper, we explore the possibility of realization of block FIR filter in transpose form configuration for areadelay 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 area delay product (ADP) and less energy per sample (EPS) 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 and less EPS than the proposed structure. Applicationspecific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involves 42% less ADP and 40% less EPS 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 and 12.8% less EPS than that of the existing direct-form block FIR structure.

(II) LITERATURE SURVEY

Finite-Impulse Response (FIR) filters play a essential role in lots of sign processing packages in conversation structures. A big type of obligations alongside spectral shaping, matched filtering, interference cancellation, channel equalization, and so forth. May be finished with these filters. As a end result, diverse architectures and implementation strategies had been proposed to improve the overall typical overall performance of filters in terms of pace and complexity. Currently, explosive proliferation in pressured and Wi-Fi conversation requirements renders traditional FIR architectures a excellent deal much less suitable for future conversation desires. However, software program radio [1]–[3] has received lots interest from the researchers international due to a strong call for reconfigurable conversation structures able to multi-substantial operations. In mild of this style, programmability and re-configurability need be taken underneath consideration in clean out form format. It’s a long way widely diagnosed that the Canonical Signed Digit (CSD) instance may be used to lessen the complexity of FIR digital easy out implementation [4]–[6]. Versatile channels, specifically, are unavoidable in numerous vital applications in interchanges, image dealing with, computer vision, data securing and control [6]-[10]. With any flexible channel, there may be necessity for a programmable channel, that is an vital reason for the expanding predominance of advanced as opposed to easy framework utilization. in packages, for example, multi-price devastation [7], discrete cosine alternate

[8], [9], channelization [10], [11], excessive proficiency video coding (HEVC) [12], huge transmission capacity photonic filter out[13] and excessive-rate correspondence [14], the channel coefficients should be run-time reconfigurable by using the blunder complaint flag or versatile to fluctuating sifting information constantly.

(III) 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

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

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

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

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

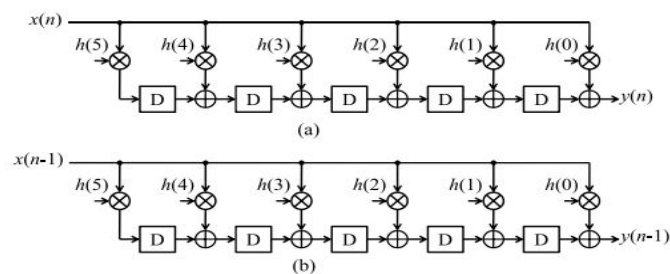


Fig. 1. DFG of transpose form structure for N = 6. (a) DFG-1 for output y(n). (b) DFG-2 for output y(n - 1).

| ccs | 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)

| ccs | 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)

Fig. 2. (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.

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 dataflow tables (DFT-1 and DFT-2) of Fig. 2. The arrows in DFT-1 and DFT-2 of Fig. 2 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 Fig. 3. DFT-3 and DFT-4 of DFG-1 and DFG-2 for nonoverlapping 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 $y(n)$, 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)

Fig. 3. DFT of DFG-1 and DFG-2 for three nonoverlapped 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 $y(n-1)$.

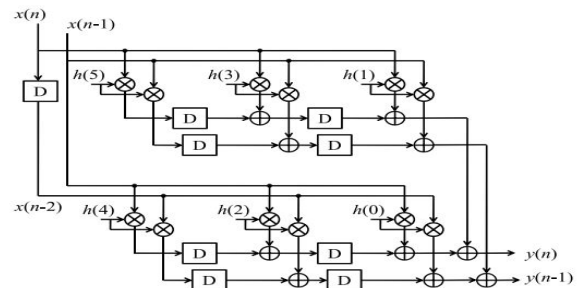


Fig. 4. Merged DFG (DFG-3: transpose form type-I configuration for block FIR structure).

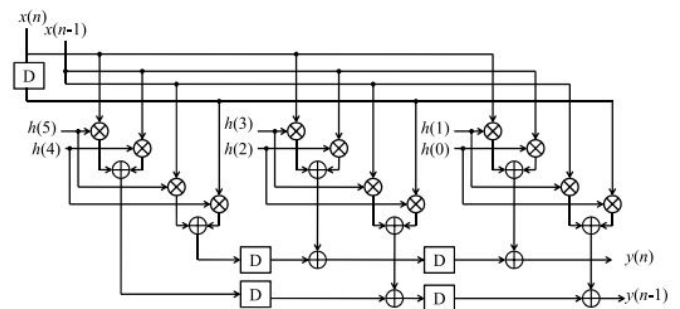


Fig. 5. DFG-4 (retimed DFG-3) transpose form type-II configuration for block FIR structure.

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 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.

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

$$y_k = X_k \cdot h \quad (3)$$

where the weight vector h is defined as

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

The input matrix X_k is defined as

$$X_k = [x_k^0 \quad x_k^1 \quad \dots \quad x_k^L \quad \dots \quad x_k^{N-1}] \quad (4)$$

Substituting (4) in (3), the matrix-vector product is expressed in the form of scalar-vector product as

$$x_k^i = [x(kL-i) \ x(kL-i-1) \ \dots \ x(kL-i-L+1)]^T. \quad (5)$$

$$y_k = \sum_{i=0}^{N-1} x_k^i \cdot h(i). \quad (6)$$

Suppose N is a composite number and decomposed as $N = M L$, 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 (5), we have

$$x_k^{l+mL} = x_{k-m}^l. \quad (7)$$

Substituting (7) in (4), we have

$$X_k = [x_k^0 \quad x_k^1 \quad \dots \quad x_k^{L-1} \quad x_{k-1}^0 \quad x_{k-1}^1 \quad \dots \quad x_{k-1}^{L-1} \quad \dots \quad x_{k-M+1}^0 \quad x_{k-M+1}^1 \quad \dots \quad x_{k-M+1}^{L-1}]. \quad (8)$$

$$y_k = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} x_{k-m}^l \cdot h(l + mL). \quad (9)$$

The input matrix X_k of (8) has an interesting feature. The data block x_0^k is the current block, while $\{x_{k-1}^0, x_{k-2}^0, \dots, x_{k-M+1}^0\}$ are blocks delayed by 1, 2, \dots , $(M-1)$ cycles. The overlapped blocks $\{x_{k-1}^1, x_{k-2}^1, \dots, x_{k-1-L+1}^1\}$ are, respectively, 1 clock cycle, 2 clock cycles, \dots , $(M-1)$ cycles delayed version of overlapped block x_{k-1}^1 . To take the advantage of this feature, the input-matrix X_k is decomposed into M small matrices S_l^k , such that S_0^k contains L inputblocks $\{x_{k-1}^0, x_{k-1}^1, \dots, x_{k-1}^{L-1}\}$, and S_1^k contains input blocks $\{x_{k-1}^0, x_{k-1}^1, \dots, x_{k-1-L+1}^1\}$. Similarly, the input block $\{x_{k-M+1}^0, x_{k-M+1}^1, \dots, x_{k-M+1-L+1}^1\}$ constitute the matrix S_{M-1}^k .

The coefficient vector h is also decomposed into small weight vectors $c_m = \{h(mL), h(mL+1), \dots, h(mL+L-1)\}$. Interestingly, S_m^k is symmetric and satisfy the following identity:

$$S_k^m = S_{k-m}^0. \quad (10)$$

According to (10), S_m^k (for $1 \leq m \leq M-1$) are m clock cycle delayed with respect to S_0^k . Computation of (9) can be expressed in matrix-vector product using S_0^{k-m} and c_m as

$$y_k = \sum_{m=0}^{M-1} r_k^m$$

$$r_k^m = S_{k-m}^0 \cdot c_m \quad (11)$$

The computations of (11) may be expressed in a recurrence form

$$Y(z) = S^0(z)[(z^{-1}(\dots(z^{-1}(z^{-1}c_{M-1} + c_{M-2}) + c_{M-3}) + \dots) + c_1) + c_0] \quad (12)$$

where $S_0(z)$ and $Y(z)$ are the z-domain representation of S_0k 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 (12). The delay operator $\{z^{-1}\}$ of (12) represents a delay for a block of data in the transpose form type-II structure that stores the product of S_0k and c_m .

(IV) 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 multistandard 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. 2.6 for the block size $L = 4$. It consists of one coefficient selection unit (CSU), one register unit (RU), M number of innerproduct units (IPUs), 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 are obtained in one clock cycle, where N is the filter length. The RU [shown in Fig. 7(a)] receives x_k during the kth cycle and produces L rows of S_0k in parallel. L rows of S_0k are transmitted to M IPUs of the proposed structure. The M IPUs also receive M short-weight vectors from the CSU

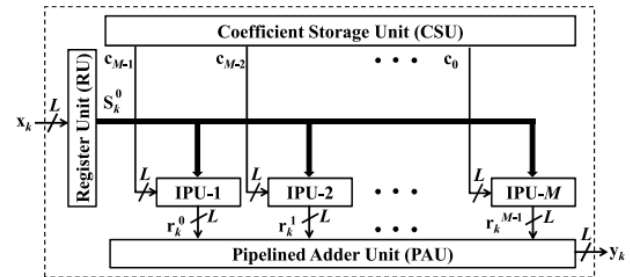


Fig. 6. Proposed structure for block FIR filter

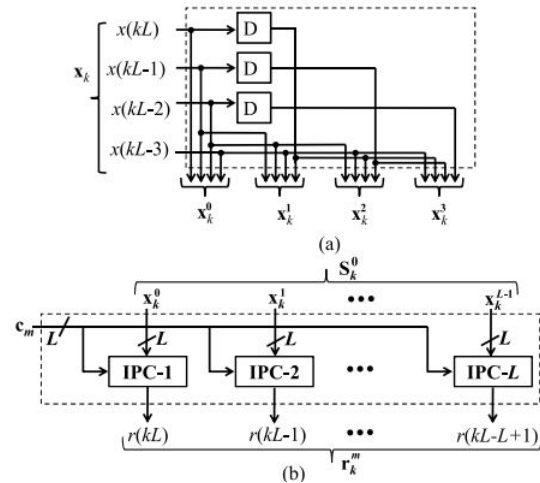


Fig. 7. (a) Internal structure of RU for block size $L = 4$. (b) Structure of $(m + 1)$ th IPU.

Such that during the kth cycle, the $(m + 1)$ th IPU receives the weight vector c_{M-m-1} from the CSU and L rows of S_0k from the RU. Each IPU performs matrix-vector product of S_0k with the short-weight vector c_m , and computes a block of L partial filter outputs (r_{km}). Therefore, each IPU performs L inner-product computations of L rows of S_0k with a common weight vector c_m . The structure of the $(m + 1)$ th IPU is shown in Fig. 7(b). It consists of L number of L-point inner-product cells (IPCs). The $(l + 1)$ th IPC receives the $(l + 1)$ th row of S_0k and the

coefficient vector c_m , and computes a partial result of inner product $r(kL - 1)$, for $0 \leq l \leq L - 1$. Internal structure of $(l + 1)$ th IPC for $L = 4$ is shown in Fig. 8(a). All the M IPU's work in parallel and produce M blocks of result (r_{km}) .

These partial inner products are added in the PAU [shown in Fig. 8(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 $T = TM + TA + TFA \log_2 L$, TM is one multiplier delay, TA is one adder delay, and TFA is one full-adder 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

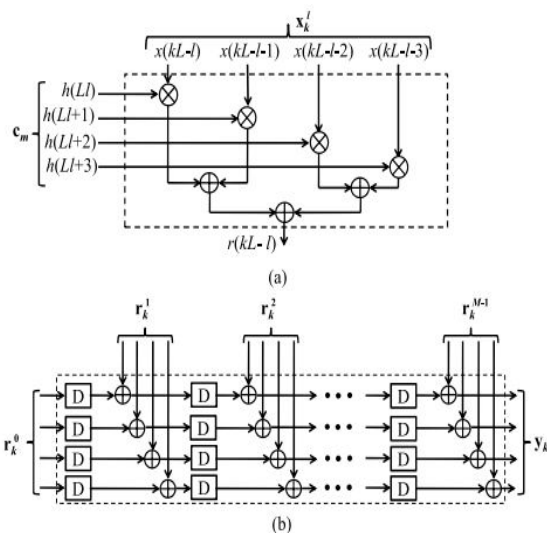


Fig. 8. (a) Internal structure of $(l + 1)$ th IPC for $L = 4$.
(b) Structure of PAU for block size $L = 4$.

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 subexpression 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

$$R = S_k^0 \cdot C \quad (14)$$

where R and C are defined as

$$R = [r_0^T \quad r_1^T \quad \cdots \quad r_{M-1}^T] \\ C = [c_0^T \quad c_1^T \quad \cdots \quad c_{M-1}^T]. \quad (15)$$

To illustrate the computation of (14) for $L = 4$ and $N = 16$, we write it as a matrix product given by (16). From (16), 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 four rows or four columns of the following input matrix:

TABLE I
MCM IN TRANSPOSE FORM BLOCK FIR FILTER OF LENGTH 16 AND BLOCK SIZE 4

| Input sample | Coefficient Group |
|--------------|--|
| $x(4k)$ | $\{h(0), h(4), h(8), h(12)\}$ |
| $x(4k - 1)$ | $\{h(0), h(4), h(8), h(12)\}$ $\{h(1), h(5), h(9), h(13)\}$ |
| $x(4k - 2)$ | $\{h(0), h(4), h(8), h(12)\}$ $\{h(1), h(5), h(9), h(13)\}$ $\{h(2), h(6), h(10), h(14)\}$ |
| $x(4k - 3)$ | $\{h(0), h(4), h(8), h(12)\}$ $\{h(1), h(5), h(9), h(13)\}$ $\{h(2), h(6), h(10), h(14)\}$ $\{h(3), h(7), h(11), h(15)\}$ |
| $x(4k - 4)$ | $\{h(1), h(5), h(9), h(13)\}$ $\{h(2), h(6), h(10), h(14)\}$ $\{h(3), h(7), h(11), h(15)\}$ |
| $x(4k - 5)$ | $\{h(2), h(6), h(10), h(14)\}$ $\{h(3), h(7), h(11), h(15)\}$ |
| $x(4k - 6)$ | $\{h(3), h(7), h(11), h(15)\}$ |

$$R = \begin{bmatrix} x(4k) & x(4k - 1) & x(4k - 2) & x(4k - 3) \\ x(4k - 1) & x(4k - 2) & x(4k - 3) & x(4k - 4) \\ x(4k - 2) & x(4k - 3) & x(4k - 4) & x(4k - 5) \\ x(4k - 3) & x(4k - 4) & x(4k - 5) & x(4k - 6) \end{bmatrix} \times \begin{bmatrix} h(0) & h(4) & h(8) & h(12) \\ h(1) & h(5) & h(9) & h(13) \\ h(2) & h(6) & h(10) & h(14) \\ h(3) & h(7) & h(11) & h(15) \end{bmatrix} \quad (16)$$

whereas $x(4k)$ appears in only one row or one column. Therefore, all the four 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 is larger that 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. 9 for the purpose of illustration. The MCM-based structure (shown in Fig. 9) involves six MCM blocks corresponding to six input samples. Each MCM block produces the necessary product terms as listed in Table I. The subexpressions 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).

(V) SIMULATION RESULTS

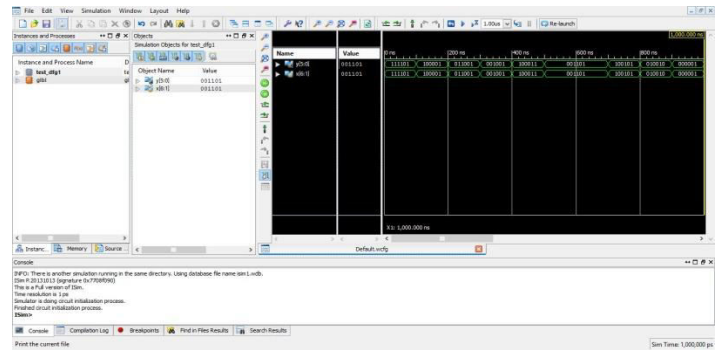


Fig 9: Simulation result of the DFG

Fig 9 shows the simulation result of the DFG architecture in which input is given through the x data signals and the output is observed in the y data signal.

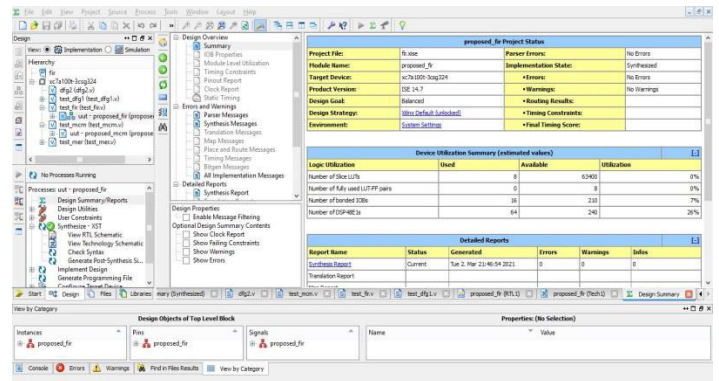


Fig 10: Summary report of the proposed FIR filter

Fig 10 presents the summary report of the proposed FIR filter architecture in which it shows the count of the number of slice registers, LUT's and number of DSP Slices are utilized in the respective FPGA family.

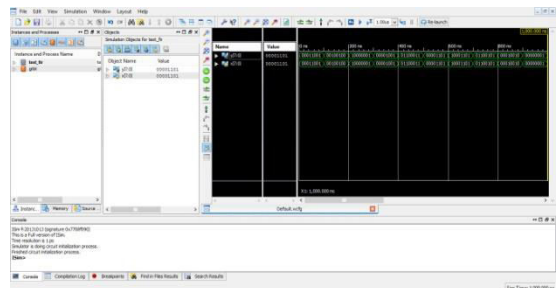


Fig 11: Simulation result of the proposed FIR filter

Fig 11 shows the simulation result of the proposed FIR filter architecture in which input is given through the x data signals and the output is observed in the y data signal.

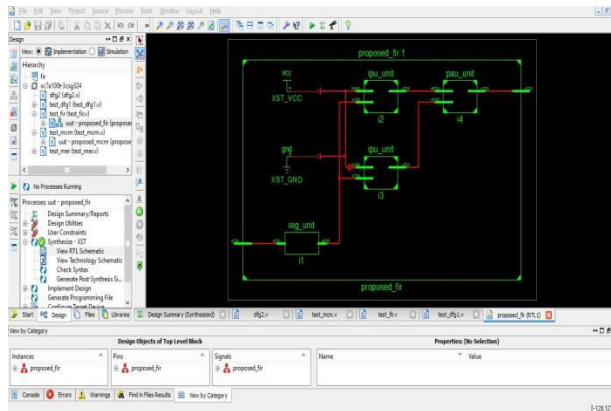


Fig 12: RTL schematic of the proposed FIR filter

Fig 12 shows the RTL schematic of the proposed FIR filter architecture in which it shows the internal structure of the design.

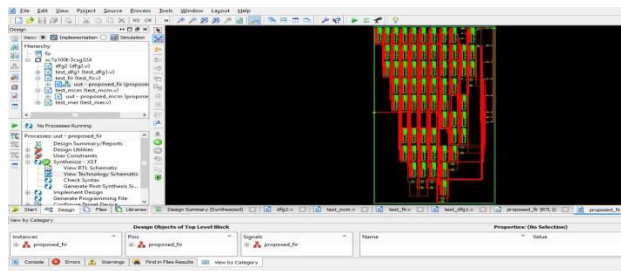


Fig 13: Technology schematic of the proposed FIR filter

Fig 13 shows the technology schematic of the proposed FIR filter architecture. In this the design is implemented with the help of LUT's and DSP slices only.

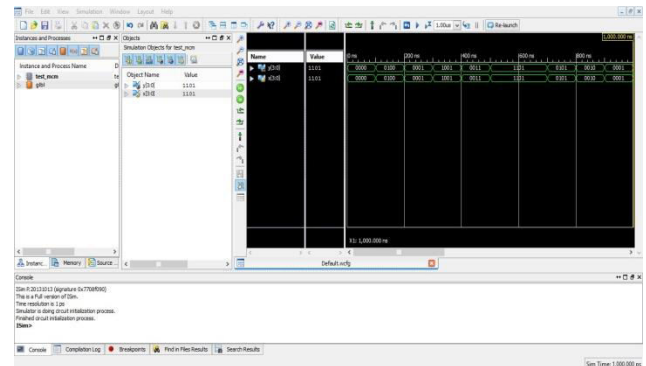


Fig 14: Simulation result of the MCM

Fig 14 shows the simulation result of the MCM architecture in which input is given through the x data signals and the output is observed in the y data signal.

(VI) Conclusion

In this paper, we have explored the possibility of realization of block FIR filters in transpose form configuration for areadelay 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. The proposed FIR filter structure achieves better performance compared with the previous architectures.

Bibliography

- [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.
- [13] P. K. Meher, "New approach to look-up-table design and memorybased realization of FIR digital filter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 3, pp. 592–603, Mar. 2010.
- [14] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York, NY, USA: Wiley, 1999.
- [15] B. K. Mohanty and P. K. Meher, "A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 921–932, Feb. 2013.
- [16] B. K. Mohanty, P. K. Meher, S. Al-Maadeed, and A. Amira, "Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 1, pp. 120–133, Jan. 2014.
- [17] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 2, pp. 217–219, Feb. 2008.
- [18] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Mag.*, vol. 6, no. 3, pp. 4–19, Jul. 1989.