International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 02 - Issue 08 || August 2017 || PP. 20-26

Design of Fir Filter Architecture Using Manifold Steady Method

Arun Vignesh N[1] and Jayanthi D[2]
[1]*Department of Electronics and Communication Engineering, GRIET, Hyderabad, India*

Abstract: The power consumption and speed are the two main challenging factors in Very Large Scale Integrated Circuit (VLSI) design techniques. The computation saving is one of the way to obtain the optimized power consumption and speed. The design of finite-impulse response (FIR) filter using transpose form structure is naturally pipelined and upholds Manifold Steady Multiplication (MSM) technique. This MSM technique results in reduction of large complex computation. But, the transpose form configurations are not supporting the block processing. In the existing method, the possibility of realization of FIR filter in transpose form configuration to achieve efficient area and delay for large order FIR filters were explored. In the FIR filter structure the ripple carry adder is used to add the partial inner products. The ripple carry adder provides efficient area utilization but its operating speed is slow. In this proposed method, the carry look ahead adder is used to increase the speed and also to reduce the area and power consumption. The proposed structure significantly reduces the area delay product (ADP) and energy per sample (EPS) than the existing FIR structure. Application-specific 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.

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

1. Introduction

FIR digital filter is widely used in several digital signal processing applications, such as speech processing, 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 Manifold Steady Multiplication (MSM) methods [7], [11]–[13]. DA-based designs use lookup tables (LUTs) to store pre computed results to reduce the computational complexity. The MSM 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 MSM scheme is more effective, when a common operand is multiplied with more number of constants. Therefore, the MSM scheme is suitable for the implementation of large order FIR filters with fixed coefficients. But, MSM 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 MSM, 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 channel-izer, where FIR filters need to be implemented in a reconfigurable hardware to support multistandard 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 canonical 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 behaviour. 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

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)

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 MSM 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 MSM scheme for the block implementation of fixed FIR filters.

The remainder of this 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 applications are presented in Section III. Hardware and time complexities along with performance comparison are presented in Section IV. Finally, the conclusion is drawn in Section V.

2. 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

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

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.

Fig. 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.

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 grey color in Fig. 1).
These redundant computations of DFG-1 and DFG-2 can be avoided using no overlapped sequence of input blocks, as shown in Fig. 2. DFT-3 and DFT-4 of DFG-1 and DFG-2 for no overlapping input blocks are, respectively, shown in Fig. 2(a) and (b). As shown in Fig. 2(a) and (b), DFT-3 and DFT-4 do not involve redundant computation. It is easy to find that the entries in grey cells in DFT-3 and DFT-4 of Fig. 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)*.

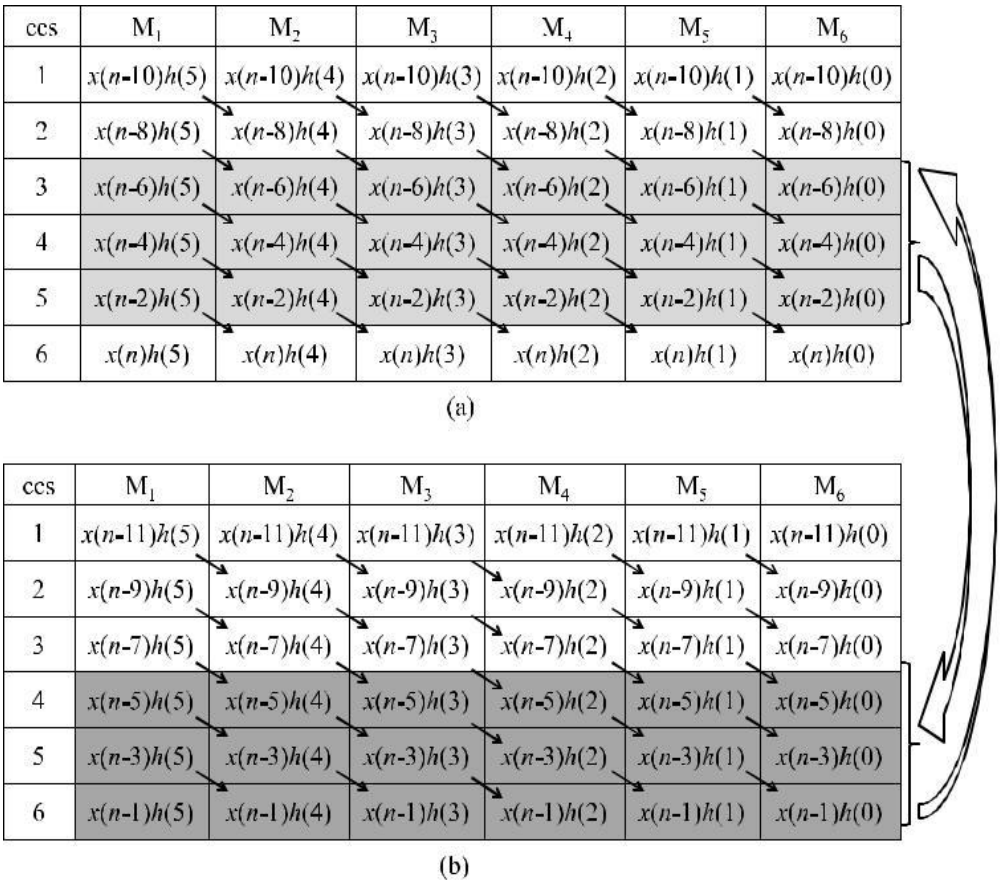


Fig.2. DFT of DFG-1 and DFG-2 for three no overlapped input blocks [*x(n), x(n−1)*], [*x(n−2), x(n−3)*], and [*x(n−4), x(n−5)*]. (a) DFT-3 for computation of output *y(n)*. (b) DFT-4 for computation of output *y(n − 1)*.

3. Proposed Structures

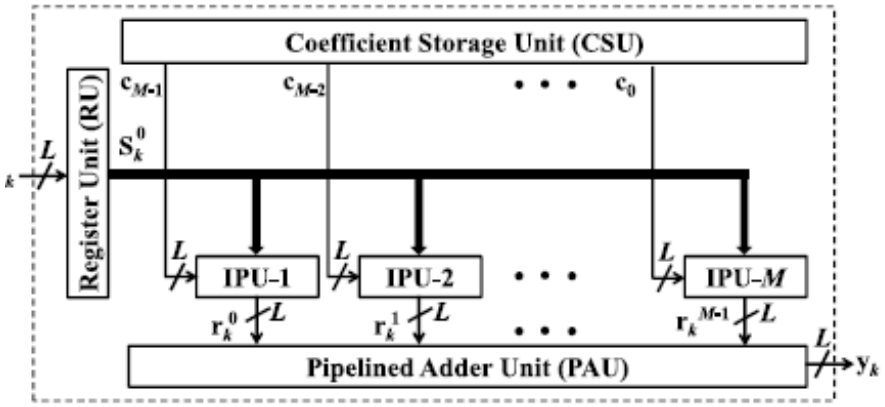
There are several applications where the coefficients of FIR filters remain fixed, while in some other applications, like SDR channelizer that requires separate FIR filters of different specifications to extract one of the desired narrowband channels from the wideband RF front end. These FIR filters 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 MSM scheme.

3.1. Proposed Structure for Transpose Form Block FIR

3.1.1. Filter for Reconfigurable Applications

The proposed structure for block FIR filter is [based on the recurrence relation of (12)] shown in Fig. 3 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. 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. 4(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

Fig 3.Proposed structure block for FIR filter

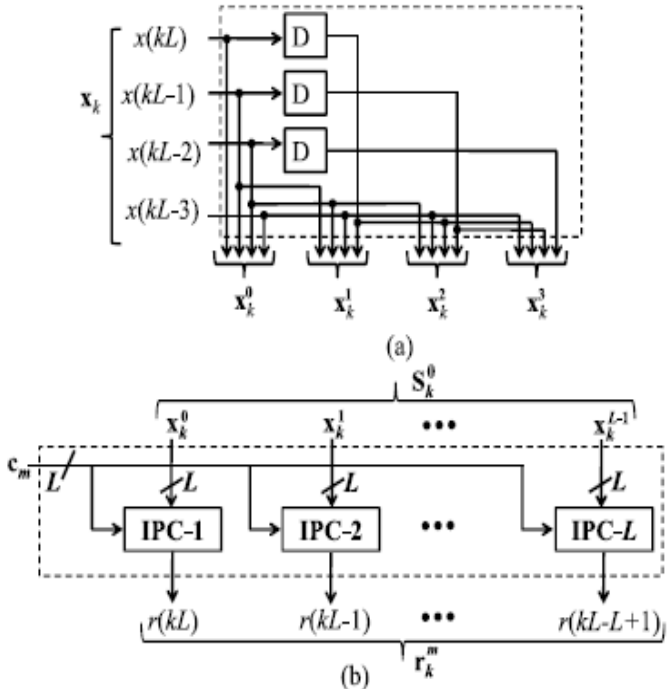
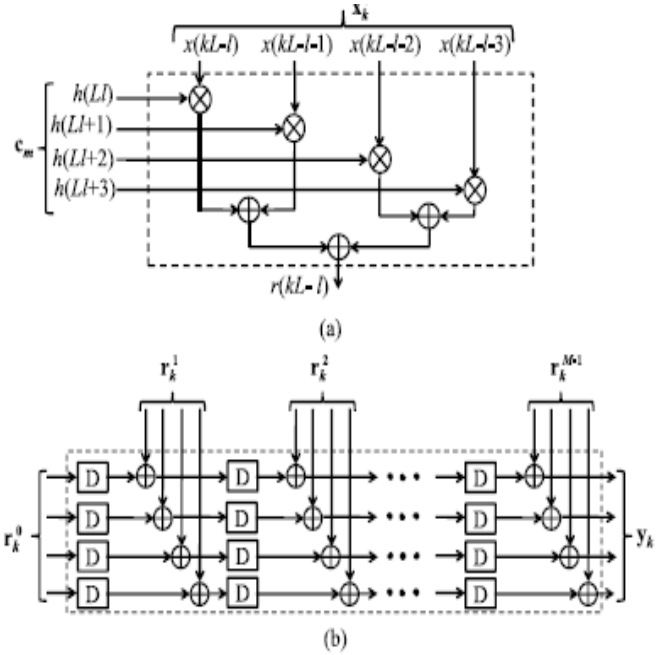


Fig. 4. (a) Internal structure of RU for block size $L = 4$. (b) Structure of $(m + 1)$th IPU.

such that during the *k*th cycle, the $(m + 1)$th IPU receives the weight vector $\mathbf{c}M-m-1$ 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 $(\mathbf{r}m\ k\ )$. 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. 4(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**0*k* and the coefficient vector **c***m*, and computes a partial result of inner product $r\ (kL - l)$, for $0 \leq l \leq L - 1$. Internal structure of $(l + 1)$th IPC for $L = 4$ is shown in Fig. 5(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. 5(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* + *T*FA log2 *L*, *TM* is one multiplier delay, *TA* is



one adder delay, and *T*FA is one full-adder delay.

Fig. 5. (a) Internal structure of (*l* + 1)th IPC for *L* = 4. (b) Structure of PAU for block size *L* = 4.

3.2. MSM-Based Implementation of Fixed-Coefficient FIR Filter

We discuss the derivation of MSM units for transpose form block FIR filter, and the design of proposed structure for fixed filters. For fixed-coefficient implementation, the CSU of Fig. 3 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 MSM units for a low-complexity realization. In the following, we show that the proposed formulation for MSM-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 [20] and to minimize the number of shift-add operations in the MSM blocks.

4. COMPLEXITIES AND PERFORMANCE CONSIDERATIONS

4.1. Hardware and Time Complexities

The proposed structure for reconfigurable application consists of one CSU, one RU, *M* IPUs, and one PAU. The CSU consists of *N* ROM units of *P* words each, where *P* is the number of FIR filters 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 RFIR 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)*] (flip flops) FFs; and processes *L* samples in every cycle where the duration of cycle period *T* = [*TM* + *TA* + *T*FA*(*log2 *L)*]. 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 [15]. We have derived the direct-form block FIR structure using [15, eq. (4)], and estimated its hardware and time complexities for comparison purpose.

4.2. Simulation Results

The proposed architecture improves the performance and power saving of FIR filter. In the proposed architecture the multiplier in conventional is replaced with pipelined modified booth multiplier. This modified booth multiplier architecture efficiently saves power and improves performance with efficient trade off comparatively with conventional architecture. The pipeline technique is widely used to improve the performance of digital circuits. As the number of pipeline stages is increased, the path delays of each stage are decreased and the overall performance of the circuit is improved.

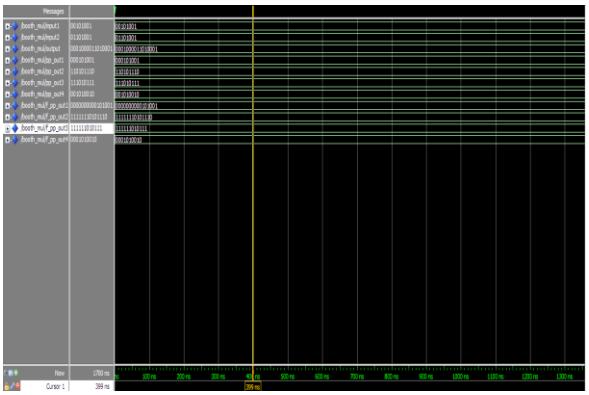


Fig 6. Output of booth multiplier

The CSD representation for a given number is unique and has two properties: The first is that the number of non-zero digits is minimal and the second is that the product of adjacent two digits is zero and widely used in implementing MSM's**.** Since CSD requires more number of adders

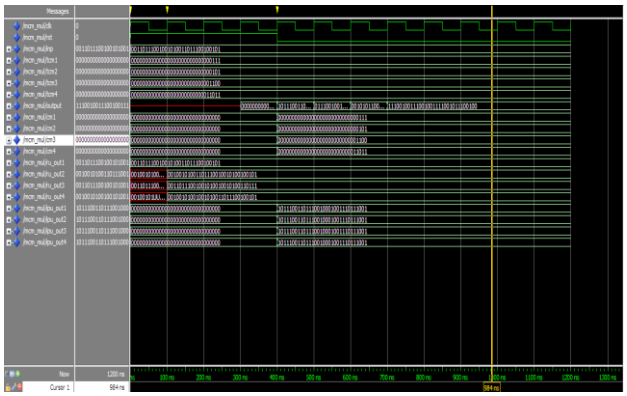


Fig 7. output of MSM multiplier

4.3. Output of Area, Delay and Power:

The area, delay and power are determined by using Xilinx software. The estimation of overall area of the existing fixed and reconfigurable application in our design which shown below.

Application	Existing			Proposed		
	Area	Power	Delay	Area	Power	Delay
Fixed	11%	186 mW	14.39 ns	7%	181 mW	4.67 ns
Reconfigurable	26%	268 mW	20.51 ns	20%	257 mW	16.89 ns

5. 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 MSM 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-from blocks FIR structure of [15].

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.
[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.
[19]. Greeshma A G, Sajithra S I, Surya kumari V,” Area Efficient Double Edge Triggered Double Tail Comparator”, IJETAE, Volume 4, Issue 4, April 2014.