FPGA Based Efficient Fast FIR Algorithm for Higher Order Digital FIR Filter

J. Selvakumar
Assistant Professor, Department of ECE,
SRM University, Kattankulathur 603 203
selva2802@gmail.com

S. Narendran
PG Student, Department of ECE,
SRM University, Kattankulathur 603 203
sridhar.naren@gmail.com

Vidhyacharan Bhaskar
Professor, Department of ECE,
SRM University, Kattankulathur 603 203
vcharan@gmail.com

Abstract *– The scope of the paper is to design a new Fast Finite-Impulse Response (FIR) Algorithms (FFAs) for parallel FIR filter structure, which are designed for symmetric coefficients that aim at reducing hardware cost in our design with a constraint that the filter tap must be a multiple of 2. The reduction in area is achieved by replacing the adder by a bulky multiplier. For example, for a 4 parallel 36-tap filter, the proposed structure saves 14 multipliers at the expense of 10 adders, whereas for a four-parallel 288-tap filter, the proposed structure saves 108 multipliers at the expense of 10 adders. Overall, the proposed parallel FIR structures can lead to significant hardware savings for symmetric coefficients from the existing FFA parallel FIR filter, especially when the length of the filter is very large.*

Keywords*:—Digital Signal Processing (DSP); Fast Finite-Impulse Response (FIR) algorithms (FFAs); Parallel FIR; symmetric convolution; Very Large Scale Integration (VLSI).*

I. Introduction

FIR filters are used in high frequency applications such as multimedia signal processing, whereas some other applications require high throughput with a low-power circuit such as Multiple-Input Multiple-Output (MIMO) systems used in mobile wireless communications. One of the drawbacks of using FIR filters is that the filter order tends to grow inversely proportional with the transition bandwidth of the filter. The order of the filter increases when narrowband transitions are required at the filter output. Hence, the design of area-efficient high-speed data path logic systems is one of the most substantial areas of research in VLSI system design. A higher order FIR digital filter is used in a video ghost canceller for broadcast television which reduces the effect of multipath signal echoes. In the proposed filter algorithm, if the order of the filter increases enormously, the area remains optimized. In the proposed algorithm, we have applied pipelining and parallel processing concepts that are used in VLSI DSP applications which can both be exploited to reduce the power consumption. Parallel processing increases the sampling rate by replicating hardware so that multiple inputs can be processed in parallel and multiple outputs are generated simultaneously. Due to the increase in the hardware cost and the block size, L, the parallel processing technique is avoided in practical implementation. Earlier work aimed at reducing the complexity of the parallel FIR Filter [1]-[7]. FFA structures successfully overcome the constraint that the hardware implementation cost of a parallel FIR filter has a linear increase along the block size L. Fast FIR algorithms (FFAs) introduced in [1]-[4] show that it can implement a L-parallel filter using approximately (2L-1) subfilter blocks, each of which is of length N/L. Using FFA technique, the number of multipliers reduces from (2N-N / L) to L x N. The ISC-based linear convolution structure is transposed to obtain a new hardware efficient fast parallel Finite-Impulse Response (FIR) filter structure, which saves a large amount of hardware cost, especially when the length of the FIR filter is large [7]. In [6]–[10], fast linear convolution is utilized to develop small-sized filtering structures and then long convolution is decomposed into several short convolutions, i.e., larger block-sized filtering structures can be constructed through iterations of the small-sized filtering structures.

The symmetry of coefficients has not been taken into consideration for the design of structures yet, which can lead to significant saving in hardware cost. In this paper, we provide new parallel FIR filter structures based on FFA consisting of advantageous polyphase decompositions, which can reduce amounts of multiplications in the subfilter section by exploiting the inherent nature of symmetric coefficients compared to the existing FFA fast parallel FIR filter structure.

This paper is organized as follows. A brief introduction of FFAs is given in Section II. In Section III, the proposed parallel FFA filter structures are presented. Section IV proposes cascaded FFA structures for symmetric convolutions. Section V investigates the complexity and comparisons. In Section VI, the description of hardware implementation and the experimental results are shown. Finally, Section VII presents the conclusions.

II. Fast FIR Algorithm (FFA)

An n-tap FIR filter which can be expressed in the general form as in (1)

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i), \quad n = 0,1,2,..,\infty, \qquad (1)$$

where $\{x(n)\}$ is an infinite-length input sequence and are the length-N FIR filter coefficients. Then, the traditional L-parallel FIR filter can be derived using polyphase decomposition as [3]

$$\sum_{p=0}^{L-1} Y_p(Z^L)Z^{-p} = \left(\sum_{q=0}^{L-1} X_q(Z^L)Z^{-q}\right)\left(\sum_{r=0}^{L-1} H_r(Z^L)Z^{-r}\right) \qquad (2)$$

978-0-7695-4902-6/12 $26.00 © 2012 IEEE
DOI 10.1109/ISED.2012.40

IEEE computer society

For L=2 parallel FIR filter structure as already discussed by Tsao and Choi [2].

A. Existing 3 x 3 Fast FIR Algorithm (L = 3 parallel)

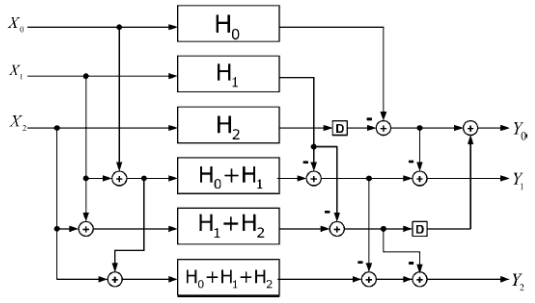


Fig. 1: Three-parallel FIR filter implementation using Exsiting FFA

With reference to the three-parallel FIR filters using FFA, the filter output namely Y_0, Y_1, Y_2 are extensively discussed [2].To utilize the symmetry of coefficients, the hardware implementation of (6) requires six length-N/3 FIR subfilter blocks, three preprocessing and seven post processing adders, and three N multipliers and $2N+4$ adders, which has reduced approximately one third over the traditional three-parallel filter hardware cost. The existing implementation of 2 x 2 FFA is shown in Fig. 1.

B. For L= 4 x 4 Parallel FIR Filter

According to (2), a four-parallel FIR filter output can be expresses as [2]

$$Y = Y_0 + Z^{-1}Y_1 + Z^{-2}Y_2 + Z^{-3}Y_3$$
$$= (X_0 + Z^{-1}X_1 + Z^{-2}X_2 + Z^{-3}X_3)(H_0 + Z^{-1}H_1 + Z^{-2}H_2 + Z^{-3}H_3)$$
$$Y = (X_0' + Z^{-1}X_1')(H_0' + Z^{-1}H_1')$$

where

$$X_0' = X_0 + Z^{-2}X_2,\ X_1' = X_1 + Z^{-2}X_3,$$
$$H_0' = H_0 + Z^{-2}H_2,\ H_1' = H_1 + Z^{-2}H_3.$$

$$Y = X_0'H_0' + Z^{-1}[(X_0' + X_1')(H_0' + H_1') - X_0'H_0' - X_1'H_1'] + Z^{-2}X_1'H_1'$$
$$X_0'H_0' = X_0H_0 + Z^{-2}[(X_0 + X_2)(H_0 + H_2) - X_0H_0 - X_2H_2] + Z^{-4}X_2H_2$$
$$X_1'H_1' = X_1H_1 + Z^{-2}[(X_1 + X_3)(H_1 + H_3) - X_1H_1 - X_3H_3] + Z^{-4}X_3H_3$$
$$(X_0' + X_1')(H_0' + H_1') = [(X_0 + X_1) + Z^{-2}(X_2 + X_3)]. \quad (3)$$

The hardware implementation of (3) requires eight length-N/4 FIR subfilter blocks, three pre-processing and fifteen post- processing adders, 9N/4 multipliers and (9N/4)+9 adders, which reduces hardware complexity by a significant factor.

III. PROPOSED FFA STRUCTURES FOR SYMMETRIC CONVOLUTIONS

The main objective of the proposed structure is to earn as many subfilter blocks as possible which contains symmetric coefficients so that half the number of multiplications in the single subfilter block can be reused for multiplications of all the taps, which is similar to the fact that a set of both odd and even symmetric coefficients would only require half the filter length of multiplications in a single FIR filter. Therefore, for an N-tap L-parallel FIR filter, the total amount of saved multipliers would be the number of subfilter blocks that contain symmetric coefficients times half the number of multiplications in a single subfilter block (N/2L).

A. 2 x 2 Proposed FIR FFA (L= 2 parallel)

From the existing L = 2 parallel FFA technique [2], a two-parallel FIR filter can also be written as

$$Y_0 = \left[\left\{\frac{1}{2}[(H_0 + H_1)(X_0 + X_1) + (H_0 - H_1)(X_0 - X_1)] - H1X1 + Z - 2H1X1\right.\right.$$

$$Y_1 = \frac{1}{2}[(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] \quad (4)$$

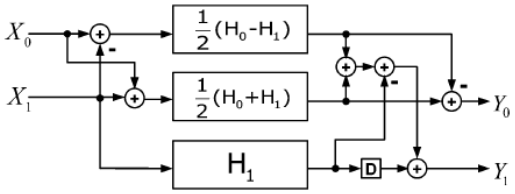


Fig. 2: Proposed FIR filter for two-parallel implementation

When it comes to a set of even symmetric coefficients, (4) can earn one more subfilter block containing symmetric coefficients than the existing FFA parallel FIR filter. Fig. 2 shows the implementation of the proposed two-parallel FIR filter based on (4). An example is demonstrated to get a clear perspective.

Example 1: Consider a 40-tap FIR filter with a set of symmetric coefficients applying to the proposed two-parallel FIR filter

{h(0), h(1), h(2), h(3), h(4), h(5), h(6),h(7),h(8),h(9),…h(39)}

where h(39) = h(0), h(38) = h(1), h(37) = h(2), h(36) = h(20), h(4) = h(19), h(5) = h(18) ,…. h(11) = h(12), applying to the proposed two-parallel FIR filter structure, and the top two subfilter blocks will be given as

$$H_0 \pm H_1 = \{h(0) \pm h(1), h(2) \pm h(3), h(4) \pm h(5), h(6) \pm h(7), ..., h(18) \pm h(19), h(20) \pm h(21), h(22) \pm h(23)\}$$

where $h(0) \pm h(1) = \pm(h(38) \pm h(39))$

$h(4) \pm h(5) = \pm(h(34) \pm h(35))$

$h(6) \pm h(7) = \pm\big(h(16) \pm h(17)\big)$, etc. (5)

As can be observed from the above example, two of the three subfilter blocks from the proposed two-parallel FIR filter structures, H_0-H_1 and H_0+H_1, are with symmetric coefficients, now, as (5), which means the subfilter block can be realized with only half the amount of multipliers required. Each output of multipliers responds to two taps. Note that the transposed direct-form FIR filter is employed. Compared to the existing FFA two-parallel FIR filter structure, the proposed FFA structure leads to one more

subfilter block which contains symmetric coefficients. However, it comes with the cost of increase in the amount of adders in preprocessing and post processing blocks. In this case, two additional adders are required for L=2.

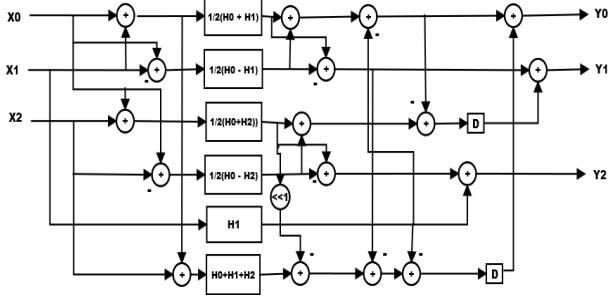
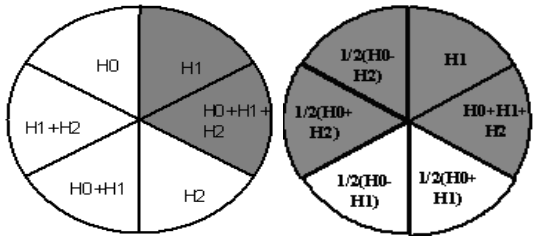
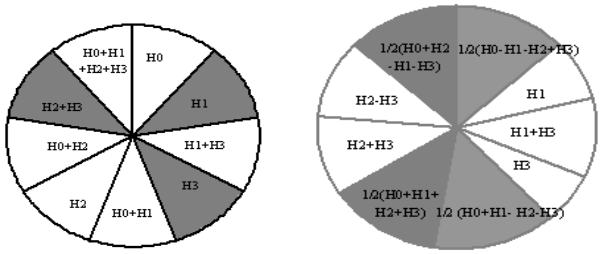


Fig. 3: Proposed three-parallel FIR filter implementation



Existing FFA Proposed FFA

Fig. 4: Comparison of sub-filter blocks between existing FFA and the proposed FFA three-parallel FIR structures



Existing FFA Proposed FFA

Fig. 5: Comparison of sub-filter blocks between existing FFA and the proposed FFA four-parallel FIR structures.

B. 3 x 3 Proposed FIR structure using FFA (L=3)

With a similar approach, from (4), the proposed three-parallel FIR filter can also be written as (6). Fig. 3 shows the implementation of the proposed three-parallel FIR filter. When the number of symmetric coefficients, N, is a multiple of 3, the proposed three-parallel FIR filter structure presented in (6) enables four subfilter blocks with symmetric coefficients in total, whereas the existing FFA parallel FIR filter structure has only two ones out of six subfilter blocks. A comparison is shown in Fig. 4, where the shadow blocks stand for the subfilter blocks which contains symmetric coefficients. Therefore, for an N-tap three-parallel FIR filter, the proposed structure can save N/3 multipliers from the existing FFA structure. However, again, the proposed three-parallel FIR structures also bring an overhead of seven additional adders in preprocessing and post processing blocks.

$$Y_0 = \left[\left\{\frac{1}{2}[(H_0+H_1)(X_0+X_1)+(H_0-H_1)(X_0-X_1)]-\right.\right.$$

$$H1X1+Z-3(H0+H1+H2)((X0+X1+X2)-(H0+H2)(X0+X2)-12(H0+H1)(X0+X1)-(H0-H1)(X0-X1)-H1X1$$

$$Y_1 = \left[\left\{\frac{1}{2}[(H_0+H_1)(X_0+X_1)-(H_0-H_1)(X_0-X_1)]\right\}+\right.$$

$$Z-312(H0+H2)(X0+X2)+(H0-H2)(X0-X2)-12(H0+H1)(X0+X1)+(H0-H1)(X0-X1)+H1X1$$

$$Y_2 = \frac{1}{2}[(H_0+H_2)(X_0+X_2)-(H_0-H_2)(X_0-X_2)]+H_1X_1 \quad (6)$$

C. 4 x 4 Proposed FFA Technique (L=4)

With a similar approach, from (9), a three-parallel FIR filter can also be written as (10). Fig. 9 shows the implementation of the proposed four-parallel FIR filter. When the number of symmetric coefficients, N, is a multiple of 4, the proposed four-parallel FIR filter structure presented in (10) enables four subfilter blocks with symmetric coefficients in total, whereas the existing FFA parallel FIR filter structure has only two ones out of six subfilter blocks. A comparison is shown in Fig. 6, where the shadow blocks stand for subfilter blocks which contain symmetric coefficients. Therefore, for an N-tap four-parallel FIR filter, the proposed structure can save 3N/8 multipliers from the existing FFA structure. However, again, the proposed four-parallel FIR structures also brings an overhead of nine additional adders in preprocessing and 21 adder in post processing blocks. Here,

$$X_0'H_0' = \left\{\frac{1}{2}\left[\left(H_0'+H_1'\right)\left(X_0'+X_1'\right)+\left(H_0'-H_1'\right)\left(X_0'-X_1'\right)\right]-H_1'X_1'\right\}$$

$$= \left\{\begin{array}{l}\frac{1}{2}\left[\left(H_0+z^{-2}H_2+H_1+z^{-2}H_3\right)\left(X_0+z^{-2}X_2+X_1+z^{-2}X_3\right)\right.\\ \left.+\left(H_0+z^{-2}H_2-H_1-z^{-2}H_3\right)\left(X_0+z^{-2}X_2-X_1-z^{-2}X_3\right)\right]\\ -\left(H_1+Z^{-2}H_3\right)\left(X_1+Z^{-2}X_3\right)\end{array}\right\} \quad (7)$$

IV. Proposed Cascaded FFA Structures for Symmetric Convolutions

The proposed parallel FIR structure enables the reuse of multipliers in parts of the subfilter blocks but also brings more adder cost in preprocessing and post processing blocks. When cascading the proposed FFA parallel FIR structures for larger parallel block factor, L, the number of adders can become larger. Therefore, other than applying the proposed FFA FIR filter structure to all the decomposed subfilter blocks, the existing FFA structures which have more compact operations in preprocessing and post processing blocks are employed for those subfilter blocks that contain no symmetric coefficients, whereas the proposed FIR filter structures are still applied to the rest of subfilter blocks with symmetric coefficients.

For example, a (*m*-by-*m*) FFA can be cascaded with a (*n-by-n*) FFA to produce a (*m* x *n*)-parallel filtering structure. The set of FIR filters that result from the application of the (*m*-by-*m*) FFAs are further decomposed

one at a time by the application of (*n-by-n*) FFA.

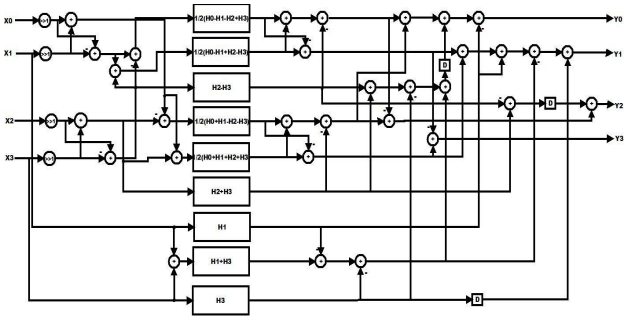


Fig. 6. Four Parallel Proposed FFA Algorithm

The resulting set of filters will be of length $\frac{N}{mn}$. When cascading the FFAs, it is important to keep track of both the number of multipliers and the number of adders required for the filtering structure. The number of required multipliers is calculated as follows:

$$M = \frac{N}{\prod_{i=1}^{r} L_i} \prod_{i=1}^{r} M_i \qquad (8)$$

where 'r' is the number of FFAs used, L_i is the block size of the FFA at step-I, M_i is the number of filters that results from the application of the i^{th} FFA, and N is the length of the filter. The number of required adders that is calculated as follows:

$$A = A_1 \prod_{i=2}^{r} L_i + \sum_{i=2}^{r} [A_i (\prod_{j=i+1}^{r} L_j)(\prod_{k=1}^{i-1} M_k) + [\prod_{i=1}^{r} M_i][\frac{N}{\prod_{i=1}^{r} L_i} - 1] \ , \qquad (9)$$

where A_i is the number of pre/post-processing adders required by the i^{th} FFA. Consider the case of cascading two (2-*by*-2) FFAs. The resulting 4-parallel filtering structure would require a total of 9*N*/4 multipliers and 20 + 9(*N*/4 -1) adders for implementation. The reduced-complexity of L=4 parallel filtering structure represents a hardware (area) savings of nearly 44% when compared to 4*N* multipliers required in the traditional L=4 parallel FIR filtering structure.

Thus, the cascading approach for parallel FIR filter structures with larger block factor L can be realized. The proposed eight-parallel FIR filter will lead to seven more symmetric subfilter blocks, equivalently 7N/16 multipliers saved for an N-tap filter than the existing FFA with the overhead of additional 54 adders.

V. Area complexity Analysis and comparison

When the L-parallel FIR filter comes with a set of symmetric coefficients of length N, the number of required multipliers for the proposed parallel FIR filter structures is provided by (10) and (11).

Case 1: When $\frac{N}{\prod_{i=1}^{r} L_i}$ is even,

$$M = \frac{N}{\prod_{i=1}^{r} L_i} (\prod_{i=1}^{r} M_i - \frac{S}{2})$$

Case 2: When $\frac{N}{\prod_{i=1}^{r} L_i}$ is odd,

$$M = \frac{N}{\prod_{i=1}^{r} L_i} (\prod_{i=1}^{r} M_i - \frac{S}{2})(\frac{N}{\prod_{i=1}^{r} L_i} - 1) \qquad (10)$$

where L_i is the small parallel block size such as (2 x 2) or (3 x 3) FFA. Here, 'r' is the number of FFAs used. M_i is the number of subfilter blocks resulting from i^{th} FFA, 'S' is the number of subfilter blocks containing symmetric coefficients. The number of the adders required in subfilter section can be given by

$$A_{sub} = \prod_{i=1}^{r} M_i (\frac{N}{\prod_{i=1}^{r} L_i} - 1) \qquad (11)$$

A comparison between the multiplier and adder complexity for the proposed and the existing FFA structures for even symmetric coefficients for different lengths 'N' (tap-order), for different level of parallelisms (L) is summarized in Table I. Also, a comparison between the proposed structures and other structures for an 12, 27, 243 and 432-tap FIR filter with parallel block size of L = 2, 3, 4 and 6 is shown in Table I.

Table I : Comparison of proposed and the existing FFA structures

(i) For L=2 Parallel FIR Filter:

Length		**12-tap**	**27-tap**	**243-tap**	**432-tap**
Multipliers	Existing FFA	18	42	365	648
	Proposed FFA	15	34	306	544
	Reduced	3	8	59	104
Adders (Pre/Post)	Existing FFA	4	4	4	4
	Proposed FFA	6	6	6	6
	Increased	2	2	2	2
Sub-Filters Block Adders		15	38	360	645

(ii) For L=3 Parallel Filter:

Length		**12 - tap**	**27 - tap**	**243-tap**	**432-tap**
Multipliers	Existing FFA	24	54	486	864
	Proposed FFA	20	45	405	720
	Reduced	4	9	81	144
Adders (Pre/Post)	Existing FFA	10	10	10	10
	Proposed FFA	17	17	17	17
	Increased	7	7	7	7
Sub-Filters Block Adders		18	48	480	858

(iii) For L=4 Parallel FIR Filter:

Length		**12-tap**	**27-tap**	**243-tap**	**432-tap**
Multipliers	Existing FFA	27	61	547	972
	Proposed FFA	22	49	441	792
	Reduced	5	12	106	180
Adders (Pre/Post)	Existing FFA	20	20	20	20
	Proposed FFA	31	31	31	31
	Increased	11	11	11	11
Sub-Filters Block Adders		18	52	518	963

(iv) For L=6 Parallel Filter,

Length	**12-tap**	**27-tap**	**243-tap**	**432-tap**

Multipliers	Existing FFA	36	81	729	1296
	Proposed FFA	30	67	603	1080
	Reduced	5	14	126	216
Adders (Pre/Post)	Existing FFA	42	42	42	42
	Proposed FFA	74	74	74	74
	Increased	32	32	32	32
Sub-Filters Block Adders		24	54	711	1278

The number of required multipliers, number of reduced multipliers, number of required adders in subfilter section, number of required adders in pre-processing and post-processing blocks are shown in Table I.

VI. FPGA Implementation Analysis:

The existing and proposed FFA structures discussed to reduce further complexity of the FFA based Digital FIR filter are implemented in Verilog HDL and targeted on Xilinx Virtex FPGA device for the filter length of L = 12 and L = 27 with word length of 8-bit. Implementation results are tabulated as comparison results in Table II.

TABLE II: Comparison of Area

Length	Structure	Area*	
		L=2	L=3
12-tap	Proposed FFA	1080	1195
	Existing FFA	1290	1860
27-tap	Proposed FFA	920	1010
	Existing FFA	1090	1980

* in terms of LUT slices/Arithmetic Blocks

TABLE III: Comparison of Delay

Length	Structure	Delay (in ns)	
		L=2	L=3
12-tap	Proposed FFA	9.95	10.56
	Existing FFA	7.8	8.12
27-tap	Proposed FFA	10.65	11.45
	Existing FFA	8.4	9.4

VII. CONCLUSIONS

In this paper, we have presented efficient parallel FIR filter structures, which are advantages for symmetric convolutions when the number of taps is a multiple of 2 or 3. Multiplier consumes the major area in the hardware for parallel FIR implementation. The proposed structure extracts the nature of even symmetric coefficients and save a significant multiplier area at the expense of additional adders in pre-processing and post processing adder block in FIR filter. From Table II, it evident that the area complexity is reduced by more than 70% for L=3 Parallel, 27-tap FIR Filter, with an increase in 18% delay for the output of filter. The percentage of reduction in area complexity may be higher for higher order filter taps. Since an adder occupies lesser area when compared to multipliers, it is advantageous to exchange multipliers with adders in terms of hardware cost. Moreover the number of increased adders stays still when the length of FIR filter becomes large, whereas the number of reduced multipliers increases along with the length of the filter. In this paper, we have proposed new parallel FIR structures consisting of advantageous polyphase decompositions dealing with symmetric convolutions comparatively better than the existing FFA structure [2] in terms of hardware consumption.

REFERENCES

[1] D. A. Parker and K. K. Parhi, "Low-Area/Power Parallel FIR Digital Filter Implementations," *Journal of VLSI Signal Processing Systems*, vol. 17, no. 1, pp. 75–92, Sep. 1997.

[2] Yu-Chi Tsao and Ken Choi, "Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions based on Fast FIR Algorithm", vol. 20, no. 2, pp. 366-370, Feb. 2012.

[3] J. G. Chung and K. K. Parhi, "Frequency-Spectrum-based low-area low-power Parallel FIR Filter Design," *EURASIP Journal of Applied Signal Processing*, vol. 2002, no. 9, pp. 444–453, Jan. 2002.

[4] K. K. Parhi*, VLSI Digital Signal Processing Systems: Design and Implementation,* Wiley New York, 1999.

[5] Z.-J. Mou and P. Duhamel, "Short-length FIR filters and their use in fast non-recursive filtering," *IEEE Transactions on Signal Processing*, vol. 39, no. 6, pp. 1322–1332, June 1991.

[6] J. I. Acha, "Computational structures for fast Implementation of L-path and L-block digital filters," *IEEE Transactions on Circuit and Systems*, vol. 36, no. 6, pp. 805–812, June 1989.

[7] C. Cheng and K. K. Parhi, "Hardware Efficient Fast Parallel FIR filter structures based on iterated short convolution," *IEEE Transactions on Circuits Systems I, Regular Papers*, vol. 51, no. 8, pp. 1492–1500, Aug. 2004.

[8] C. Cheng and K. K. Parhi, "Further complexity reduction of parallel FIR filters," *Proceedings of the IEEE International Symposium on Circuits Systems (ISCAS 2005)*, Kobe, Japan, May 2005.

[9] C. Cheng and K. K. Parhi, "Low-cost Parallel FIR structures with 2-stage parallelism," *IEEE Transactions on Circuits Systems I, Regular Papers*, vol. 54, no. 2, pp. 280–290, Feb. 2007.

[10] I. S. Lin, and S. K. Mitra, "Overlapped Block Digital Filtering," *IEEE Transactions on Circuits Systems II, Analog Digital. Signal Processing*, vol. 43, no. 8, pp. 586–596, Aug. 1996.