

Efficient Implementation of Parallel Linear Phase FIR Filters using Polyphase Decomposition

Jani Thivya.T

PG scholar, Department of ECE
Kumaraguru College of Technology
Coimbatore-641 049

R.Latha

Asso.Professor, Department of ECE
Kumaraguru College of Technology
Coimbatore-641 049

ABSTRACT

This paper presents an efficient approach that greatly reduces the hardware consumption during the design of FIR filters. Linear phase FIR filters are designed by exploiting the nature of symmetric odd coefficients. The efficient usage of coefficients limits the number of multipliers while simultaneously increasing the number of adders, which does not influence the hardware cost to a greater extent. Parallel processing together with linear phasing is a powerful technique which can be used to increase the throughput of the FIR filter or reduce the power consumption of the FIR filter. Replacement of adders instead of multipliers is advantageous because adders weigh less in cost in terms of its silicon area and also the number of sub filter blocks remains fixed and does not increase along with the length of the FIR filter. By using the combination of fast FIR filtering and area reduction technique, a major reduction of multipliers is done.

General Terms

Fast Fir Algorithm(FFA), Polyphase Decomposition

Keywords

Digital Signal Processing (DSP) - FIR filter –Linear phase – Parallel processing – odd coefficients.

1. INTRODUCTION

Digital Signal Processing plays a major role in multimedia applications where filters find their own importance. Among those filters, FIR filters' demand gets higher and higher day by day. Filter is the most fundamental processing element in any digital processing system. FIR filters find their applications in DSP systems ordering from communications to video processing. In certain applications, FIR filters must be low-power or high speed supporting structures. For certain applications like video television broadcasts, higher order FIR filters are essential. Design of these filters leads to hardware complexity and consumption of area and power. There are certain techniques to reduce the complexity of the larger size filter blocks. In order to design these filters, [1,2,4] polyphase decomposition is to be carried out where small-sized sub filter blocks are derived first and those sub-structures are cascaded or iterated to construct larger size filter blocks. This decomposition for FIR filter is used as a processing core to implement sub filters of proposed parallel FIR filters. Furthermore there have been papers proposing the FIR implementation using [5] pipelining and parallel processing. Pipelining leads to increase in number of latches and system latency. Parallel processing increases the sampling rate by replicating the hardware. But parallel processing loses its

advantage in practical implementation. Both the techniques reduce the power consumption to some extent. Considering the inefficiency of the above techniques, the basic nature of symmetric coefficients together with the polyphase decomposition is exploited in this paper to further reduce the amount of multipliers. For large values of N, [6-10] fast FIR filtering algorithms (FFAs) have been implemented where the polynomial formulation of FIR filters are equivalent to the filtering operations. The resulting FIR structure using FFA of length (N/L) require (2N-N/L) multipliers, anywhere the number of filtering operations are (2L-1). Without using the FFA technique, the required number of multipliers is N×L. For example, let N=4 and L=2. The traditional 2-parallel approach requires 8 multipliers where 2-parallel fast filtering requires 6 multipliers. As FFA is of most importance, a brief introduction of it is discussed in the Section 2. In Section 3, the proposed parallel structures using FFA are implemented. In section 4, mathematical formulae for multipliers are shown. In section 5, comparison result is shown.

2. FAST FIR ALGORITHM (FFA)

Consider an N-tap filter expressed in general form as

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

Where $x(n)$ an infinite length is input sequence and $h(i)$ represents the length-N filter FIR coefficients. In order to show the difference between the traditional parallel and the fast parallel approaches, the various structures have been derived. The traditional L-parallel filter using the polyphase decomposition can be given as,

$$\sum_{i=0}^{L-1} Y_i(z^L)z^{-i} = \sum_{j=0}^{L-1} H_j(z^L)z^{-j} \sum_{k=0}^{L-1} X_k(z^L)z^{-k} \quad (2)$$

Where $Y_i(z) = \sum_{m=0}^{\infty} z^{-m} y_{mL+i}$, $H_i(z) = \sum_{m=0}^{L-1} z^{-m} h_{mL+i}$, $X_i(z) = \sum_{m=0}^{\infty} z^{-m} x_{mL+i}$ for $i = 0,1,\dots,L-1$

This equation shows that the filtering equation can be realized using L^2 FIR filters of length N/L. This complexity can be reduced using FFA structures.

2.1 2×2(L=2) FFAs

From (2) with L=2, it can be given as,

$$Y_0 + z^{-1}Y_1 = (H_0 + z^{-1}H_1)(X_0 + z^{-1}X_1),$$

$$= H_0X_0 + z^{-1}(H_0X_1 + H_1X_0) + z^{-2}H_1X_1. \quad (3)$$

This implies that,

$$Y_0 = H_0X_0 + z^{-2}H_1X_1$$

$$Y_1 = H_0X_1 + H_1X_1 + H_1X_0 \quad (4)$$

Traditional implementation of (4) is shown in fig.1. This structure computes a block of two outputs using 4 length N/2 FIR filters and post processing additions, which requires 2N multipliers and 2N-2 adders.

Equ.(4) is obtained in a different form as,

$$Y_0 = H_0X_0 + z^{-2}H_1X_1$$

$$Y_1 = (H_0 + H_1)(X_0 + X_1) - H_0X_0 - H_1X_1 \quad (5)$$

Where $H_{i+j} = H_i + H_j$ and $X_{i+j} = X_i + X_j$.

Implementation of (5) is shown in fig.2. This structure computes a block of 2 outputs using 3 length N/2 FIR filters and 4 preprocessing and post processing additions, which require 3N/2 multipliers and 3(N/2-1)+4 adders.

When an FIR filter is constructed using a multiplier less approach, the hardware consumption is directly proportional to the number of non-zero bits in the filter coefficients.

As the implementation cost of a multiplier is much greater than that of an adder, the cost to implement the parallel filtering structure can be approximated as being proportional to the number of multipliers required for the implementation. The hardware consumption of parallel fast FIR filter is 25% less when compared to traditional parallel FIR filter.

2.2 3x3(L=3) FFAs

The 3x3 FFA is similar to 2x2 FFA from the stand point that it uses pre/post processing additions to reduce the number of multipliers needed in the implementation block structure. This implementation produces a structure of block size 3. Similar to previous implementation, this structure of 3x3 fast parallel FIR is compared to the traditional 3x3 parallel FIR.

From (2) with L=3, the traditional FIR structure can be given as,

$$Y_0 = H_0X_0 + z^{-3}(H_1X_2 + H_2X_1)$$

$$Y_1 = (H_0X_1 + H_1X_0) + z^{-3}H_2X_2$$

$$Y_2 = H_0X_2 + H_1X_1 + H_2X_0 \quad (6)$$

From the above equation of direct implementation of traditional FIR, it is shown that, it computes a block of 3 outputs using 9 length N/3 FIR filters and 6 post processing additions, which require 3N multipliers and 3N-3 adders.

The traditional 3-parallel FIR structure requires 3N multiplications and 3(N-1) additions.

By manipulating (5) through a series of steps, the number of filtering operations can be reduced which in turn reduces the total number of multipliers required for realizing 3-parallel fast filtering structure.

The equation for 3-parallel fast FIR filtering can be obtained by manipulation of equation (5) with L=3.

Fig.3. shows the three-fast parallel FIR filter structure.

$$Y_0 = H_0X_0 - z^{-3}H_2X_2 + z^{-3}[(H_1 + H_2)(X_1 + X_2) - H_1X_1]$$

$$Y_1 = [(H_0 + H_1)(X_0 + X_1) - H_1X_1] - [H_0X_0 - z^{-3}H_2X_2] \quad (7)$$

$$Y_2 = [(H_0 + H_1 + H_2)(X_0 + X_1 + X_2)] - [(H_0 + H_1)(X_0 + X_1) - H_1X_1] - [H_1 + H_2)(X_1 + X_2) - H_1X_1]$$

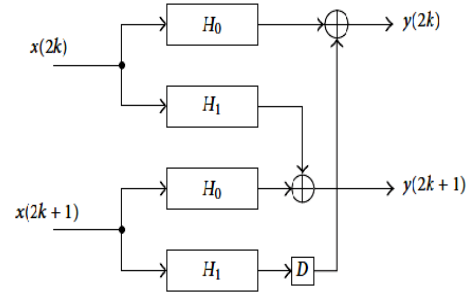


Fig.1. Traditional 2-parallel FIR filter

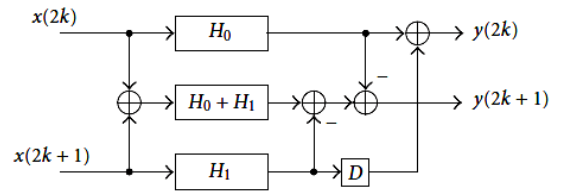


Fig.2. 2-parallel implementation using FFA

3. PROPOSED STRUCTURES FOR SYMMETRIC CONVOLUTIONS OF ODD LENGTH

The main idea is to generate half the number of multipliers within a single subfilter block which can be utilized for the multiplication of whole taps. So it is necessary to manipulate the polyphase decomposition to earn as many subfilter blocks as possible exploiting the symmetry of coefficients.

3.1 3x3 Proposed FFA (L=3)

In this section, new three-parallel FIR filter structures are proposed, which enables more multipliers sharing in the subfilter section and, therefore, can save more hardware cost over the existing FFA. The existing two-parallel FFA structure naturally has benefits to symmetric convolutions in odd length. When it comes to a set of odd-length symmetric coefficients, i.e., H_0 and H_1 shown in Fig.1.

3.1.1 Proposed Structure 3A, ((N mod 3) = 0):

For a set of symmetric coefficients in odd length N, when (N mod 3) equals zero, (8) can earn two more subfilter blocks containing symmetric coefficients than (7) is shown in Fig.4.

This can be described with an example. Considering a 27-tap FIR filter with a set of symmetric coefficients as follows:

$$\{h(0), h(1), h(2), h(3), h(4), h(5), \dots, h(25), h(26)\}$$

Where $h(0) = h(26), h(1) = h(25), h(2) = h(24), h(3) = h(23), \dots, h(12) = h(14)$, applying to the proposed structure 3 A, and then, two more subfilter blocks are gained with symmetric coefficients as,

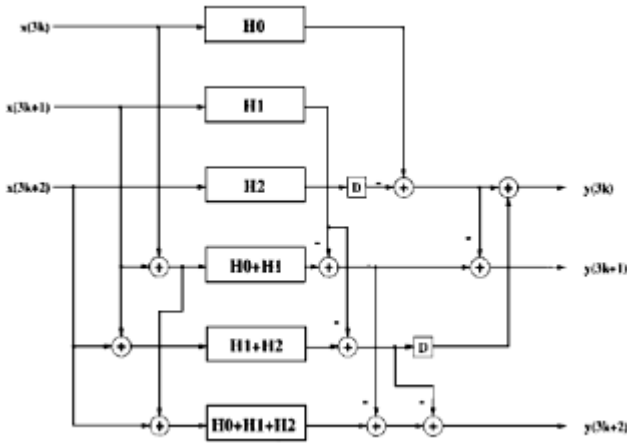


Fig.3. 3-parallel FIR implementation using FFA

$$H_0 \pm H_2 = \{h(0) \pm h(2), h(3) \pm h(5), h(6) \pm h(8), \dots, h(18) \pm h(20), h(21) \pm h(23), h(24) \pm h(26)\}$$

Where $h(0) \pm h(2) = \pm(h(24) \pm h(26))$
 $h(3) \pm h(5) = \pm(h(21) \pm h(23))$
 $h(6) \pm h(8) = \pm(h(18) \pm h(20))$
 $h(9) \pm h(11) = \pm(h(15) \pm h(17))$

$$Y_0 = H_0X_0 + z^{-3} \times \{(H_1 + H_2)(X_1 + X_2) - H_1X_1 - \left((H_0 + H_2)(X_0 + X_2) - H_0X_0 - \frac{1}{2} [(H_0 + H_2)(X_0 + X_2) - (H_0 - H_2)(X_0 - X_2)] \right)\}$$

$$Y_1 = (H_0 + H_1 + H_2)(X_0 + X_1 + X_2) - (H_1 + H_2)(X_1 + X_2) - (H_0 + H_2)(X_0 + X_2) + \left\{ (H_0 + H_2)(X_0 + X_2) - \frac{1}{2} \times [(H_0 + H_2)(X_0 + X_2) - (H_0 - H_2)(X_0 - X_2)] - H_0X_0 \right\} + z^{-3} \{ (H_0 + H_2)(X_0 + X_2) - (H_0 - H_2)(X_0 - X_2) \} - H_0X_0$$

$$Y_2 = H_1X_1 + \frac{1}{2} \times [(H_0 + H_2)(X_0 + X_2) - (H_0 - H_2)(X_0 - X_2)] \quad (8)$$

Comparing with the existing FFA three-parallel FIR filter structure, the proposed structure leads to two more subfilter blocks, which contain symmetric coefficients. Hence for an N-tap three-parallel FIR filter, the proposed structure can save N/3 multipliers from the existing FFA structure.

3.1.2 Proposed Structure 3B, ((N mod 3)=1):

For a set of symmetric coefficients in odd length N, when (N mod 3) equals 1, such as N=25, the proposed structure 3B can earn one more subfilter block with symmetric coefficients over the existing FFA as shown in equ.(9) i.e., presented in Fig.5.

$$Y_0 = H_0X_0 + z^{-3} \times \left\{ \frac{1}{2} [(H_1 + H_2)(X_1 + X_2) - (H_1 - H_2)(X_1 - X_2)] \right\}$$

$$Y_1 = (H_0 + H_1 + H_2)(X_0 + X_1 + X_2) - (H_1 + H_2)(X_1 + X_2) - (H_0 + H_1)(X_0 + X_1) + \left\{ (H_1 + H_2)(X_1 + X_2) - \frac{1}{2} \times [(H_1 + H_2)(X_1 + X_2) - (H_1 - H_2)(X_1 - X_2)] - H_2X_2 \right\} + z^{-3}H_2X_2$$

$$Y_2 = (H_0 + H_1 + H_2)(X_0 + X_1 + X_2) - (H_1 + H_2)(X_1 + X_2) - (H_0 + H_1)(X_0 + X_1) + 2 \left\{ (H_1 + H_2)(X_1 + X_2) - \frac{1}{2} \times [(H_1 + H_2)(X_1 + X_2)] - H_2X_2 \right\} \quad (9)$$

3.1.3 Proposed structure 3C, ((N mod 3)=2):

For a set of symmetric coefficients in odd length N, when (N mod 3) equals 2, such as N=23, the proposed structure 3C in Fig.6, represented in (10), can earn one more sub filter block containing symmetric coefficients over the existing FFA.

$$Y_0 = \left\{ \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) + (H_0 - H_1)(X_0 - X_1)] - H_1X_1 \right\} + z^{-2}H_1X_1$$

$$Y_1 = \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] + z^{-3}H_2X_2$$

$$Y_2 = H_1X_1 + \left\{ (H_0 + H_2)(X_0 + X_2) - \left\{ (H_0 + H_1)(X_0 + X_1) - \frac{1}{2} \times [(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] - H_1X_1 \right\} - H_2X_2 \right\} \quad (10)$$

3.2 Proposed Cascading Scheme for FFA

In proposed cascading process, instead of applying the existing small-sized structured FFA's to every stage, interleaving of multiple various small-sized structures can be done to fully exploit the symmetry of coefficients. The cascading of FFA's is a straight-forward application. For example, a (m x m) FFA can be cascaded with a (n x n) FFA to produce (m x n) parallel filtering structure. The resulting filters will be of length $\frac{N}{m \times n}$. During the cascading of the FFA's, it is important to keep track of both the number of multipliers and the number of adders required for the filtering structure.

4. MATHEMATICAL FORMULAE

The required number of multipliers for a L-parallel filter with symmetric coefficients of odd length N can be estimated by (11) and (12) as,

Case 1:

When,

$$\left[\frac{N}{\prod_{i=1}^r l_i} \right] \text{ is even, } M = \left[\frac{N}{\prod_{i=1}^r l_i} \right] \left(\prod_{i=1}^r M_i - \frac{S}{2} \right) \quad (11)$$

Case 2:

When,

$$\left[\frac{\square}{\prod_{i=1}^r \square_i} \right] \square \square \square \square, \square = \left[\frac{\square}{\prod_{i=1}^r \square_i} \right] \prod_{i=1}^r \square_i - \frac{\square}{2} \left(\left[\frac{\square}{\prod_{i=1}^r \square_i} \right] - 1 \right) \quad (12)$$

$\square \square$ is the small parallel block size such as (2 x 2) or (3 x 3) FFA.

\square is the number of FFAs used.

$\square \square$ is the number of subfilter blocks resulted from the \square th FFA.

\square is the number of subfilter blocks containing symmetric coefficients.

The number of required adders in the subfilter section can be shown by using the below formula (13),

$$\square_{\square\square\square} = \prod_{\square=1}^{\square} \square_{\square} \left(\left\lceil \frac{\square}{\prod_{\square=1}^{\square} \square_{\square}} \right\rceil - 1 \right) \quad (13)$$

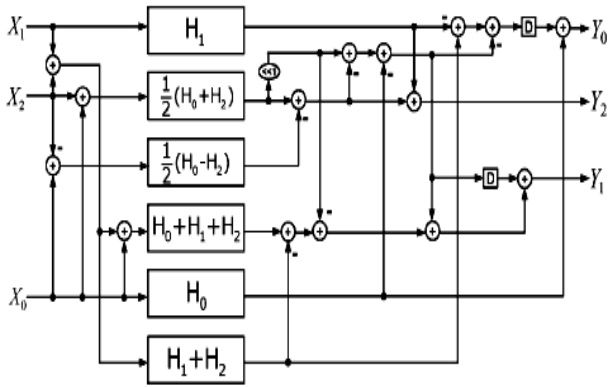


Fig.4. Implementation of the proposed 3A structure

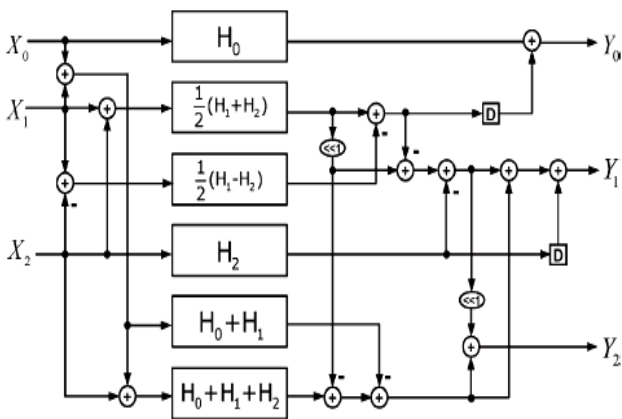


Fig.5. Implementation of the proposed 3B structure

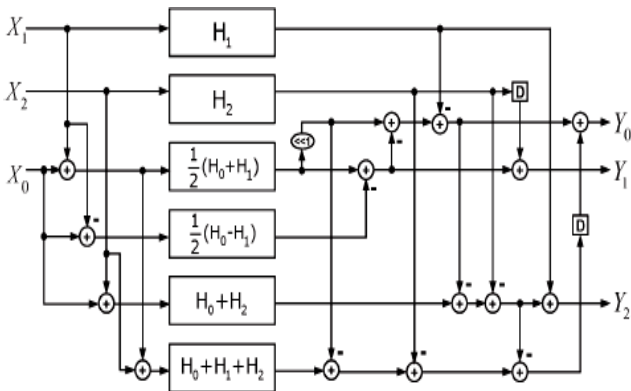


Fig.6. Implementation of the proposed 3C structure

Table 1.

Comparison between N-Tap Proposed Structures And Existing FFA Structure, Number Of Required Multipliers (R.M), Subfilter Block With Symmetric Coefficients (S.S), Number Of Increased Adders (I.A)

N mod 3	Architecture	S.S	R.M	I.A
0	FFA	2	$\frac{\square}{3} - 1$	5
	Proposed 3A	4		
1	FFA	2	$\frac{1}{2} \left(\left\lceil \frac{\square}{3} \right\rceil - 1 \right)$	3
	Proposed 3B	3		
2	FFA	2	$\frac{1}{2} \left(\left\lceil \frac{\square}{3} \right\rceil \right)$	5
	Proposed 3C	3		

5. COMPARISON RESULTS

Table 2.

Comparison of Synthesis Results for the Structures 2-Parallel (2P), 3-Parallel (3P), Proposed3a (P3A), Proposed3b (P3B), Proposed3c (P3C)

STRUCTURE	2P	3P	P3A	P3B	P3C
DEVICE UTILIZATION (%)	29	66	29	27	25
LOGIC DISTRIBUTION (%)	31	72	31	27	27

Table 3.

Comparison of Power Consumption For 2-Parallel (2P), 3-Parallel (3P)

	2P	3P
POWER(mW)	25	32

6. CONCLUSION AND FUTURE WORK

This paper has presented the new parallel linear-phase FIR structures which are highly beneficial to the symmetric convolutions of odd length. Multipliers play a major role in terms of area and power consumption in FIR implementation. Since multipliers outweigh adders in hardware cost, it is economical to replace multipliers with adders. The proposed new parallel filter blocks exploit the nature of symmetric odd coefficients and further reduce the amount of multipliers with

adders. However, the numbers of reduced multipliers increases along with the length of FIR filter whereas the number adders remain still. The reduction of multipliers by mathematical examination is yet to be done and comparison of area and power synthesis result comparison for proposed structures are yet to be provided.

7. REFERENCES

- [1] D. A. Parker and K. K. Parhi, "Low area-power parallel FIR digital filter implementations," *J.VLSI Signal Process.Syst.*, vol.17, no.1, pp.75-92, Sep.1997.
- [2] J. G. Chung and K. K. Parhi, "Frequency-spectrum based low-power parallel FIR filter design," *EURASIP J.Appl.Signal Process.*, vol.2002, no.9, pp.444-453, Jan.2002.
- [3] K. K. Parhi, *VLSI Digital Signal Processing Systems : Design & Implementation*, Newyork : Wiley,1999.
- [4] Z. -J. Mou and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering," *IEEE Trans. Signal Process.*, vol.39, no.6, pp.1322-1332, Jun.1991.
- [5] Y. C. Tsao and K. Choi, "Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm," *IEEE Trans. Very Large Scale Integr.(VLSI) Syst.*, vol.20, no.2, pp.366-371, Feb.2010.
- [6] J. I. Acha, "Computational structures for fast implementation of L-path and L-block digital filters," *IEEE Trans.Circuits Syst.*, vol.36, no.6, pp-805-812, Jun.1989.
- [7] C. Cheng and K. K. Parhi, "Hardware efficient fast parallel FIR filter structures based on iterated short convolution," *IEEE. Trans. Circuits Syst.I, Reg.Papers*, vol.51, no.8, pp.1492-1500, Aug.2004
- [8] C.Cheng and K. K. Parhi. "Further complexity reduction of parallel FIR filters," in *Proc.IEEE ISCAS*, May 2005, vol.2, pp. 1835-1838.
- [9] C.Cheng and K. K. Parhi, "Low-cost parallel FIR Structures with 2-stage parallelism," *IEEE Trans.Circuits Syst.I, Reg.Papers*, vol.54, no.2, pp.280-290, Feb.2007.
- [10] I. -S. Lin and S. K. Mithra, "Overlapped block digital filtering," *IEEE Trans.Circuits Syst.II,Analog Digit. Signal Process.*, vol.43, no.8, pp-586-596, Aug.1996.