

# Comparative Analysis of Design Methodologies for Parallel FIR Filter

Nongmaithem Lalleima  
Chanu  
Post Graduate Scholar  
DIT University  
Dhradun, India

Bidyaxmi Devi  
Tensubam  
Post Graduate Scholar  
DIT University  
Dehradun, India

Vimal Kant Pandey  
Assistant Professor  
DIT University  
Dehradun, India

## ABSTRACT

The importance of DSP systems with low power, low area and high performance appear to be increasing with no visible sign of saturation. Digital filters play a vital role in digital systems where Finite Impulse Response (FIR) filters are one of the most widely used fundamental devices. This review paper deal with the design and implementation of parallel FIR filter structure on FPGA using 4 different parallel processing methodologies with minimal cost of hardware. Since adders have no effect to the filter length and occupy less area than the multipliers, here multipliers are exchanged with adders. The paper hence describes the comparative performance analysis of traditional parallel FIR filter with respect to the FFA, transposition and symmetric convolution based parallel FIR filter with featuring the advantage of reduced hardware complexity to accurate processing with conservation of filter dynamic.

## General Terms

Fast FIR algorithm (FFA), parallel processing

## Keywords

Digital signal Processing (DSP), Finite Impulse Response (FIR), symmetric convolution, Field programmable gate array (FPGA).

## 1. INTRODUCTION

Digital Signal Processing (DSP) has a rich history and its importance is evident in diverse fields. Execution of DSP system requires high performance digital filters and its performance counts both in hardware platform employed and computation structure. Digital FIR filters are the basic and common component in many DSP systems which promotes revolutionary advances in several fields of applications such as radar, noise reduction, image and video processing, biomedical signal processing and wireless communication.

Generally, the most often used measures of efficiency of an algorithm are speed, power and area. Hence a good relation is necessary between the algorithm and technology applied for its implementation. Higher order filter demands additional arithmetic operation, hardware and power and consequently, the main goal in designing digital filter is to minimize these parameters [8], [9]. DSP systems with low power are one of the chief interests in this era and this can be exploited with the help of two techniques, namely parallel and pipelining processing. In this paper, FIR filter implementation is described with the help of 4 different techniques such as traditional pipelining, fast FIR, parallel filters by transposition and symmetric convolution. The throughput and power consumption of the original filter is reduced by parallel/block

processing with digital FIR filter. FIR filter parallel processing involves the hardware units replications of original filter which in-turn increases the effective sampling speed, since multiple outputs of parallel processing are computed by parallel in a clock period. The implementation can be ascertained either by software or hardware solutions. The paper focuses to obtained parallel FIR filters using different methodologies and its central design concept is to build filters with minimal multiplication and better performance. The paper is organized as follows: section II describes the basics of FIR filter. Section III briefly illustrates the needs and uses of pipelining and parallel processing technique, section IV deals with the above mentioned methodologies of FIR filter implementation, result and conclusion are in section V and VI respectively. All the mentioned methodologies are described with the help of 2-parallel FIR filters which are designed using MATLAB/Simulink and synthesized by Xilinx 10.1.

## 2. FINITE IMPULSE RESPONSE FILTER

The difference equation which defines the input and output signal relations for FIR filter is given as

$$y[n]=a_0x(n)+a_1x(n-1)+a_2x(n-2)+\dots+a_Nx(n-N) \quad (1)$$

where  $x(n)$  is the input signal,  $y(n)$  is the output signal,  $a_i$  is the filter coefficients and  $N$  is the filter order. The transfer function of a FIR filter is define as

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \quad (2)$$

Where  $h(n)$  is the impulse response of the filter. Figure 1 shows the 4-tap FIR filter with unit sample response  $h(n)=\{a,b,c,d\}$  and considering this filter as the platform to easily demonstrate and define the basic parallel processing using polyphase decomposition where the input sequence and the filter coefficients can be decomposed into odd and even parts, the expression of 4-tap FIR filter using equation (1) is given as

$$y(n)=ax(n)+bx(n-1)+cx(n-2)+dx(n-3)$$

The system function using equation (2) can be written as

$$\begin{aligned} H(z) &= \sum_{n=0}^{4-1} h(n)z^{-n} \\ &= h(a)+h(b)z^{-1}+h(c)z^{-2}+h(d)z^{-3} \end{aligned}$$

$$Y(z)=h(a)X(z)+h(b)z^{-1}X(z)+h(c)z^{-2}X(z)+h(d)z^{-3}X(z)$$

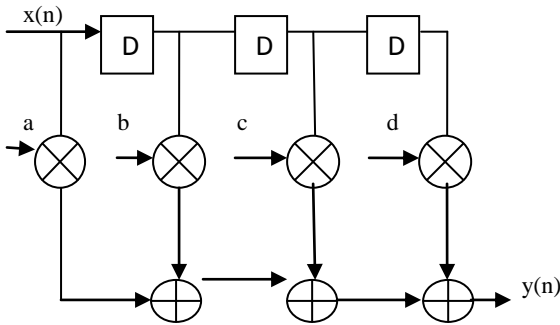


Figure 1: 4-tap FIR filter.

### 3. PIPELINING AND PARALLEL PROCESSING

Pipelining is a transformation technique used to reduced the critical path of a system by introducing latches along the data path. Pipelining also helps to either increase the speed or sample speed or to reduced the power consumption at same speed [2]. With the help of pipelining the power of the pipelined system can be reduced by a factor of  $\beta^2$  as compared with the original FIR filter and power consumption of pipelined filter is given as

$$P_{pip} = \beta^2 P_{seq}$$

where  $\beta$  is the power consumption reduction factor and  $P_{seq}$  is the power consumption of a FIR filter. Parallel processing increases the effective sampling rate by the level of parallelism thereby computing multiple outputs in a clock period that is hardware is replicated so that multiple inputs can be processed and multiple outputs can be produced at the same time. Similarly, power consumption of L-Parallel processing system also reduces power consumption by a factor of  $\beta^2$  as that of the original system.

### 4. PARALLEL FIR FILTER STRUCTURES

Parallel processing of an FIR filter replicates the hardware units of the original filter. If A is the area of the original circuit, then L-parallel circuits requires  $L \times A$  area, that is the area of the circuit increases linearly with the block size. Here, 4 structures of  $2 \times 2$  parallel FIR filters ( $L=2$ ) are designed using 4 different methods of parallel processing.

#### 4.1 Parallel FIR filter using Polyphase decomposition

Polyphase decomposition is a common method for FIR digital filter structure realization where the small sized parallel FIR are derived first and then large block sized ones can be constructed by cascading or iterating small size parallel FIR filter blocks [1],[11]. In time domain, an N-tap FIR filter can be express as

$$y(n) = \sum_{i=0}^{N-1} h(n)x(n-i), n=0,1,2,3,\dots,\infty$$

where  $x(n)$  is an infinite length input sequence and  $h(n)$  is the filter coefficient of length N. In Z-domain,

$$Y(z) = H(z)X(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \sum_{m=0}^{\infty} x(m)z^{-m} \quad (3)$$

General standardized relation of L-parallel traditional FIR filter which can be implemented using polyphase decomposition is

$$\sum_{i=0}^{L-1} Y_i(z^L)z^{-i} = \sum_{j=0}^{L-1} H_j(z^L)z^{-j} \sum_{i=0}^{L-1} X_i(z^L)z^{-i} \quad (4)$$

where  $X_i(z)$ ,  $H_j(z)$  and  $Y_i(z)$  are the poly-phase components of input, filter transfer function and output respectively and polyphase components are given as:

$$X_i(z) = \sum_{k=0}^{\infty} z^{-k} x(Lk+i), H_j(z) = \sum_{k=0}^{N/L-1} z^{-k} h(Lk+j) \text{ and } Y_i(z) = \sum_{k=0}^{\infty} z^{-k} y(Lk+i) \text{ where } i, j \text{ and } l=0,1,2,\dots,L-1.$$

For a 4-tap FIR filter, the input sequence and coefficient can be decomposed into even and odd parts as follows:

$$\begin{aligned} X(z) &= x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} \\ &= x(0) + x(2)z^{-2} + z^{-1}\{x(1) + x(3)z^{-2}\} \\ &= X_0(z^2) + z^{-1}X_1(z^2) \end{aligned}$$

where  $X_0(z^2)$  and  $X_1(z^2)$  are the Z-transforms of  $x(2k)$  and  $x(2k+1)$  that is for even and odd parts for  $0 \leq k < \infty$ . Similarly,  $H(z)$  can be decomposed as

$$H(z) = H_0(z^2) + z^{-1}H_1(z^2)$$

where  $H_0(z^2)$  and  $H_1(z^2)$  are even and odd subfilter of length  $N/2$  respectively [7]. The output sequence can be computed as

$$\begin{aligned} Y(z) &= Y_0(z^2) + z^{-1}Y_1(z^2) \\ &= \{X_0(z^2) + z^{-1}X_1(z^2)\} \{H_0(z^2) + z^{-1}H_1(z^2)\} \quad (5) \end{aligned}$$

The 2-parallel FIR filter structure consists of two filter input ( $X_0, X_1$ ), two filter coefficients ( $H_0, H_1$ ) and two outputs ( $Y_0, Y_1$ ). The above equation (5) gives the  $2 \times 2$  parallel FIR filter outputs:

$$\begin{aligned} Y_0(z^2) &= X_0(z^2)H_0(z^2) + z^{-2}X_1(z^2)H_1(z^2) \text{ and} \\ Y_1(z^2) &= X_0(z^2)H_1(z^2) + X_1(z^2)H_0(z^2) \quad (6) \end{aligned}$$

The traditional 2-parallel FIR filter is shown in the figure 2 using equation (6) which requires  $2N$  multiplications and  $2(N-1)$  additions.

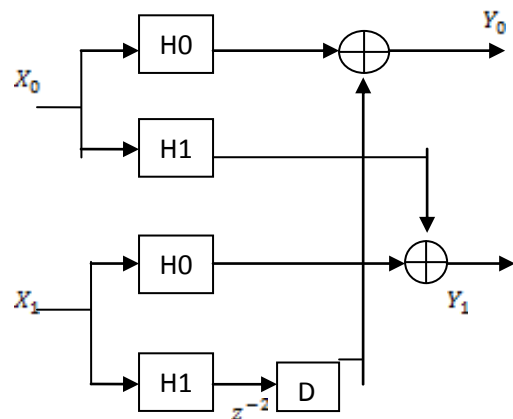


Figure 2: Traditional 2-parallel FIR filter structure.

The matrix form of equation (6) is  $Y=HX$ .

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} H_0 & z^{-2}H_1 \\ H_1 & H_0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix} \quad (7)$$

Traditional parallel structure occupies large silicon area and consumes more power, so the hardware overhead incurred by the traditional parallel processing. Due to the limitations in design area, it is more beneficial to realize parallel FIR filtering structure that consumes less area than traditional parallel FIR filtering. Hence, efficient methods such as Fast FIR parallel algorithm and symmetric convolution based FFA are employed to exploit the hardware complexity and power consumption of the system.

### 4.2 Fast FIR algorithm based parallel FIR filter

The L-parallel filter can be implemented using approximately  $(2L-1)$  subfilter blocks each of length  $N/L$ . FFA slowly diminishes the hardware complication complexity due to the increasing block size as compare to the traditional method where the resulting parallel structure using FFA requires  $(2L-N/L)$  multiplications [7],[12]. Figure 4 shows the 2-parallel fast FIR filter structure containing three subfilters of length  $N/2$  which requires  $3N/2$  multipliers and  $3(N/2-1)+4$  additions. Here, equations (6) can be rewritten as:

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

$$Y_1 = (H_0 + H_1)(X_0 + X_1) - H_0X_0 - H_1X_1 \quad (8)$$

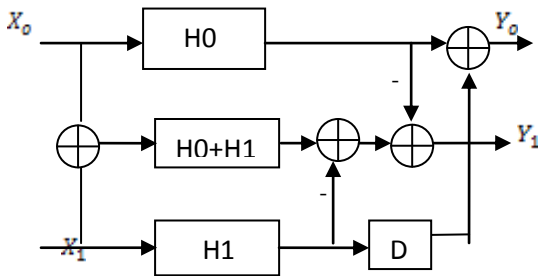


Figure 3: 2-parallel FFA implementation structure.

This filter has one preprocessing and three post-processing adders and equation (7) can also be written in matrix form as  $Y=QHPX$

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & z^{-1} \\ -1 & 1 & -1 \end{bmatrix} \text{diag} \begin{bmatrix} H_0 \\ H_0 + H_1 \\ H_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix} \quad (9)$$

Where Q and P defines the post and preprocessing matrix respectively and the notation  $\text{diag} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix}$  represents  $N \times N$  diagonal matrix [3], [6].

### 4.3 Transposition based FFA parallel filter

With the help of transposition technique, another equivalent parallel structure can be derive from any parallel FIR filter structure where the transposed architecture has the same hardware complexity but has different finite wordlength performance. The general matrix form of parallel filtering operation to derive another parallel filter using transpose operation is  $Y=HX$ , where H is an  $L \times L$  matrix. The

equivalent parallel structure can be obtained by taking transpose of the H matrix and then flipping the vector X and Y [5], [7] that is:  $Y_F = H^T X_F$ , where the flipped vectors  $X_F$  and  $Y_F$  are given as:

$$X_F = [X_{L-1} X_{L-2} \dots X_0]^T \text{ and } Y_F = [Y_{L-1} Y_{L-2} \dots Y_0]^T$$

The 2-parallel FIR filter structure shown in figure 3 can be reformulated using transposition operation and equation (7) results to:

$$\begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix} = \begin{bmatrix} H_0 & H_1 \\ z^{-2}H_1 & H_0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_0 \end{bmatrix} \quad (10)$$

The signal flow graph of 2-parallel fast FIR filter (figure 4) after transposition and its reduced complexity architecture of equation (11) is shown in figure 5 and 6 respectively. Its resultant matrix form relation is

$$Y_F = P^T H^T Q^T X_F.$$

$$\begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \text{diag} \begin{bmatrix} H_0 \\ H_0 + H_1 \\ H_1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ z^{-2} & -1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_0 \end{bmatrix} \quad (11)$$

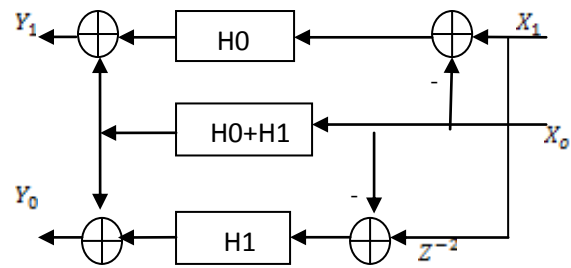


Figure 5: Transposed signal flow graph of figure 3.

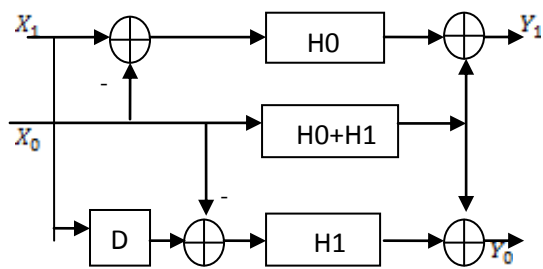


Figure 6: Transposed reduced complexity.

### 4.4 FFA based on symmetric convolution

To design symmetric convolution based fast FIR filter, symmetry of coefficients are taken in consideration to obtain more reduction in the number of multiplications which are being replaced by additions[4],[10]. The resultant filter structure provides low power, area efficient and high speed design, shown in figure 5. In this method, the poly-phase decomposition is manipulated to have sub-filters blocks with symmetric coefficients and each sub-filter block reuses  $N/2L$  multiplications that is half the number of multiplication. The filter structure has two preprocessing and four post-processing adders. The filter output equation is given as:

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

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

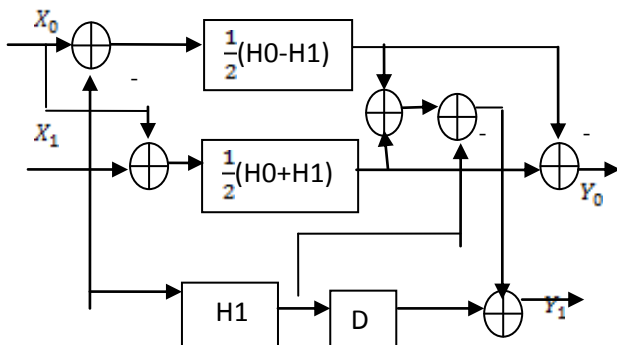


Figure 7: Symmetric convolution based 2-parallel FIR filter.

## 5. EXPERIMENTAL RESULT

The design and implementation of 2-parallel FIR filters using 4 different methodologies were successfully carried out. Simulation and synthesis of parallel filters have been accomplished on Spartan 3A/3AN series FPGA with target device Xc3s50a-5ft256. The synthesis report results are tabulated in table1 with multiplication complexity of 2-parallel filters.

Table 1: Device utilization and multiplication complexity summary of four 2- parallel FIR filter structures.

Filter structure	No. of slices out of 704	No. 4 inputs of LUTs out of 1408	Multiplication complexity of 2-parallel FIR filter
Traditional FIR	21	32	2N
FFA based FIR	17	30	3N/2
Transposition based FFA filter	21	32	3N/2
Symmetric Convolution based FFA filter	17	30	3N/2L

## 6. CONCLUSION

In this paper, 4 different parallel processing methodologies of an FIR filter have been presented. Compared to the traditional parallel processing, FFA and symmetric convolution based structures save significant number of multipliers with an expense of additional adders and exploits the hardware complexity incurred by the traditional structure. However, transposition based FFA parallel filter has different finite wordlength performance with same computational

complexity. L-parallel filter with higher order can be design using these methods to obtain a consistent quality of output.

## 7. ACKNOWLEDGEMENT

The authors gratefully acknowledge and would like to express gratitude to Mr.Vimal Kant Panday, for his guidance, support and invaluable suggestions in the course of completion of paper.

## 8. REFERENCES

- [1] Acha JI. 1989. Computational Structures for Fast Implementation of L-path and L-block digital filters. IEEE Trans.Circuit and Systems I. Vol.36, No.6, pp:805-812.
- [2] Chang TS and Jen CW. 2001. Hardware Efficient Pipelined Programmable FIR Filter Design. Computers and Digital Techniques, in IEEE proceeding. Vol.148, No.6, pp:227-232.
- [3] Chao Cheng, Keshab K Parhi. 2004. Hardware Efficient fast Parallel FIR Filter Structure Based on Iterated Short Convolution.IEEE Trans.Circuit and System I. Vol.51, No.8, pp:1492-1500.
- [4] Chao Cheng and Keshab K Parhi. 2005. Further Complexity Reduction of Parallel FIR Filters.in Proc. IEEE International Symposium on Circuit and Systems I. Vol.2, pp:1835-1838.
- [5] Chao Cheng and Keshab K Parhi. 2007. Low Cost Parallel FIR Filter Structures with 2-stage Parallelism.IEEE Trans. on Circuit System I. Vol.54, No.2, pp:280-290.
- [6] Jin-Gyum Chung, Keshab K Parhi (2002).Frequency Spectrum Based Low-area low-power Parallel FIR Filter design. EURASIP Journal on Applied Signal Processing.9:944-953.
- [7] Keshab K Parhi. 2012. VLSI Digital Signal Processing Systems Design and Impementation. In. Willey Student (ed.) Chapter 3 and chapter 9.
- [8] Lin I.S. and Mitra S.K. 1996.Overlapped Block Digital Filtering. IEEE Trans. on Circuit Systems II Analog Digital Signal Processing. Vol.43, No.8, pp:586-596.
- [9] Parkar D.A. and Parhi K. K. 1997. Low Area/Power Parallel FIR Digital Filter Implementation.J.VLSI Signal Processing and Systems. Vol.17, No.1, pp:75-92.
- [10] Yu-Chi Tsao and Ken Choi. 2012. Area Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR algorithm. IEEE Trans. on VLSI Systems. Vol.20, No.2, pp:366-371.
- [11] Zhi-Jian Mou and Pierre Duhamel. 1987. Fast FIR Filter Algorithms and Implementations. IEEE Trans. on Signal Processing.Vol.13, No.4, pp:377-384.
- [12] Zhi-Jian Mou and Pierre Duhamel. 1991. Short-Length FIR Filters and their use in fast Non- Recursive Filtering. IEEE Trans. on Signal Processing. Vol.39, No.6, pp:1322-1332.