

# Design of Low Area and High Speed Parallel FIR Digital Filter using Fast FIR Algorithm

Gomathi.S<sup>1</sup>, Dr.S.Sasikala<sup>2</sup>, D.Karpaga priya<sup>3</sup>, S.Kowsalya<sup>4</sup>, M.S.Karan Kishore Ananth<sup>5</sup>

<sup>1</sup>Assistant Professor, Department of ECE, Kongu Engineering College, India, samgomathi@gmail.com

<sup>2</sup> Associate Professor, Department of ECE, Kongu Engineering College, India, sasikalapriyaadarsan@gmail.com

<sup>3,4,5</sup> Student, Department of ECE, Kongu Engineering College, India,

## ABSTRACT

Finite Impulse Response (FIR) is thought to be one amongst the most important operations in digital signal processing. Parallel FIR filter has various applications in video compression and equalizers. Implementation of traditional parallel FIR filter offers high speed at the expense of increased hardware by the block size. This huge hardware penalty can't be tolerated in many design situation. So, it's necessary to style parallel FIR filter structures that is implemented with low area than traditional block FIR filtering structures. This paper addresses the design of higher order Parallel FIR filters using small sized filters using FFA algorithm. The Parallel Fast FIR Algorithm specially targets within the sharing of filter coefficients. The Parallel Fast FIR algorithm gives the output with delay of 13.0825 ns which is 47 you quicker than Parallel FIR Algorithm which produces the output with delay of 17.9675 ns. The Parallel Fast FIR Algorithm has been implemented in Four tap, Eight tap FIR filter and sixteen tap FIR filter yet in Parallel FIR Algorithm and Traditional FIR algorithm.

**Key words :** Finite impulse response, delay, sharing of filter coefficients and Fast FIR algorithm.

## 1. INTRODUCTION

High Speed and Low Power processors are at a greater demand. Many application uses FIR filter which operates at high frequency like video processing[1] and in multiple-input multiple-output (MIMO) systems[2][18]where high throughput and low power is mandate. Computation Sharing Multiplier provides 41% reduction in power [12].In Parallel processing technique multiple inputs are processed at a time and multiple outputs are produced. The hardware is increased block size times and thus area increases. The data processing technique loses its advantage in practical implementation as there is increase in hardware linearly with respect to the block size. Literature provides a wide variety of papers that propose the ways to implement it with lesser hardware[2][3][4-6]. Developing small size parallel FIR filters and then cascading it to build large ones is a solution[6]. L-parallel filter using approximately (2L-1) subfilter blocks can be implemented, each of which is of length N/L usingFast FIR algorithms (FFAs)[15] . There is an advantage when FFA structures are used. It overcomes the constraint of increase in the hardware implementation as the block size increases. Parallel FFA implementation offers reduction in multiplication operation

from L x N to (2N-N/L). This paper offers the design of higher order filters using poly phase decompositions by the application of FFA . The reduction in area is achieved by the implementation of FFA .Many High Level transformations have been used to design several VLSI Architectures.[6][7] [13-17].

## 2.DESIGN APPROACH FOR FIR FILTER

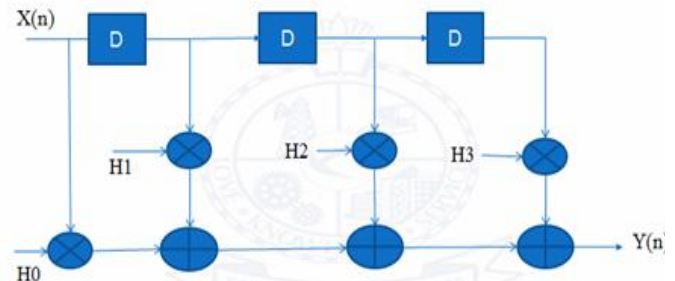
In this paper FIR filters are designed in three different forms conventional, traditional parallel filter and Parallel FFA filters. The design of all the three forms are separately given in the forthcoming sections.

### 2.1Conventional FIR Filter

Finite impulse response (FIR) filter is considered one amongst the foremost operations in digital signal processing. A linear time invariant (LTI) FIR filter is one of the basic building blocks common to most DSP systems. The output of an FIR filter is a sequence generated by convolving the sequence of the input samples with N filter coefficients. The filter expression can be described by

$$Y(n) = \sum_{k=0}^{N-1} h(k)x(n - k) \quad (1)$$

Where N is the length of the filter, h(k)denotes the kth coefficient, and x(n-k)denotes the sampled input data at time n-k.



**Figure 1:** Conventional 4 Tap FIR Filter

A 4 tap FIR Filter structure as realised by equation 2 and the structure is shown in Figure 1.It is a Single Input and Single Output (SISO) system with the output

$$Y(n) = H0*x(n) + H1*x(n-1) + H2*x(n-2) + H3*x(n-3) \quad (2)$$

**2.2 Traditional Parallel FIR Filter**

Convolution is the main operation used in FIR filter which has a high computational complexity. It makes use of a special hardware which is more suitable or increasing the computational performance. The SIS0 system must be converted into an MIMO (multiple input and multiple output) system to obtain a parallel processing structure. A traditional 4 parallel FIR filter accepts 4 inputs per clock cycle and thus the level of parallel processing  $L=4$ [19]. Block processing is applied to FIR filter to improve the throughput or to achieve low power. In previous designs of FIR filter, FIR filter involved the replication of hardware unit in the original design in its application. An area of  $L \times A$  is required by an  $L$ -parallel circuit if the area of the original circuit is  $A$ . As there is a limitation in the design area, the increase in the computational complexity incurred by parallel processing cannot be endured. Hence it is better to recognize the parallel FIR filtering structure than traditional parallel FIR filtering structure as it occupies less area.

**2.3 Formulation of Traditional Parallel FIR Filters Using Polyphase Decomposition**

An  $N$ -tap FIR filter obtained from an input sequence  $x(n)$  of infinite length sequence and the impulse sequence  $h(n)$  of length  $N$ , in  $z$ -domain as [15]

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

The  $L$ -parallel FIR filters are derived by using poly phase decomposition by decomposing  $X(z), H(z)$  and  $Y(z)$  into  $L$  sub sequences as follows :

$$X_1(z) = \sum_{k=0}^{\infty} z^{-1} x(Lk + j), i = 0, 1, \dots, L - 1 \quad (4)$$

$$H_1(z) = \sum_{k=0}^{\binom{N}{L}-1} z^{-k} h(Lk + j), j = 0, 1, \dots, L - 1 \quad (5)$$

$$Y_1(z) = \sum_{k=0}^{\infty} z^{-k} y(Lk + l), l = 0, 1, \dots, L - 1 \quad (6)$$

The  $L$  output subsequence  $y(Lk+l)$  ( $0 \leq l \leq L - 1, 0 \leq k \leq \infty$ ) can be computed using an combination of  $L$  sub filters from the  $L$  input subsequence's as

$$y_k = z^{-L} \sum_{i=k+1}^{L-1} H_i X_{L+k-i} + \sum_{i=0}^k H_i X_{k-i}, 0 \leq k \leq L - 2 \quad (7)$$

$$Y_{L-1} = \sum_{i=0}^{L-1} H_i X_{L-1-i} \quad (8)$$

For a 4 tap, the design procedure for the traditional parallel filter The input sequence  $x(z)$  is given as

$$X(z) = X_0(z^4) + z^{-1} X_1(z^4) + z^{-2} X_2(z^4) + z^{-3} X_3(z^4) \quad (9)$$

$$H(z) = H_0(z^4) + z^{-1} H_1(z^4) + z^{-2} H_2(z^4) + z^{-3} H_3(z^4) \quad (10)$$

$$Y(z) = Y_0 + z^{-1} Y_1 + z^{-2} Y_2 + z^{-3} Y_3 = X(z)H(z) = (H_0 + z^{-1} H_1 + z^{-2} H_2 + z^{-3} H_3)(X_0 + z^{-1} X_1 + z^{-2} X_2 + z^{-3} X_3) \quad (11)$$

$$= X_0 H_0 + z^{-4} X_1 H_3 + z^{-4} X_2 H_2 + z^{-4} X_3 H_1 + z^{-1} (X_0 H_1 + X_1 H_0 + z^{-4} X_3 H_2 + z^{-4} X_2 H_3) + z^{-2} (X_0 H_2 + X_1 H_1 + X_2 H_0 + z^{-4} X_3 H_3) + z^{-3} (X_1 H_2 + X_3 H_0 + X_2 H_1 + X_0 H_3)$$

$$Y_0 = X_0 H_0 + z^{-4} X_1 H_3 + z^{-4} X_2 H_2 + z^{-4} X_3 H_1$$

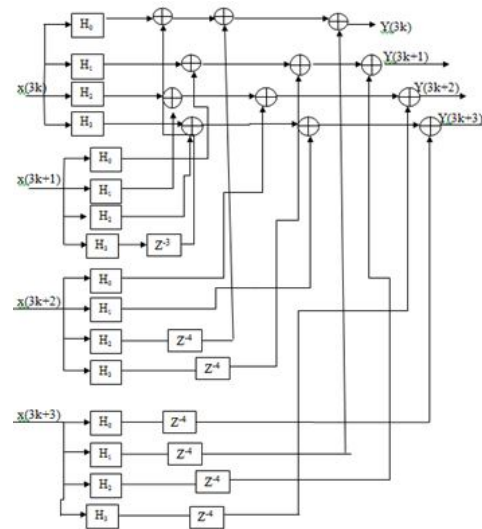
$$Y_1 = X_0 H_1 + X_1 H_0 + z^{-4} X_3 H_2 + z^{-4} X_2 H_3$$

$$Y_2 = X_0 H_2 + X_1 H_1 + X_2 H_0 + z^{-4} X_3 H_3$$

$$Y_3 = H_2 X_1 + H_0 X_3 + H_3 X_0 + H_1 X_2$$

This can also be written in matrix form as  $Y=HX$

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} H_0 & z^{-4} H_3 & z^{-4} H_2 & z^{-4} H_1 \\ H_1 & H_0 & z^{-4} H_3 & z^{-4} H_2 \\ H_2 & H_1 & H_0 & z^{-4} H_3 \\ H_3 & H_2 & H_1 & H_0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix}$$



**Figure 2:** Traditional 4 Tap Parallel FIR Filter

Figure 2 shows the resulting 4 tap traditional parallel FIR filter by polyphase decomposition, which requires  $2.25(N)$  multiplications and  $9(0.25N-1+20)$  additions.

**2.3 Parallel FIR Filters Based On Fast FIR Algorithms**

The number of samples processed in parallel in a clock cycle gets increased with the complexity of traditional block filter. So to reduce the hardware complexity, Fast FIR algorithms are developed. This Fast Filtering Algorithms (FFA's) will reduce the complexity of parallel filtering structures.  $(2N-N/L)$  multiplication is required for the parallel filtering structure. The number of multiplications are reduced with significant increase in the number of adders for higher values of  $N$ . Replacing multipliers with add operations is advantageous because adders have a smaller implementation cost than multiplier in terms of silicon area. It becomes unmanageable when the number of adders increase for higher  $L$  values. Therefore, a balance between multipliers and adders can be maintained. An  $(n \times n)$  FFA produces a FIR filtering structure which is equivalent to a parallel FIR filter of block size  $n$ . These structures produces a set of filters of length  $N/n$  consisting of  $n$  filters that are introduced during ployphase

decomposition. With the addition of some pre- and post-processing steps that are performed in junction with the filtering operations the proper filter transfer function is realised. Higher Order filters can be designed by recursively applying lower order parallel FFA. This paper contributes to the design of higher order filters with 4,8,16 tap length using the lower order FFA designed. Consider the design of 4 parallel FIR structure. The design is derived by applying 2 parallel FFA recursively. The design is obtained by following process:

For a block of size 4,

$$Y = Y_0 + z^{-1} Y_1 + z^{-2} Y_2 + z^{-3} Y_3 \tag{12}$$

$$= (X_0 + z^{-1} X_1 + z^{-2} X_2 + z^{-3} X_3) (H_0 + z^{-1} H_1 + z^{-2} H_2 + z^{-3} H_3) \tag{13}$$

Reduced complexity 4-parallel FIR filter is obtained by applying 2-parallel FFA and then applying it for second time. Equation 13 can be written as

$$Y = (X_0' + z^{-1} X_1') (H_0' + z^{-1} H_1') \tag{14}$$

where  $X_0' = X_0 + z^{-2} X_2$  ;  $X_1' = X_1 + z^{-2} X_3$

$H_0' = H_0 + z^{-2} H_2$  ;  $H_1' = H_1 + z^{-2} H_3$

When the 2-parallel FFA is applied for first time , the output equations become

$$Y = X_0' H_0' + z^{-2} X_1' H_1' + z^{-1} ((X_0' + X_1') (H_0' + H_1') - X_0' H_0' - X_1' H_1') \tag{15}$$

By applying the 2-parallel FFA again  $X_0' H_0'$  ,  $z^{-2} X_1' H_1'$  and  $(X_0' + X_1') (H_0' + H_1')$  to in the above equation we get,

$X_0' H_0'$  term can be solved as

$$X_0' H_0' = (X_0' + z^{-1} X_2') (H_0' + z^{-1} H_2') = X_0' H_0' + z^{-4} X_2' H_2' + z^{-2} ((X_0' + X_2') (H_0' + H_2') - X_0' H_0' - X_2' H_2') \tag{16}$$

$X_1' H_1'$  term can be solved as

$$X_1' H_1' = (X_1' + z^{-1} X_3') (H_1' + z^{-1} H_3') = X_1' H_1' + z^{-4} X_3' H_3' + z^{-2} ((X_1' + X_3') (H_1' + H_3') - X_1' H_1' - X_3' H_3') \tag{17}$$

$(X_0' + X_1') (H_0' + H_1')$  term can be solved as

$$= [(X_0' + X_1') + z^{-2} (X_2' + X_3')] [(H_0' + H_1') + z^{-2} (H_2' + H_3')] \tag{18}$$

$$= (X_0' + X_1') (H_0' + H_1') + z^{-4} (X_2' + X_3') (H_2' + H_3') + z^{-2} [(X_0' + X_1' + X_2' + X_3') (H_0' + H_1' + H_2' + H_3') - (X_0' + X_1') (H_0' + H_1') - (X_2' + X_3') (H_2' + H_3')] \tag{19}$$

After performing these calculation the structure is drawn as shown below in Figure 3.

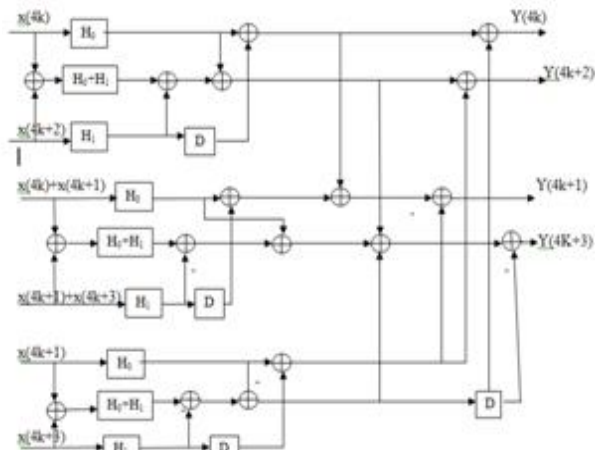


Figure 3: 4-parallel fast FIR filter by cascading two 2-parallel FFA

From the analysis it is understood that the resulting 4-parallel filtering structure requires 9N/4 multiplications and 20+9(N/4 - 1) additions for implementation. The reduced complexity 4-parallel filtering structure represents an hardware (area) savings of nearly 44% when compared to the 4N multiplications required in the traditional 4-parallel filtering structure.

**2.4 Design of 8 Tap Parallel FIR Filter Using 4 parallel FFA:**

The design of 8 tap parallel FFA filter can be done in two different ways namely cascading two four tap FFA filter and cascading four two tap FFA filter .For designing eight tap using two 4 tap FFA filter , the design equation will be as follows. It is known that the output equation can be written as

$$Y = Y_0 + z^{-1} Y_1 + z^{-2} Y_2 + z^{-3} Y_3 + z^{-4} Y_4 + z^{-5} Y_5 + z^{-6} Y_6 + z^{-7} Y_7 = (X_0 + z^{-1} X_1 + z^{-2} X_2 + z^{-3} X_3 + z^{-4} X_4 + z^{-5} X_5 + z^{-6} X_6 + z^{-7} X_7) (H_0 + z^{-1} H_1 + z^{-2} H_2 + z^{-3} H_3 + z^{-4} H_4 + z^{-5} H_5 + z^{-6} H_6 + z^{-7} H_7) \tag{20}$$

Reduction of complexity in 8 tap parallel FFA filter is done by first applying 4 tap parallel FFA and then applying it for the second time.

$$Y = (X_0' + z^{-1} X_1') (H_0' + z^{-1} H_1') \tag{21}$$

where  $X_0' = X_0 + z^{-2} X_2 + z^{-4} X_4 + z^{-6} X_6$  ;

$X_1' = X_1 + z^{-2} X_3 + z^{-4} X_5 + z^{-6} X_7$

$H_0' = H_0 + z^{-2} H_2 + z^{-4} H_4 + z^{-6} H_6$  ;

$H_1' = H_1 + z^{-2} H_3 + z^{-4} H_5 + z^{-6} H_7$

When the 4-parallel FFA is applied for first time , the output equations become

$$Y = X_0' H_0' + z^{-2} X_1' H_1' + z^{-1} ((X_0' + X_1') (H_0' + H_1') - X_0' H_0' - X_1' H_1') \tag{22}$$

By applying the FFA again  $X_0' H_0'$  ,  $z^{-2} X_1' H_1'$  and  $(X_0' + X_1') (H_0' + H_1')$  to in the above equation we get,

$X_0' H_0'$  term can be solved as

$$X_0' H_0' = (X_0' + z^{-2} X_2' + z^{-4} X_4' + z^{-6} X_6') (H_0' + z^{-2} H_2' + z^{-4} H_4' + z^{-6} H_6') = (X_0'' + z^{-1} X_1'') (H_0'' + z^{-1} H_1'') \tag{23}$$

$X_0'' = X_0' + z^{-2} X_2'$  ;  $X_1'' = X_1' + z^{-2} X_3'$

$H_0'' = H_0' + z^{-2} H_2'$  ;  $H_1'' = H_1' + z^{-2} H_3'$

$$Y = X_0'' H_0'' + z^{-2} X_1'' H_1'' + z^{-1} ((X_0'' + X_1'') (H_0'' + H_1'') - X_0'' H_0'' - X_1'' H_1'') \tag{24}$$

$X_0'' H_0''$  term can be solved as

$$X_0'' H_0'' = (X_0'' + z^{-2} X_2'') (H_0'' + z^{-2} H_2'') = X_0'' H_0'' + z^{-4} X_2'' H_2'' + z^{-2} ((X_0'' + X_2'') (H_0'' + H_2'') - X_0'' H_0'' - X_2'' H_2'') \tag{25}$$

$X_1'' H_1''$  term can be solved as

$$X_1'' H_1'' = (X_1'' + z^{-2} X_3'') (H_1'' + z^{-2} H_3'') = X_1'' H_1'' + z^{-4} X_3'' H_3'' + z^{-2} ((X_1'' + X_3'') (H_1'' + H_3'') - X_1'' H_1'' - X_3'' H_3'') \tag{26}$$

$(X_0'' + X_1'') (H_0'' + H_1'')$  term can be solved as

$$= [(X_0'' + X_1'') + z^{-2} (X_2'' + X_3'')] [(H_0'' + H_1'') + z^{-2} (H_2'' + H_3'')] = (X_0'' + X_1'') (H_0'' + H_1'') + z^{-4} (X_2'' + X_3'') (H_2'' + H_3'') + z^{-2} [(X_0'' + X_1'' + X_2'' + X_3'') (H_0'' + H_1'' + H_2'' + H_3'') - (X_0'' + X_1'') (H_0'' + H_1'') - (X_2'' + X_3'') (H_2'' + H_3'')] \tag{27}$$

After performing these steps the structure is drawn as shown below in Figure 4.



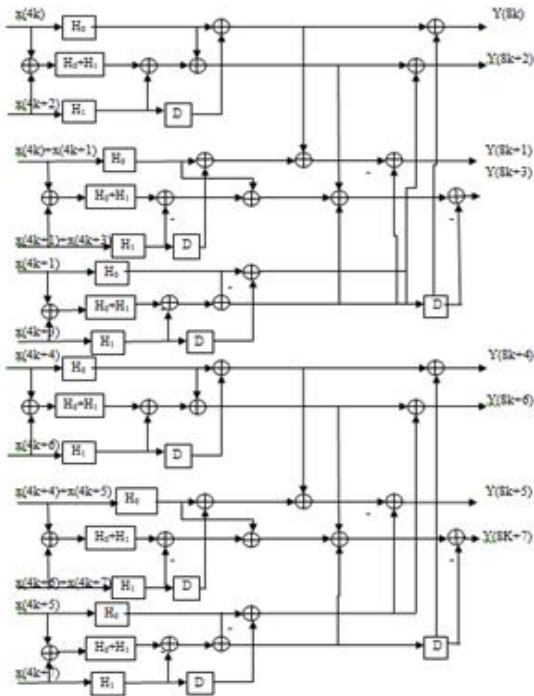


Figure 4: Eight Parallel Fast FIR filter by cascading two 4-tap FFA filter

2.5 Design of 8 Tap Parallel FIR Filter Using 2 parallel FFA:

For designing eight tap using 2 tap FFA filter , the design equation will be as follows:

$$Y = Y_0 + z^{-1} Y_1 + z^{-2} Y_2 + z^{-3} Y_3 + z^{-4} Y_4 + z^{-5} Y_5 + z^{-6} Y_6 + z^{-7} Y_7$$

$$= (X_0 + z^{-1} X_1 + z^{-2} X_2 + z^{-3} X_3 + z^{-4} X_4 + z^{-5} X_5 + z^{-6} X_6 + z^{-7} X_7) (H_0 + z^{-1} H_1 + z^{-2} H_2 + z^{-3} H_3 + z^{-4} H_4 + z^{-5} H_5 + z^{-6} H_6 + z^{-7} H_7)$$

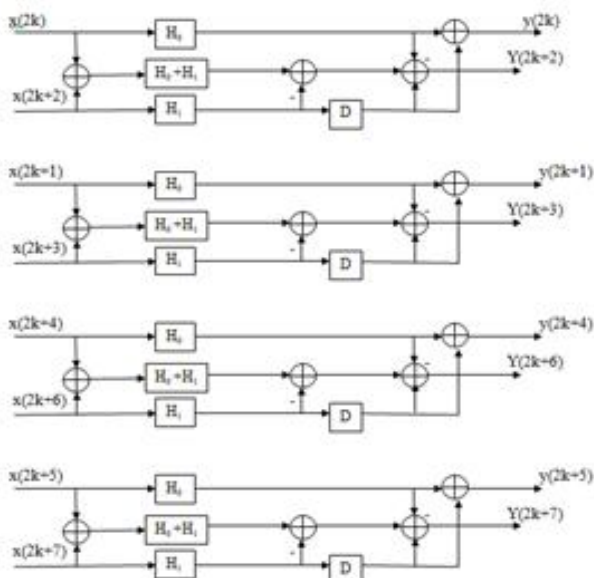


Figure 5: Eight Parallel Fast FIR filter by cascading four 2-tap FFA filter

Reduction of complexity in 8 tap parallel FFA filter is done by first applying 2 tap parallel FFA and then applying it for the necessary times. Here in this design 2 FFA will be applied four times to obtain a eight FFA architecture. By applying the 2-parallel FFA again  $X_0 + H_0$  ,  $z^{-1} X_1 + H_1$  and  $(X_0 + X_1) (H_0 + H_1)$  in the above equation the structure for 8 tap parallel FIR filter using 2 FFA is obtained as in Figure 5.

3. SIMULATION RESULTS

The conventional, traditional parallel and parallel FFA designs for various order of the filter 2,3,4,8,16 are verilog coded and simulated in Modelsim. The simulations results for 4 tap all forms of FIR filter designed is shown in figure 6. The Figure 6 shows the output of the Four Tap Conventional FIR filter.

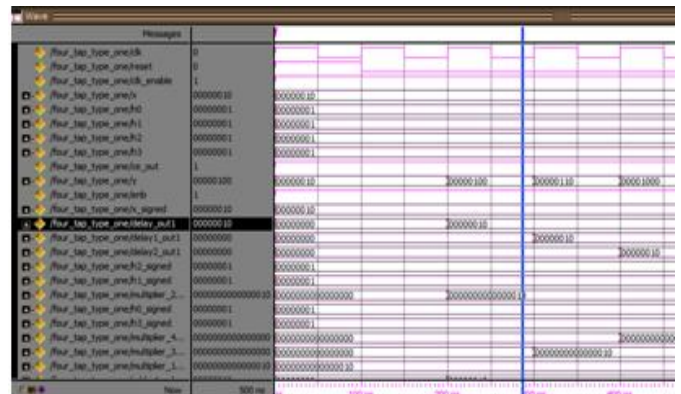


Figure 6: Output of Conventional 4 tap Filter

When the input is  $x=00000010$  and the filter coefficients are  $h_0=00000001$ ;  $h_1=00000001$ ;  $h_2=00000001$ ;  $h_3=00000001$ , the output is  $y=00000100$ . These outputs are produced in the fifth clock cycle by traditional FIR filter.

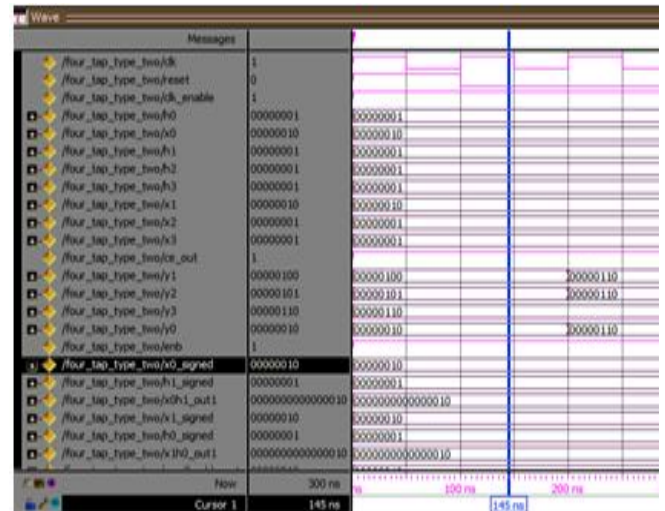


Figure 7: Output of Traditional 4 Parallel Filter

Figure 7 shows the output of traditional parallel FIR filter of order 4. When the inputs are  $x_0=00000010$ ;  $x_1=00000010$ ;  $x_2=00000001$ ;  $x_3=00000001$  and the filter coefficients are

$h_0=00000001$ ;  $h_1=00000001$ ;  $h_2=00000001$ ;  $h_3=00000001$ , the outputs are  $y_0=0000010$ ;  $y_1=00000100$ ;  $y_2=00000101$ ;  $y_3=00000110$ . These outputs are produced in the third clock cycle by parallel FIR filter.

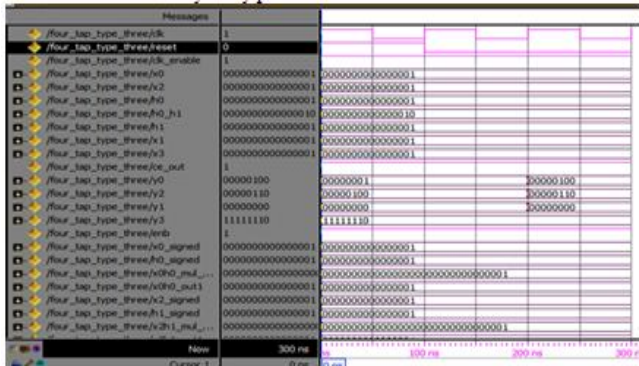


Figure 8: Output of 4 Tap Parallel FFA FIR Filter

Figure 8 reveals the output of proposed parallel FFA filter. When the inputs are  $x_0=0000000000000001$ ;  $x_1=0000000000000001$ ;  $x_2=0000000000000001$ ;  $x_3=0000000000000001$  and the filter coefficients are  $h_0=0000000000000001$ ;  $h_1=0000000000000001$ , the outputs are  $y_0=00000100$ ;  $y_1=00000000$ ;  $y_2=00000110$ ;  $y_3=11111110$ . These outputs are produced in the third clock cycle by parallel Fast FIR Algorithm.

#### 4. PERFORMANCE ANALYSIS OF THE DESIGNED FIR FILTERS

The designed FIR filters of different order of all three forms are implemented in Xilinx which belongs to the family of Spartan 3E with package PQ208 of speed grade -5. Table 1 represents the comparison between the conventional FIR filter, Traditional parallel FIR filter and Parallel Fast FIR filter algorithm in terms slices, LUTs and delay time.

Table 1 : Hardware Utilisation Details

Parameters	2 Tap Filter			3 Tap Filter			4 Tap Filter			8 Tap Filter			16 Tap Filter		
	CF	TPF	FFA	CF	TPF	FFA	CF	TPF	FFA	CF	TPF	FFA	CF	TPF	FFA
Adders	1	2	2	2	10	6	3	12	6	7	56	12	15	240	80
Number of Slice Registers	20	18	7	281	181	17	584	453	30	600	450	362	712	547	428
No of LUT's	36	31	12	500	324	30	806	613	46	1034	964	640	1112	1040	823
Maximum Combinational Path Delay(ns)	13.99	11.45	10.82	21.03	18.07	13.03	31.33	20.25	15.12	31.12	22.10	13.41	32.56	29.22	23.12

The results of traditional FIR filter , parallel FIR filter and parallel Fast FIR filter are compared based on the number of slices,4 input LUTs and maximum combinational delay. It is observed that on an average the number of LUT's is decreased to 55.2% and Combinational delay is reduced to 65.88%.

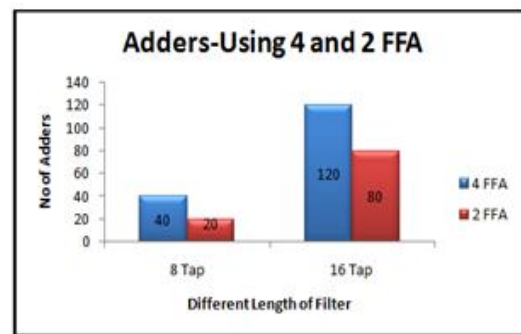
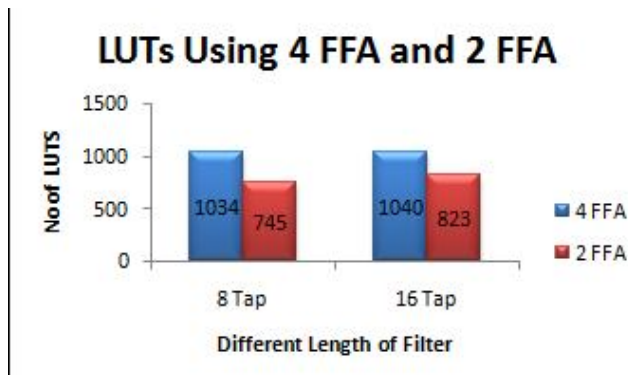


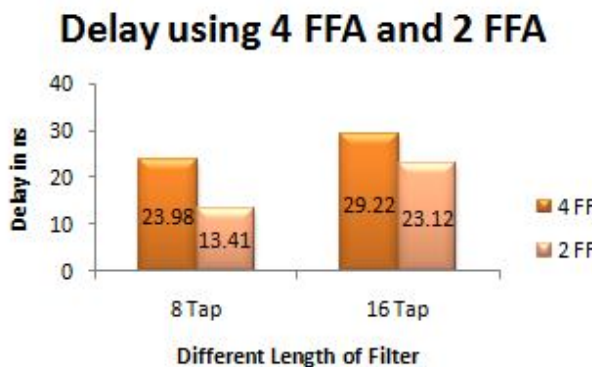
Figure 9: Comparison of Adders using 4 and 2 FFA

Figure 9 shows the results of design of 8 tap and 16 tap using lower order FFA. It is seen that there is reduction of 20% in delay and 30% in slices. Here it is better to prefer eight tap using 2 FFA than 4 FFA because it has better reduction in delay and the slices compared to others.



**Figure 10:** Comparison of Area using 4 and 2 FFA

Figure 10 reveals the comparison of area for the design of 8 and 16 tap FIR filter using 4 FFA and 2 FFA. It is seen that there is reduction of 28% in area using 2 FFA in 8 Tap and 21% using 2 FFA in 16 Tap. Thus 2 FFA can recursively be applied to design higher order filters.



**Figure 11:** Comparison of Delay using 4 FFA and 2 FFA

Figure 11 reveals the comparison of delay for the design of 8 and 16 tap FIR filter using 4 FFA and 2 FFA. It is seen that there is reduction of 25% in delay on an average using 2 FFA in 8 Tap and 16 Tap. Thus 2 FFA can recursively be applied to design higher order filters.

## 5. CONCLUSION

This paper addressed the usage of Parallel Fast FIR algorithm (FFA) for various filter orders. It is observed that on an average of 42.39% of reduction in the hardware was achieved. The Parallel Fast FIR filter shows delay reduction of about 34.4275% when compared to Parallel FIR filter. Thus high speed and low area can be achieved if FFA is implemented for different applications. In future, the folding

architecture can be implemented for the reduction of adders in Parallel Fast FIR algorithm.

## REFERENCES

1. B. K. Mohanty and P. K. Meher, "A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications," *IEEE Trans. VLSI Syst.*, vol. 24, no. 2, pp. 444-452, 2016.  
<https://doi.org/10.1109/TVLSI.2015.2412556>
2. S. Mula, V. C. Gogineni, and A. S. Dhar, "Algorithm and architecture design of adaptive filters with error nonlinearities," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 9, pp. 2588-2601, 2017.
3. S. Mula, V. C. Gogineni, and A. S. Dhar, "Algorithm and VLSI architecture design of proportionate-type LMS adaptive filters for sparse system identification," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 9, pp. 1750-1762, 2018.
4. B. Y. Kong and I. C. Park, "FIR filter synthesis based on interleaved processing of coefficient generation and multiplier-block synthesis," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 31, no. 8, pp. 1169-1179, 2017.
5. M. Aktan, A. Yurdakul, and G. Dizinda, "An algorithm for the design of low-power hardware-efficient FIR filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 7, pp. 1536-1545, 2015.
6. Gomathi Swaminathan, G. Murugesan, S. Sasikala, L. Murali, "A novel implementation of combined systolic and folded architectures for adaptive filters in FPGA," *Microprocessors and Microsystems*, vol. 74, 2020.
7. Gomathi Swaminathan, Murugesan Govindaswamy and Sasikala Subramaniyam, "Performance Analysis of Karatsuba Vedic Multiplier and Computation Sharing Multiplier in the Adaptive Filter Design," *International Journal of Innovative Technology and Exploring Engineering*, volume 9, no. 2, pp. 4425-4429, Dec 2019.
8. Y. Xing, X. Yindong, W. Houjun, G. Guangkun, and Y. Linglong, "Synchronization method of multiple multiplexed DACs," in *Proc. 13th IEEE International Conference on Electron. Meas. Instrum.*, 17-23, October 2017.  
<https://doi.org/10.1109/ICEMI.2017.8265907>
9. Y. Park and K. A. Remley, "Two-stage correction for wideband wireless signal generators with time-interleaved digital-to-analog converters," in *Proc. 83rd ARFTG Microw. Meas. Conf.*, 1-4, June 2016.
10. L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Design of lowpower multiple constant multiplications using low-complexity minimum depth operations" in *Proc. Great Lakes Symp. VLSI*, 79-84, 2017.
11. E. Kyritsis and K. Pekmestzi, "Hardware Efficient Fast FIR Filter Based on Karatsuba Algorithm" in *Proc of IEEE International Conference on Modern Circuits and systems technologies (MOCAS)*, 1-4, 2016.
12. Kaushik Roy, Jongsun Park, Woopyo Jeong, Hunsoo Choo, H. Mahmoodi Meimand and Yongtao Wang, "High Performance and Low Power FIR filter Design Based on Sharing Multiplication," *ISLPED*.

13. **Gomathi, S, Murugesan, G & Jayaprintha, Design Of Systolic Architecture For Various Adaptive Filters For Noise Cancellation**, in *Proc.of 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015,IEEE Explorer  
DOI: 10.1109/ICSCN.2015.7219907.
14. **Gomathi, S, Murugesan, G, Sasikala, S & Chitra, M, Area Efficient Implementation of Adaptive filters using High Level Transformation**, in *Proc of International Conference on Intelligent Computing and Communication for Smart World (I2C2SW)*,2018,IEEE Explorer,  
DOI: 10.1109/I2C2SW45816.2018.8997199
15. S. Sasikala, G. Murugesan. **Efficient digit serial architecture for sign least mean square adaptive filter for denoising of artifacts in ECG signals**. *International Journal of Biomedical Engineering and Technology* 23 (2/3/4) (2017) 335–344,
16. Sasikala S, Gomathi S, Naveen kumar D, Praveenraaj R K, Priyadharshini B. **FPGA Implementation of Adaptive LMS Lattice Filter**. *Bioscience Bitechology Research Communications, Special Issue*, Volume 13, No.3, pp. 121-125(2020).
17. C. Sathya, S. Sasikala, G. Murugesan, **Denoising ECG signal using combination of ENSLMS and ZALMS algorithms**, *Proceedings of 3rd International Conference on Signal Processing, Communication and Networking*, August 2015, DOI: 10.1109/ICSCN.2015.7219911 .
18. M.Chitra, S.Sasikala, S.Gomathi, A.Neetheswaran, C.Reetha. **Design of Cascaded Adaptive Filter for ECG Denoising Applications**. *International Journal of Emerging Trends in Engineering Research*, Vol. 8, No. 5, May 2020.  
<https://doi.org/10.30534/ijeter/2020/61852020>
19. Keshab.K.Parhi, *VLSI Digital Signal Processing*, Wiley Publications, (Reprint:2008)