IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 1, Ver. II (Jan. 2014), PP 39-46
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A Novel Fast FIR Algorithm for Area-Efficient Parallel FIR Digital Filter Structures Utilizes Symmetric Convolutions

Pavithra.D[1], Ashok Kumar.M[2]

[1]*(PG Scholar in Communication Systems, Adhiyamaan College Of Engineering, Hosur,India)*
[2]*(AP in Electronics And Communication Engineering, Adhiyamaan College Of Engineering, Hosur,India)*

Abstract: *Based on fast finite-impulse response (FIR) algorithms (FFAs), this paper proposes new parallel FIR filter structures, which are beneficial to symmetric coefficients in terms of the hardware cost, under the condition that the number of taps is a multiple of 2 or 3. The proposed parallel FIR structures exploit the inherent nature of symmetric coefficients reducing half the number of multipliers in sub filter section at the expense of additional adders in preprocessing and post processing blocks. Exchanging multipliers with adders is advantageous because adders weigh less than multipliers in terms of silicon area; in addition, the overhead from the additional adders in preprocessing and post processing blocks stay fixed and do not increase along with the length of the FIR filter, whereas the number of reduced multipliers increases along with the length of the FIR filter. For example, for a four-parallel 72-tap filter, the proposed structure saves 27 multipliers at the expense of 11 adders, whereas for a four-parallel 576-tap filter, the proposed structure saves 216 multipliers at the expense of 11 adders still. Overall, the proposed parallel FIR structures can lead to significant hardware savings for symmetric convolutions from the existing FFA parallel FIR filter, especially when the length of the filter is large*

Keywords: *FIR (Finite Impulse Response), FFA, Adder, Mulitplier*

I. INTRODUCTION

1.1 GENERAL

Finite-impulse response (FIR) digital filters are one of the most widely used fundamental devices performed in DSP systems, ranging from wireless communications to video and image processing. However, in both categories of method, when it comes to symmetric convolutions, the symmetry of coefficients has not been taken into consideration for the design of structures yet, which can lead to a significant saving in hardware cost. In this paper, we provide new parallel FIR filter structures based on FFA consisting of advantageous poly phase decompositions, which can reduce amounts of multiplications in the sub filter section by exploiting the inherent nature of the symmetric coefficients, compared to the existing FFA fast parallel FIR filter structure. Pipelining shortens the critical path by interleaving pipelining latches along the data path, at the price of increasing the number of latches and the system latency, whereas parallel processing increase the sampling rate by replicating hardware so that multiple inputs can be processed in parallel and multiple outputs are generated at the same time, at the expense of increased area. Both techniques can reduce the power consumption by lowering the supply voltage, where the sampling speed does not increase. In this paper, parallel processing in the digital FIR filter will be discussed. Due to its linear increase in the hardware implementation cost brought by the increase of the block size, the parallel processing technique loses its advantage in practical implementation.

1.2. OBJECTIVE

FFA structures successfully break the constraint that the hardware implementation cost of a parallel FIR filter has a linear increase along with the block size L. It reduces the required number of multipliers to (2N-N/L) from N x L. The fast linear convolution is utilized to develop the small-sized filtering structures and then a long convolution is decomposed into several short convolutions, larger block-sized filtering structures can be constructed through iterations of the small-sized filtering structures. We provide new parallel FIR filter structures based on FFA consisting of advantageous poly phase decompositions, which can reduce amounts of multiplications in the sub filter section by exploiting the inherent nature of the symmetric coefficients, compared to the existing FFA fast parallel FIR filter structure.

1.3 LITERATURES

Traditional FIR filter structure and FFA based FIR filter structure and symmetric convolution based FFA FIR filter is designed for 2-parallel filter (2*2). These entire filter structures are designed based on Carry Save Adder (CSA) and Ripple Carry Adder (RCA).This paper presents an iterated short convolution (ISC) algorithm, based on the mixed radix algorithm and fast convolution algorithm. This 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. Due to its linear increase in the hardware implementation cost brought by the increase of the block size L, the parallel processing technique loses its advantage in practical implementation. There have been a few papers. proposing ways to reduce the complexity of the parallel FIR filter in the past [1]–[9]. In [1]–[4], polyphase decomposition is mainly manipulated, where the small-sized parallel FIR filter structures are derived first and then the larger block-sized ones can be constructed by cascading or iterating small-sized parallel FIR filtering blocks. Fast FIR algorithms (FFAs) introduced in [1]–[3] shows that it can implement a L-parallel filter using approximately 2L-1 subfilter blocks, each of which is of length N/L. FFA structures successfully break the constraint that the hardware implementation cost of a parallel FIR filter has a linear increase along with the block size. It reduces the required number of multipliers to 2N - N/L, from LXN. In [5]–[9], the fast linear convolution is utilized to develop the small-sized filtering structures and then a 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.

1.5 PROPOSED SYSTEM

This paper proposes new parallel FIR filter structures, which are beneficial to symmetric coefficients in terms of the hardware cost, under the condition that the number of taps is a multiple of 2 or 3. The proposed parallel FIR structures exploit the inherent nature of symmetric coefficients reducing half the number of multipliers in sub filter section at the expense of additional adders in preprocessing and post processing blocks. Exchanging multipliers adders is advantageous because adders weigh less than multipliers in terms of silicon area; in addition, the overhead from the additional adders in preprocessing and post processing blocks stay fixed and do not increase along with the length of the FIR filter, whereas the number of reduced multipliers increases along with the length of the FIR filter.

II. Parallel Fir Digital Filter Structures For Symmetric Convolutions Based On Fast Fir Algorithm

2.1 DESCRIPTION ABOUT MODULES

These algorithms usually first derive smaller length fast parallel filters and then cascade or iterate them to design parallel finite-impulse response (FIR) filters with long block sizes. An approach to increase the throughput of FIR filters with reduced complexity hardware was presented. This approach starts with the short convolution algorithms, which are transposed to obtain computationally efficient parallel filter structures. Parallel FIR filters are implemented by using poly phase decomposition and fast FIR algorithms (FFA). The FFAs are iterated to get fast parallel FIR Algorithms for larger block sizes. Although and the small-sized parallel filter structures are computationally efficient, the number of required delay elements increases with the increase of the level of parallelism. However, the transpose of the linear convolution structure in is an optimal parallel FIR filter structure in terms of the required delay elements. While the Toeplitz-matrix factorization procedure in buries additional delay elements inside the diagonal sub filter matrix and the algorithm in places additional delay elements in the post addition matrix, parallel FIR filtering structure based on the transpose of the linear convolution structure requires no additional delays inside the convolution matrix. Furthermore, the positions of the delay elements in this transposed linear convolution structure are nicely placed and thus this structure is more regular. a set of fast block filtering algorithms are derived based on fast short-length linear convolution algorithms to realize the parallel processing of sub filters, However, when the convolution length increases, the number of additions increases dramatically, which leads to complex pre addition and post addition matrices that are not practical for hardware implementation. Therefore, if we could use fast convolution algorithms to decompose the convolution matrix with simple pre addition and post addition matrices, we can get computationally efficient parallel FIR filter with reduced number of required delay elements. Fortunately, we can use the mixed radix algorithm. Which decomposes the convolution matrix with tensor product into two short convolutions this algorithm is combined with fast two and three point convolution algorithms to obtain a general iterated short convolution algorithm (ISCA) Although fast convolution of any length can be derived from Cook–Toom algorithm or Wino grad algorithm their perdition or post addition matrices may contain elements not in the set , which makes them not suitable for hardware implementation of iterated convolution algorithm. However, in both categories of method, when it comes to symmetric convolutions, the symmetry of coefficients has not been taken into consideration for the design of structures yet, which can lead to a significant saving in hardware cost. In this paper, we provide new parallel FIR filter structures based on FFA consisting of advantageous polyphone decompositions, which can reduce amounts of multiplications in the sub filter section by exploiting the inherent nature of the symmetric coefficients, compared to the existing FFA fast parallel FIR filter structure To utilize the symmetry of coefficients, the main idea behind the proposed structures is actually pretty intuitive, to manipulate the polyphone decomposition to earn as many sub filter blocks as possible which contain symmetric coefficients so that half the number of multiplications in the single sub filter block can be reused for the multiplications of whole taps, which is similar to the fact that a set of symmetric coefficients would only require half the filter length of multiplications in a single FIR filter.

2.2 TWO-PARALLEL FIR FILTER:

The implementation of bellow equation will require three FIR subfilter blocks of length N/2, one preprocessing and three post processing adders, and 3N/2 multipliers and 3(N/2-1)+4 adders, which reduces approximately one fourth over the traditional two-parallel filter hardware cost from bellow equation. The two-parallel L=4 FIR filter implementation using FFA obtained from bellow equation is shown in Fig. 1

$$Y_0=H_0 X_0 + Z^{-2} H_1X_1$$

$$Y_1= (H_0X_0) (X_0+X_1) - H_0X_0-H_1X_1$$

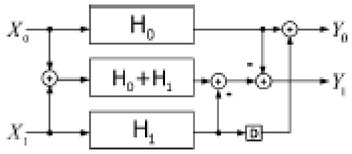


Fig. 1. Two Parallel FIR Filter

2.3 THREE-PARALLEL FIR FILTER:

The Three-parallel FIR filter hardware implementation of requires six length-N/3 FIR sub filter 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.

$$Y_0=H_0 X_0 - Z^{-3} H_2X_2+Z^{-2} x [(H_1+H_2) (X_1+X_2)-H_1X_1]$$

$$Y_1= [(H_0+H_1) (X_0+X_1) – H_1X_1] - (H_0X_0-(H_0X_0-Z^{-3}H_2X_2)$$

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

The implementation obtained from above equation and it is shown in Fig. 2.

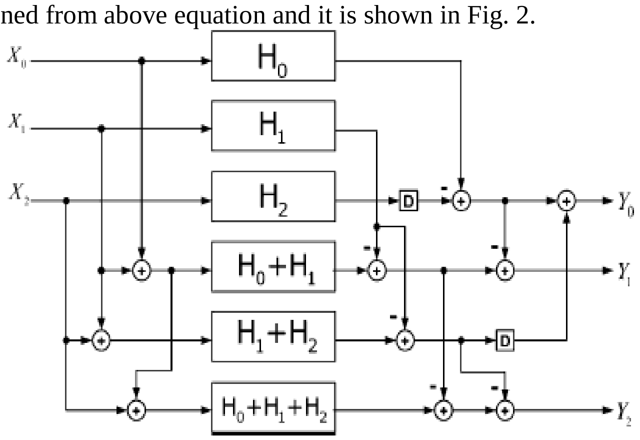


Fig 2. Three-parallel FIR filter

2.4 PROPOSED TWO-PARALLEL FIR FILTER:

Proposed Two-Parallel FIR Filter is derived by bellow equation and the architecture is shown in Fig 3.

$$Y_0= \{1/2 [(H_0 + H_1) (X_0+X_1) + (H_0+H_1) (X_0 – X_1) - H_1X_1\} + Z^{-2}H_1X_1$$

$$Y_1= ½ [(H_0+H_1) (X_0+X_1) – (H_0 - H_1) (X_0 – X_1)]$$

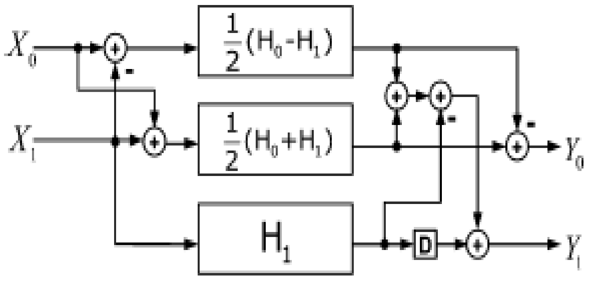


Fig 3. Proposed Two-Parallel FIR Filter

2.5 PROPOSED THREE-PARALLEL FIR FILTER:

A three-parallel FIR filter can also be written as bellow equation. Fig. 4 shows implementation of the proposed three-parallel FIR filter. When the number of symmetric coefficients N is the multiple of 3, the proposed three-parallel FIR filter structure presented based on bellow equation and enables four sub filter blocks with symmetric coefficients in total, whereas the existing FFA parallel FIR filter structure has only two ones out of six sub filter blocks.

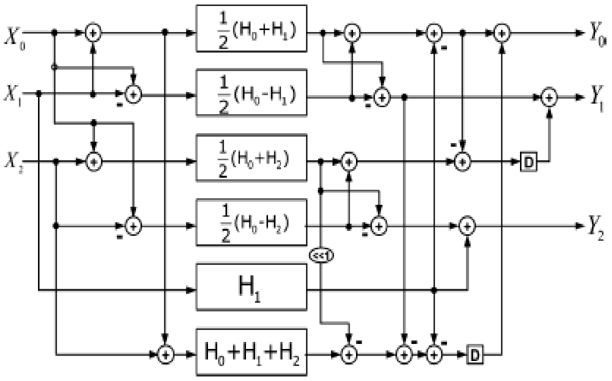


Fig 4. Proposed Three-Parallel FIR Filter

$$Y_0=1/2\ [(H_0+H_1)(X_0+X_1) + (H_0–H_1)\ (X_0-X_1)] – H_1X_1 + Z^{-3}\{(H_0+H_1+H_2)\ (X_0+X_1+X_2)-(H_1+H_2)(\ X_0+X_2) - 1/2\ [(H_0 + H_1)(X_0+X_1) + (H_0 – H_1)(X_0-X_1) – H_1X_1\}$$

$$Y_1= ½\ [(H_0+H_1)\ (X_0+X_1) – (H_0 - H_1)(X_0 - X_2)]+Z^{-3}\{1/2[(H_0+H_2)\ (X_0+X_2) + (H_0-H_2)(X_0 –X_2)] -1/2\ [(H_0+H_1)(X_0+X_1)+(H_0-H_1)(X_0-X_1)]+H_1X_1\}$$

$$Y_2= ½\ [(H_0 +H_2)\ (X_0+ X_2)] - [(H_0 – H_2)\ (X_0-X_2) + H_1X_1$$

2.6 PROPOSED FOUR-PARALLEL FIR FILTER:

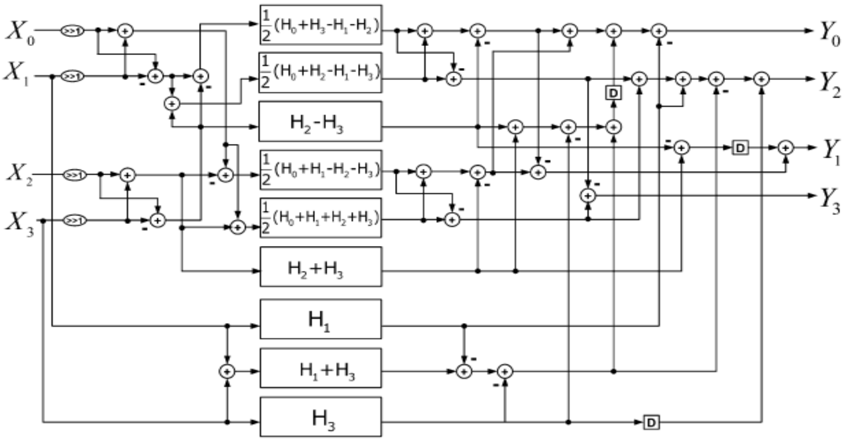

X_0
X_1
X_2
X_3
>>1
$\frac{1}{2}(H_0+H_3-H_1-H_2)$
$\frac{1}{2}(H_0+H_2-H_1-H_3)$
H_2-H_3
$\frac{1}{2}(H_0+H_1-H_2-H_3)$
$\frac{1}{2}(H_0+H_1+H_2+H_3)$
H_2+H_3
H_1
H_1+H_3
H_3
D
Y_0
Y_2
Y_1
Y_3

Fig 5. Proposed Four-Parallel FIR Filter

2.7 ALGORITHM:

Consider an N-tap FIR filter which can be expressed in the general form as

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

Where {x (n)} is an infinite-length input sequence and {h(i)}are the length-N FIR filter coefficients. Then, the traditional L- parallel FIR filter can be derived using polyphase decomposition as

$$\sum_{p=0}^{L-1} Y_p(z^L)z^{-p} = \sum_{g=0}^{L-1} X(z^L)z^{-q} \sum_{r=0}^{L-1} X(z^L)z^{-r}$$

From this FIR filtering equation, it shows that the traditional FIR filter will require L2-FIR sub filter blocks of length N/L for implementation. By the similar approach, a three-parallel FIR filter using FFA can be expressed as

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

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

$$Y_2=[(H_0+H_1+H_2)(X_0+X_1+X_2)]–[(H_0+H_1)(X_0+X_1)-H_1X_1]-[(H_1+H_2)(X_1+X_2)- H_1X_1]……………….(2)$$

The hardware implementation of above equation requires six length N/3 FIR sub filter blocks, three preprocessing and seven post processing adders, and three N multipliers and 2N+1 adders, which has reduced approximately one third over the traditional three-parallel filter hardware cost.

III. FAST FIR ALGORITHM

3.1 FIR WINDOW FILTERS:

The FIR filters Rectangular, Bartlett, Henning, Hamming, Blackman, Kaiser, and Dolph-Chebyshev are all Window FIR filters. The name "Window" comes from the fact that these filters are created by scaling a sinc (SIN(X)/X) pattern with a window to produce the desired frequency effect. All Filter Solutions FIR Window filters create the same sinc pattern, which is the Fourier Transform of a square pulse. The FIR selection then determines the window to use to scale the sinc pattern and thereby create the desired filter. If you look closely at the windows shown below, they are all similar, but unique. The Kaiser and Dolph-Chebyshev are adjustable. It is the uniqueness of each window that creates a unique FIR Window filter.

3.2 REMEZ FILTER

The Remez Filter is by far the most computationally intensive FIR filter offered by Filter Solutions. It is not a window filter, but is generated with iterative error-reducing algorithms designed to reduce the pass band error. In addition to allowing stop band ratio and frequency definition, the Remez filter allows the "Ripple Ratio" to be defined as well. The ripple ratio is defined to be the arithmetic pass band ripple divided by twice the arithmetic stop band ripple. The Remez filter always contains an odd number of taps. If an even number is entered, Filter Solutions will reduce the tap count by one, and create the filter with an odd number of taps.

3.3 RAISED COSINE FILTER

The ideal raised cosine filter frequency response consists of unity gain at low frequencies, a raised cosine function in the middle, and total attenuation at thigh frequencies. The width of the middle frequencies are defined by the roll off factor constant Alpha, (0<Altpha<1). In Filter Solutions, the pass band frequency is defined as the 50% signal attenuation point. When the pass band frequency of a raised cosine filter is set to half the sample rate, then the impulse response Nyquist's first criteria is satisfied in that the impulse response is zero for T = NTs, where N is an integer, and T is the sample rate.

3.4 FAST PARALLEL FIR ALGORITHMS:

Parallel FIR filters with long block sizes can be designed by cascading smaller length fast parallel filters an m-parallel FFA can be cascaded with an n-parallel FFA to produce an -parallel filtering structure. The set of FIR filters resulting from the application of the m-parallel FFA can be further decomposed, one at a time, by the application of the n-parallel FFA. The resulting set of filters will be of length. when cascading the FFAs, it is important to keep track of both the number of multiplications and the number of additions required for the filtering structure.

IV. Simulation Implementation

4.1 HARDWARE DESCRIPTION LANGUAGE (HDL):

Two very popular HDLs are VHDL and VERILOG. In this text we use VHDL. VHDL is an acronym for VHSIC hardware description language; VHSIC in turn is an acronym for very high speed integrated circuit. The development of VHDL was funded by the U.S. Department of Defense (DoD) in the early 1980s. The syntax of VHDL is similar to that of programming language ADA; however, it has some significant differences from ADA. We present the important concepts of VHDL, especially the ones that are used in digital circuit design. VHDL can provide unambiguous representation of a design at different levels of abstraction as shown. Modern CAD (computer-aided design) tools can generate gate-level implementation of a design from its VHDL description. A behavioral VHDL description of a circuit describes the function of the circuit in terms of its inputs using the types of statements used in a high-level programming language. The objective is to describe the correct operation of a circuit to be designed without being concerned with redundant details. This description does not specify how the function is actually implemented; thus the same description may result in several implementations of a circuit. The register transfer level (RTL) description of a circuit specifies the flow of data from an input or a register to another register or the output of the circuit through combinational logic blocks. The RTL description is also known as data flow description. The structural level description specifies what components a circuit is composed of and how these components are interconnected. This is similar to the logic schematic diagram of a circuit. The purpose of this tutorial is to describe the modeling language VHDL. VHDL includes facilities for describing logical structure and function of digital systems at a number of levels of abstraction, from system level down to the gate level. It is intended, among other things, as a modeling language for specification and simulation. We can also use it for hardware synthesis if we restrict ourselves to a subset that can be automatically translated into hardware. VHDL arose out of the United States government's Very High Speed Integrated Circuits (VHSIC) program.

4.2 LEVELS OF REPRESENTATION AND ABSTRACTION

A digital system can be represented at different levels of abstraction [1]. This keeps the description and design of complex systems manageable. Figure 1 shows different levels of abstraction.

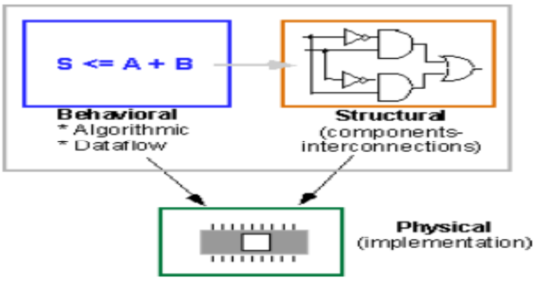


Fig-6 Behavioral, Structural and Physical

4.2.1 BEHAVIORAL LEVEL

The highest level of abstraction is the behavioral level that describes a system in terms of what it does (or how it behaves) rather than in terms of its components and interconnection between them. A behavioral description specifies the relationship between the input and output signals. This could be a Boolean expression or a more abstract description such as the Register Transfer or Algorithmic level.

4.2.2 STRUCTURAL LEVEL

The structural level, on the other hand, describes a system as a collection of gates and components that are interconnected to perform a desired function. A structural description could be compared to a schematic of interconnected logic gates. It is a representation that is usually closer to the physical realization of a system. For the example above, the structural representation is shown.

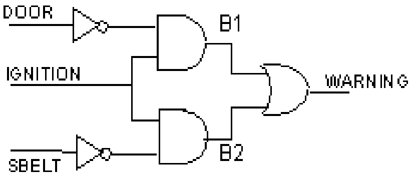


Fig 7. Structural representation of a "buzzer" circuit

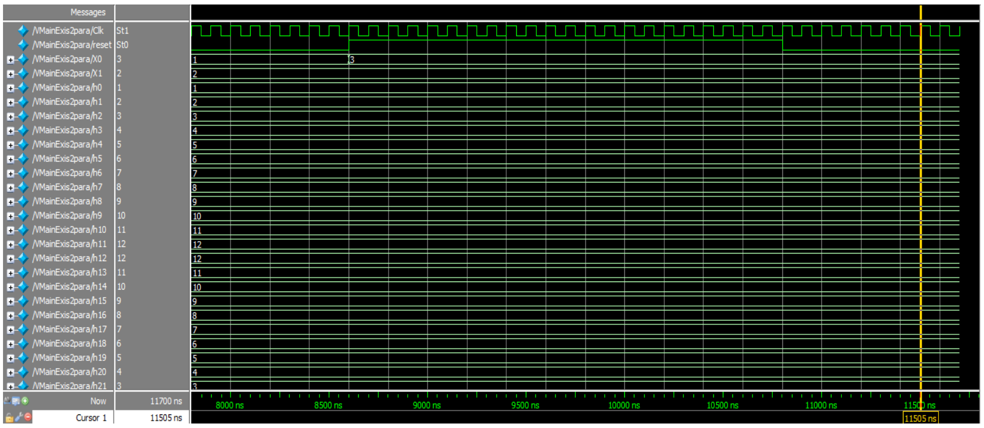
VHDL allows one to describe a digital system at the structural or the behavioral level. The behavioral level can be further divided into two kinds of styles: Data flow and Algorithmic. The dataflow representation describes how data moves through the system. This is typically done in terms of data flow between registers (Register Transfer level). The data flow model makes use of concurrent statements that are executed in parallel as soon as data arrives at the input. On the other hand, sequential statements are executed in the sequence that they are specified. VHDL allows both concurrent and sequential signal assignments that will determine the manner in which they are executed.

4.3 VERILOG

Verilog is one of the two major Hardware Description Languages (HDL) used by hardware designers in industry and academia Verilog is very C-like and liked by electrical and computer engineers as most learn the C language in college. Verilog was introduced in 1985 by Gateway Design System Corporation, now a part of Cadence Design Systems, Inc.'s Systems Division.

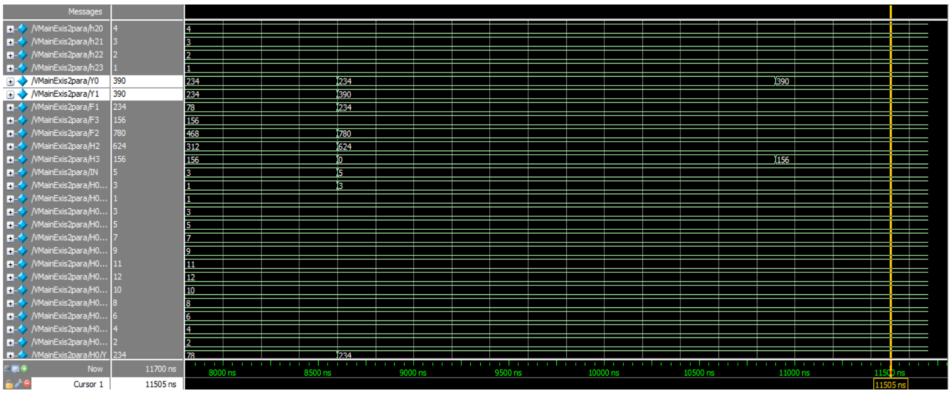
V. Outputs

EXISTING TWO PARALLEL FIR:



```
Messages
/MainExis2para/Clk     St1
/MainExis2para/reset   St0
/MainExis2para/X0      3
/MainExis2para/X1      2
/MainExis2para/h0      1
/MainExis2para/h1      2
/MainExis2para/h2      3
/MainExis2para/h3      4
/MainExis2para/h4      5
/MainExis2para/h5      6
/MainExis2para/h6      7
/MainExis2para/h7      8
/MainExis2para/h8      9
/MainExis2para/h9      10
/MainExis2para/h10     11
/MainExis2para/h11     12
/MainExis2para/h12     12
/MainExis2para/h13     11
/MainExis2para/h14     10
/MainExis2para/h15     9
/MainExis2para/h16     8
/MainExis2para/h17     7
/MainExis2para/h18     6
/MainExis2para/h19     5
/MainExis2para/h20     4
/MainExis2para/h21     3
Now                    11700 ns
Cursor 1               11505 ns
```

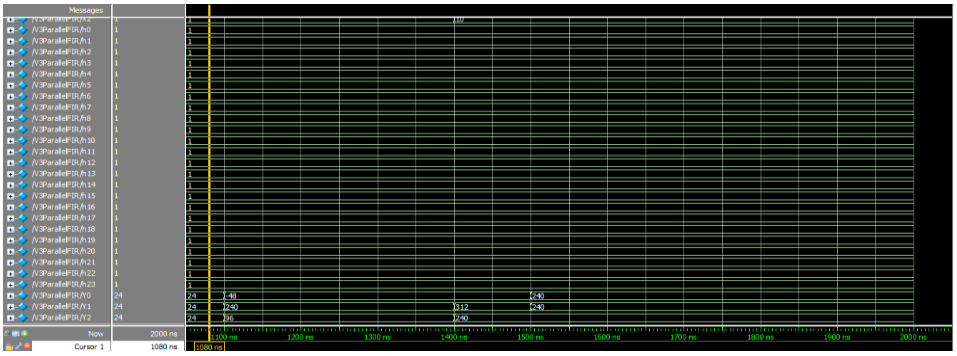
Fig 8. Existing two parallel FIR



```
Messages
/MainExis2para/h20     4
/MainExis2para/h21     3
/MainExis2para/h22     2
/MainExis2para/h23     1
/MainExis2para/Y0      390
/MainExis2para/Y1      390
/MainExis2para/F1      234
/MainExis2para/F3      156
/MainExis2para/F2      780
/MainExis2para/H2      624
/MainExis2para/H3      156
/MainExis2para/IN      5
/MainExis2para/H0...   3
/MainExis2para/H0...   1
/MainExis2para/H0...   3
/MainExis2para/H0...   5
/MainExis2para/H0...   7
/MainExis2para/H0...   9
/MainExis2para/H0...   11
/MainExis2para/H0...   12
/MainExis2para/H0...   10
/MainExis2para/H0...   8
/MainExis2para/H0...   6
/MainExis2para/H0...   4
/MainExis2para/H0...   2
/MainExis2para/H0/Y    234
Now                    11700 ns
Cursor 1               11505 ns
```

Fig 9.Existing two parallel FIR

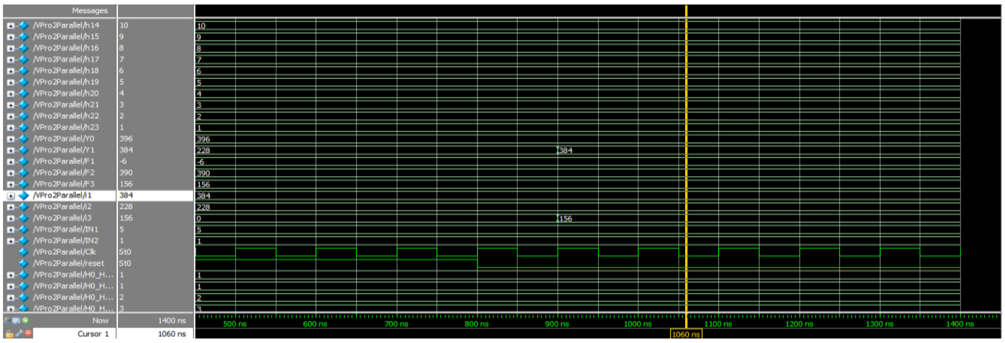
EXISTING THREE PARALLEL FIR FILTER:



```
Messages
/3ParallelFIR/X2       1
/3ParallelFIR/h0       1
/3ParallelFIR/h1       1
/3ParallelFIR/h2       1
/3ParallelFIR/h3       1
/3ParallelFIR/h4       1
/3ParallelFIR/h5       1
/3ParallelFIR/h6       1
/3ParallelFIR/h7       1
/3ParallelFIR/h8       1
/3ParallelFIR/h9       1
/3ParallelFIR/h10      1
/3ParallelFIR/h11      1
/3ParallelFIR/h12      1
/3ParallelFIR/h13      1
/3ParallelFIR/h14      1
/3ParallelFIR/h15      1
/3ParallelFIR/h16      1
/3ParallelFIR/h17      1
/3ParallelFIR/h18      1
/3ParallelFIR/h19      1
/3ParallelFIR/h20      1
/3ParallelFIR/h21      1
/3ParallelFIR/h22      1
/3ParallelFIR/h23      1
/3ParallelFIR/Y0       24
/3ParallelFIR/Y1       24
/3ParallelFIR/Y2       24
Now                    2000 ns
Cursor 1               1080 ns
```

Fig 10. Existing three parallel FIR

PROPOSED TWO PARALLEL FIR FILTER:



```
Messages
/Pro2Parallel/h14      10
/Pro2Parallel/h15      9
/Pro2Parallel/h16      8
/Pro2Parallel/h17      7
/Pro2Parallel/h18      6
/Pro2Parallel/h19      5
/Pro2Parallel/h20      4
/Pro2Parallel/h21      3
/Pro2Parallel/h22      2
/Pro2Parallel/h23      1
/Pro2Parallel/Y0       396
/Pro2Parallel/Y1       384
/Pro2Parallel/F1       -6
/Pro2Parallel/F2       390
/Pro2Parallel/F3       156
/Pro2Parallel/i1       384
/Pro2Parallel/i2       228
/Pro2Parallel/i3       156
/Pro2Parallel/IN1      5
/Pro2Parallel/IN2      1
/Pro2Parallel/Clk      St0
/Pro2Parallel/reset    St0
/Pro2Parallel/H0_H...  1
/Pro2Parallel/H0_H...  1
/Pro2Parallel/H0_H...  2
/Pro2Parallel/H0_H...  3
Now                    1400 ns
Cursor 1               1060 ns
```

Fig 11. Proposed two parallel FIR

VI. Conclusion

In this paper, we have presented new parallel FIR filter structures. Multipliers are the major portions in hardware consumption for the parallel FIR filter implementation. The proposed new structure exploits the nature of even symmetric coefficients and save a significant amount of multipliers at the expense of additional adders. Since multipliers outweigh adders in hardware cost, it is profitable to exchange multipliers with adders. 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 FIR filter. Consequently, the larger the length of FIR filters is, the more the proposed structures can save from the existing FFA structures, with respect to the hardware cost. Overall, in this paper, we have provided new parallel FIR structures consisting of advantageous polyphase decompositions dealing with symmetric convolutions comparatively better than the existing FFA structures in terms of hardware consumption.

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, 1997.

[2] J. G. Chung and K. K. Parhi, “Frequency-spectrum-based low-area low-power parallel FIR filter design,” *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 9, pp. 444–453, 2002.

[3] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: 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] J. I. Acha, “Computational structures for fast implementation of L-path and L-block digital filters,” *IEEE Trans. Circuit Syst.*, vol. 36, no. 6, pp. 805–812, Jun. 1989.

[6] 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.

[7] C.Cheng and K. K.Parhi, “Furthur complexity reduction of parallel FIR filters,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS 2005)*, Kobe, Japan, May 2005.

[8] 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.

[9] I.-S. Lin and S. K. Mitra, “Overlapped block digital filtering,” *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 8, pp. 586–596, Aug. 1996.

[10] *“Design Compiler User Guide,”* ver. B-2008.09, Synopsys Inc., Sep.2008