

# An Area-Efficient Fast Parallel FIR Digital Filter Structure Using Modified SQRT Carry Select Adder

Shari.R

Department of Electronics and Communication Engineering

Archana College of Engineering,

Alappuzha Dist.

Email:sharir101@gmail.com

**Abstract**— Based on fast finite-impulse response (FIR) algorithms (FFAs), a new parallel FIR filter structures is implemented. They are beneficial to symmetric coefficients in terms of the area, power and hardware overhead. 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 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. Here modified SQRT Carry select adder is replaced with the existing architecture, to achieve low power and less area. These ideas are implemented using VHDL in Xilinx and Modelsim tools. A significant reduction in power consumption up to 12% and area reduction up to 6% are observed when compared to the existing structure.

**Keywords**— Finite impulse response(FIR) ,Fast FIR algorithm(FFA),SQRT Carry Select Adder.

## I. INTRODUCTION

In DSP digital filters are very important because of their extraordinary performance in DSP applications. Two uses of digital filters are signal separation and signal restoration. Signal separation is needed when a signal has been contaminated with interference, noise, or other signals. Signal restoration is used when a signal has been distorted in some way. A finite impulse response (FIR) filter is a type of a digital filter. Here the impulse response is finite because it settles to zero in a finite number of sample intervals. This is in contrast to infinite impulse response (IIR) filters. IIR filters have internal feedback and may continue to respond indefinitely. The impulse response of an Nth-order FIR filter lasts for  $N+1$  sample, and then dies to zero. The finite-impulse response (FIR) filter is one of the fundamental processing elements in any digital signal processing (DSP) system. FIR filters are used in DSP applications that range from video and image processing to wireless communications. In some applications, such as video processing, the FIR filter circuit must be able to operate at high frequencies. But in some other applications, such as cellular telephony, the FIR filter circuit must be a low-power circuit, capable of operating at moderate frequencies. Parallel, or block, processing can be applied to digital FIR filters to either increase the effective throughput or reduce the power consumption of the original filter. Traditionally, the application of parallel processing of

an FIR filter involves the replication of the hardware units that exist in the original filter. If the area required by the original circuit is A, then the L-parallel circuit requires an area of  $L \times A$ . With the continuing trend to reduce chip size and integrate multi-chip solutions into a single chip solution, it is important to limit the silicon area required to implement a parallel FIR digital filter in its VLSI implementation. In many design situations, the hardware overhead incurred by parallel processing cannot be tolerated due to limitations in design area. Therefore, it is advantageous to have parallel FIR filtering structures that consume less area than traditional parallel FIR filtering structures.

There have been a few papers that propose a number of ways to reduce the complexity of the parallel FIR filter in the past. Here polyphase decomposition is mainly manipulated so that 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) shows that it can implement a 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 L. It reduces the required number of multipliers to  $(2N-N/L)$  from  $L \times N$ . In later papers, 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.

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 polyphase decompositions, which can reduce amounts of multiplications in the subfilter section by exploiting the inherent nature of the symmetric coefficients, compared to the existing FFA fast parallel FIR filter structure[1]. In order to speed up the processing carry select adder is utilized. By replacing existing adder with modified SQRT carry select adder[2], area and power can be further reduced.

## II. METHODOLOGY

### A. FAST FIR ALGORITHM (FFA)

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

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

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

$$\sum_{p=0}^{L-1} Y_p(z^L)z^{-p} = \sum_{q=0}^{L-1} X_q(z^L)z^{-q} \sum_{r=0}^{L-1} H_r(z^L)z^{-r} \quad (2)$$

where ,

$$X_q = \sum_{k=0}^{\infty} z^{-k} x(Lk+q),$$

$$H_r = \sum_{k=0}^{\left(\frac{N}{L}\right)-1} z^{-k} x(Lk+r),$$

$$Y_p = \sum_{k=0}^{\infty} z^{-k} x(Lk+p), \text{ for } p, q, r=0, 1, 2, 3, \dots, L-1.$$

### 2X2 FFA (L=2)

From (2), a two-parallel FIR filter can be expressed as,

$$\begin{aligned} Y_0 + z^{-1}Y_1 &= (H_0 + z^{-1}H_1)(X_0 + z^{-1}X_1) \\ &= H_0X_0 + z^{-1}(H_0X_1 + H_1X_0) + z^{-2}H_1X_1 \end{aligned} \quad (3)$$

Implying that,

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

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

From equation (4), the traditional two-parallel filter structure will require four length-N/2 FIR subfilter blocks, two post processing adders, and totally 2N multipliers and 2N-2 adders. Although (4) can be rewritten as,

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

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

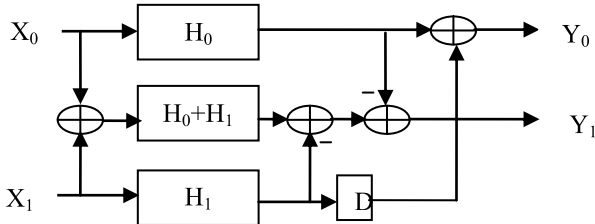


Fig.1 - Two-parallel FIR filter implementation using FFA

In order to implement (5), it will require three FIR subfilter blocks of Length N/2, one preprocessing and three postprocessing adders, and 3N/2 multipliers and 3(N/2-1)+1 adders. So that it can reduce approximately one fourth over the traditional two-parallel filter hardware cost from (4). The two-parallel (L=2) FIR filter implementation using FFA obtained from (5) is shown in Figure 1.

### B. MODIFIED FFA STRUCTURES FOR SYMMETRIC CONVOLUTIONS

By utilizing the symmetry of coefficients, earn as many subfilter 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. It 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. Hence, for an N-tap L-parallel FIR filter the total amount of saved multipliers would be the number of sub filter blocks that contain symmetric coefficients times half the number of multiplications in a single sub filter block N/2L.

### 2X2 Modified FFA (L=2)

By rewriting equation(4), a two-parallel FIR filter can also be written as

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

$$Y_1 = 1/2[(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] \quad (6)$$

In the case of a set of even symmetric coefficients, (6) can earn one more sub filter block containing symmetric coefficients than (5), the existing FFA parallel FIR filter. Figure 2 shows implementation of the proposed two-parallel FIR filter based on (6).

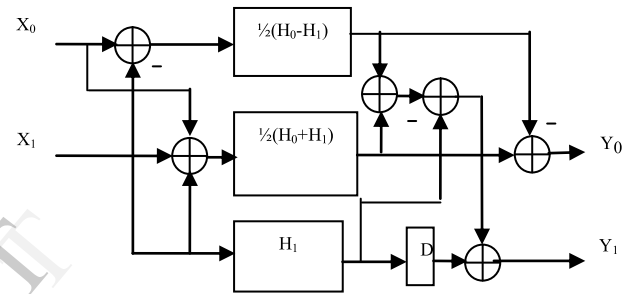


Fig.2 - Modified two-parallel FIR filter implementation

For example, consider a 24-tap FIR filter with a set of symmetric coefficients  $\{h(0), h(1), h(2), h(3), h(4), h(5), h(6), h(7), h(8), h(9), \dots, h(23)\}$  where  $h(0)=h(23), h(1)=h(22), h(2)=h(21), h(3)=h(20), h(4)=h(19), h(5)=h(18), h(6)=h(17), h(7)=h(16), h(8)=h(15), \dots, h(11)=h(12)$ , applying to the proposed two-parallel FIR filter structure.

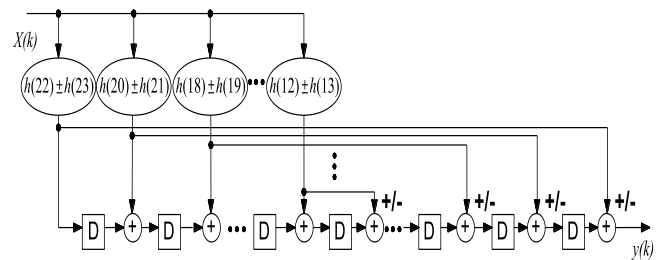


Fig.3 - Subfilter block implementation with symmetric coefficients

Then the top two subfilter blocks will be as

$$\begin{aligned} H_0 \pm H_1 &= \{h(0) \pm h(1), h(2) \pm h(3), h(4) \pm h(5), h(6) \pm h(7), \\ &h(8) \pm h(9), \dots, h(18) \pm h(19), \\ &h(20) \pm h(21), h(22) \pm h(23)\} \end{aligned}$$

where

$$\begin{aligned}
 h(0) \pm h(1) &= \pm(h(22) \pm h(23)) \\
 h(2) \pm h(3) &= \pm(h(20) \pm h(21)) \\
 h(4) \pm h(5) &= \pm(h(18) \pm h(19)) \\
 h(6) \pm h(7) &= \pm(h(16) \pm h(17)) \\
 h(8) \pm h(9) &= \pm(h(14) \pm h(15)) \dots\dots\dots (7)
 \end{aligned}$$

From the example above, two of three subfilter blocks from the proposed two-parallel FIR filter structure,  $H_0+H_1$  and  $H_0-H_1$ , are with symmetric coefficients as shown in (7). That means the subfilterblock can be realized with only half the amount of multipliers required as shown in the figure 3.

### III.PROBLEM DEFINITION

#### FIR FILTER USING CONVENTIONAL CARRY SELECT ADDER

Here FIR Filter using FFA is implemented using carry select adder[8]. Adders are one of the most widely used components in integrated circuits, designing efficient adders has been the goal of much research in VLSI design. Ripple Carry Adders (RCAs) have the most compact design ( $O(n)$  area) among all types of adders, but they are the slowest types of adders ( $O(n)$  time). On the other hand, Carry Look-ahead Adders (CLAs) are the fastest adders ( $O(\log(n))$  time), but they are the worst from the area point of view ( $O(n \log(n))$  area).

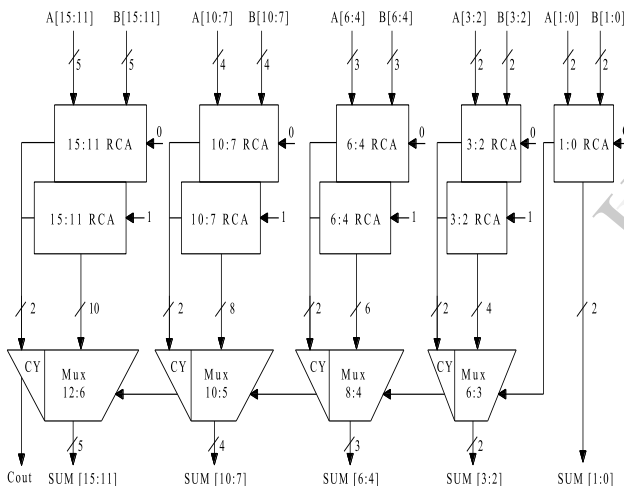


Fig.4 - Regular Sqrt Carry Select Adder

Carry Select Adders (CSAs) have been considered as a compromise solution between RCAs and CLAs ( $O(n)$  time and  $O(2n)$  area) because they offer a good tradeoff between the compact area of RCAs and the short delay of CLAs. The conventional  $n$ -bit CSA consists of one  $n/2$ -bit adder for the lower half of the bits and two  $n/2$ -bit adders for the upper half of the bits. Of the two latter adders, one performs the addition with the assumption that  $Cin=0$ , whereas the other does this with the assumption that  $Cin=1$ . Using a multiplexer and the value of carry out that is propagated from the adder for the  $n/2$  least significant bits, the correct value of the most significant part of the addition can be selected. Although this technique has the drawback of increasing the area, it speeds up the addition operation.

#### IV.PROPOSED FIR FILTER WITH MODIFIED Sqrt CARRY SELECT ADDER

The CSLA is used in many computational systems to improve the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. But still, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input  $Cin=0$  and  $Cin=1$ , then the final sum and carry are selected by the multiplexers (mux).

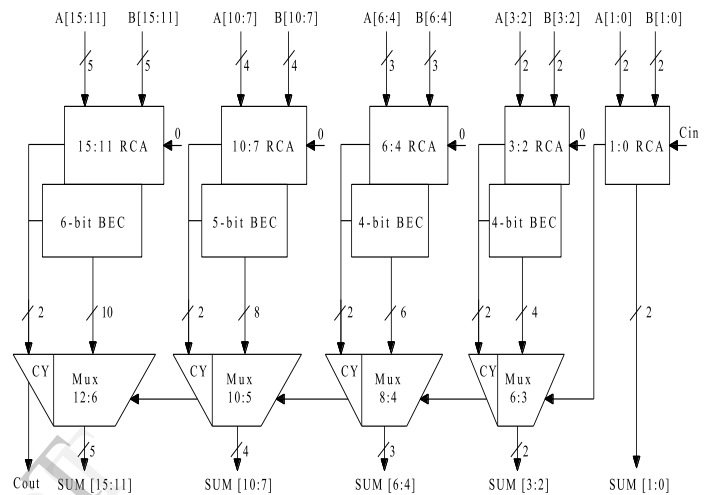


Fig.5 - Modified Sqrt Carry Select Adder

Here a Binary to Excess-1 Converter (BEC) is used instead of RCA with  $Cin=1$  in the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic[2] is that it utilizes lesser number of logic gates than the  $n$ -bit Full Adder (FA) structure.

#### BEC

To replace the  $n$ -bit RCA, an  $n+1$ -bit BEC is required. The structure of a 4-bit BEC is as shown in the figure 6.

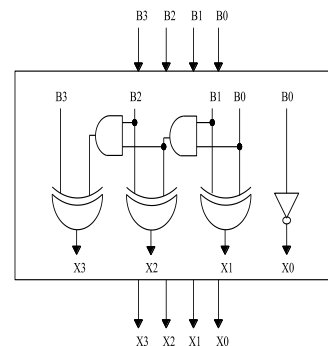


Fig.6 - 4-bit BEC

Fig.7 shows how the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. One input of the 8:4 mux gets as the RCA output (B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial results in parallel. Then

MUX is used to select either the BEC output or RCA output according to the control signal  $C_{in}$ . The importance of the BEC logic is that it requires only lesser number of logic gates compared to RCA. As a result a larger silicon area reduction when the CSLA with large number of bits are designed.

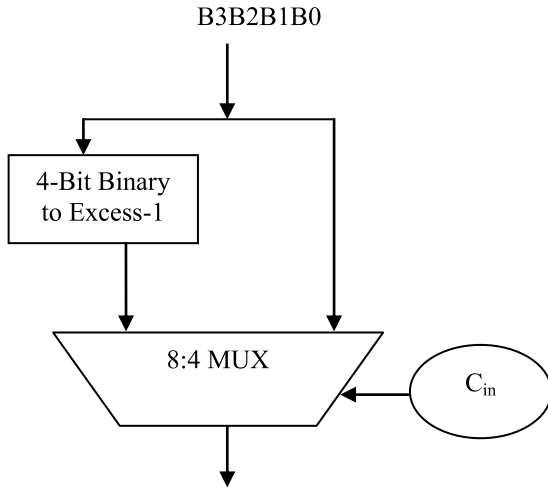


Fig.7 - 4-b BEC with 8:4 MUX

The Boolean expressions of the 4-bit BEC is obtained as (the functional symbols of ~NOT, &AND, ^XOR)

$$\begin{aligned} X_0 &= \sim B_0 \\ X_1 &= B_0 \wedge B_1 \\ X_2 &= B_2 \wedge (B_0 \wedge B_1) \\ X_3 &= B_3 \wedge (B_0 \wedge B_1 \wedge B_2). \end{aligned}$$

TABLE I - FUNCTIONAL TABLE OF 4-B BEC

B[3:0]	X[3:0]
0000	0001
0001	0010.
.	.
.	.
.	.
.	.
1110	1111
1111	0000

## V. EXPERIMENTAL RESULTS

The simulation is done in modelsim and Xilinx tool is used for power estimation and area analysis. Spartan 3 is the FPGA used for the implementation process. The power has been reduced from 2.053 to 1.788. Moreover the number of 4-input LUT's have been reduced from 3712 to 3246. Thus, the power and area estimation have proven to be efficient when compared to the existing FIR structure.

TABLE II - COMPARISON BETWEEN EXISTING AND PROPOSED FIR FILTER STRUCTURE

Architecture	Power(mW)	No: of gates	No: of slices
Existing FIR	2.053	22272	1924
Proposed FIR	1.788	19476	1666

## VI. CONCLUSION

In this paper, a new parallel FIR filter structures is introduced, which are beneficial to symmetric convolutions. For the parallel FIR filter implementation multipliers consumes a major portion of the hardware. The proposed new structure saves a significant amount of multipliers at the expense of additional adders. Since adders weigh less than multipliers, it is profitable to exchange multipliers with adders. In addition, the number of increased adders stays still when the length of FIR filter becomes large. Also, along with the length of FIR filter the number of reduced multipliers increases. Therefore, 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 the proposed parallel structure uses modified SQR carry select adder. So that the area and power can be much reduced. The proposed structure is implemented using VHDL language in Xilinx and Modelsim tools. With this proposed structure power can be reduced upto 12% and area can be reduced upto 6%.

## REFERENCES

- [1] Yu-Chi Tsao and Ken Choi, "Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm", *IEEE trans. Very large scale integration (vlsi) systems*, vol.3, no.1, pp.1-5, Nov.2010.
- [2] B. Ramkumar and Harish M Kittur, "Low-Power and Area-Efficient Carry Select Adder", *IEEE trans. Very large scale integration (vlsi) systems*, vol.3, no.1, pp.1-5, Nov.2010.
- [3] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for lowpower applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, vol. 4, pp. 4082-4085.
- [4] O. J. Bedrij, "Carry-select adder," *IRE Trans. Electron. Comput.*, pp.340-344, 1962.
- [5] 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.
- [6] B. Ramkumar and Harish M Kittur, "Low-Power and Area-Efficient Carry Select Adder", *IEEE trans. Very large scale integration (vlsi) systems*, pp.1-5, Feb 2011.
- [7] C. Cheng and K. K. Parhi, "Further 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, "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.
- [9] 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.
- [10] B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," *Eur. J. Sci. Res.*, vol. 42, no. 1, pp. 53–58, 2010.
- [11] T. Y. Ceiang and M. J. Hsiao, "Carry-select adder using single ripple carry adder," *Electron. Lett.*, vol. 34, no. 22, pp. 2101–2103, Oct. 1998.
- [12] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," *Electron. Lett.*, vol. 37, no. 10, pp. 614–615, May 2001.
- [13] Youngjoon Kim and Lee-Sup Kim, "A Low Power Carry Select Adder With Reduced Area," *Electron. Lett.*, vol. 37, no. 10, pp. 614–615, May 2001.
- [14] 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.
- [15] 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.

IJERT