IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 1, Ver. I (Jan. 2014), PP 06-09
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Reducing computation delay of Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm by using CSA

G Ramya Sudha[1], Ch.Manohar[2]
M.Tech, Department of ECE, KIET (1)
Asst Professor, Department of ECE, KIET (2)
Kakinada, AP (1) (2)

Abstract: *In this paper, our main focus is to reduce computation delay proposed in previous FFA algorithm based FIR digital filter. Parallel FIR filter structure with fast finite-impulse response (FIR) along with symmetric coefficients reduces 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. The proposed parallel filters uses normal adders (full adder and ripple carry adder) that take more time to execute the program. Therefore, we replaced Ripple Carry Adder (RSA) with Carry Save Adder (CSA) and finally presented the comparison between the timing delays with RSA and CSA. 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. All the simulations observed in modelsim6.4b simulator, synthesis by Xilinx ISE tool.*
Key words*: CSA (carry save adder), FFA, FIR.*

I. Introduction

The demand for high intensive data application and low power digital signal processing (DSP) is increasing day-by-day. FIR filter is the fundamental device used in DSP to meet those requirements. Many applications require FIR filter to operate at low, moderate and high frequencies. Some applications need the FIR filter to operate at high frequencies such as video processing, whereas some other applications request high throughput with a low-power circuit such as multiple-input multiple-output (MIMO) systems used in cellular wireless communication. Furthermore, when narrow transition-band characteristics are required, the much higher order in the FIR filter is unavoidable. For example, a 576-tap digital filter is used in a video ghost canceller for broadcast television, which reduces the effect of multipath signal echoes. On the other hand, parallel and pipelining processing are two techniques used in DSP applications, which can both be exploited to reduce the power consumption. 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 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 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*N. 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.

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 de-compositions, 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. This paper is organized as follows. A brief introduction of FFAs is given in Section II. In Section III, the proposed parallel FIR filter structures are presented. Section IV investigates the proposed CSA in FFA. In Section V, shows the experimental results. Section VI gives the conclusion.

II. Fast Fir Algorithm (FFA)

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

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

where is an infinite-length input sequence and are the length- FIR filter coefficients. Then, the traditional 0 parallel FIR filter can be derived using polyphase decomposition as [3]

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

where $X_q = \sum_{k=0}^{(N/L)-1} z^{-k}x(Lk+r), Y_p = \sum_{k=0}^{\infty} z^{-k}x(Lk+q), H_r = \sum_{k=0}^{\infty} z^{-k}x(Lk+p),$

For p,q,r =0,1,2,3 …,L-1

It shows that the traditional FIR filter will require -FIR sub filter blocks of length for implementation.

A. 2×2 FFA $(L = 2)$

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

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

Implying that

$$\begin{aligned} Y_0 &= H_0X_0 + z^{-2}H_1X_1, \\ Y_1 &= H_0X_1 + H_1X_0. \end{aligned} \qquad (4)$$

Equation (4) shows the traditional two-parallel filter structure, which will require four length-N/2 FIR subfilter blocks, two post -processing

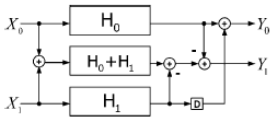

Fig. 1. Two-parallel FIR filter implementation using FFA.

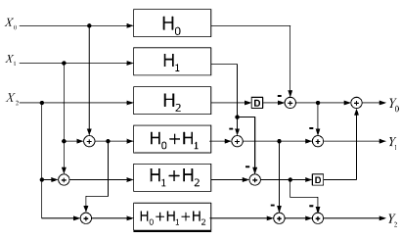

Fig: 2. Three parallel FIR filter implementation using FFA.

adders, and totally 2N multipliers and2N-2 adders. However, (4)
can be written as

$$Y_0 = H_0X_0 + z^{-2}H_1X_1$$
$$Y_1 = (H_0 + H_1)(X_0 + X_1) - H_0X_0 - H_1X_1. \qquad (5)$$

The implementation of (5) 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 (4). The two-parallel (L=2) FIR filter implementation using FFA obtained from (5) is shown in Fig. 1.

B. 3X3 FFA (L=3)

By the similar approach, a three-parallel FIR filter using FFA can be expressed as

$$\begin{aligned} Y_0 &= H_0X_0 - z^{-3}H_2X_2 + z^{-3} \\ &\quad \times [(H_1 + H_2)(X_1 + X_2) - H_1X_1] \\ Y_1 &= [(H_0 + H_1)(X_0 + X_1) - H_1X_1] \\ &\quad - (H_0X_0 - z^{-3}H_2X_2) \\ Y_2 &= [(H_0 + H_1 + H_2)(X_0 + X_1 + X_2)] \\ &\quad - [(H_0 + H_1)(X_0 + X_1) - H_1X_1] \\ &\quad - [(H_1 + H_2)(X_1 + X_2) - H_1X_1]. \qquad (6) \end{aligned}$$

The hardware implementation of (6) requires six length-N/3FIR 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. The implementation obtained from (6) is shown in Fig. 2.

III. Symmetric Convolutions based on FFA Structures

The main idea behind the proposed structures is actually pretty intuitive, to manipulate the polyphase decomposition to earn as many subfilter blocks as possible which contain symmetric coefficients so that half the number of multiplications in the single subfilter 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. Therefore, for an N-tap L-parallel FIR filter the total amount of saved multipliers would be the number of subfilter blocks that contain symmetric coefficients times half the number of multiplications in a single subfilter block 2N/L.

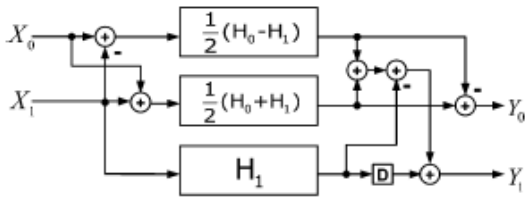

Fig:3 Proposed two-parallel FIR filter implementation

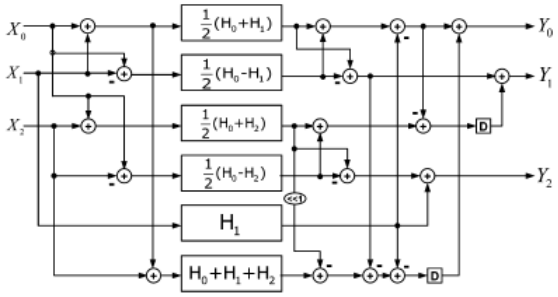

Fig. 4. Proposed three-parallel FIR filter implementation.

IV. Proposed Carry-Save Adder in FFA

A carry-save adder is a type of digital adder, used in computer microarchitecture to compute the sum of three or more n-bit numbers in binary. It differs from other digital adders in that it outputs two numbers of the same dimensions as the inputs, one which is a sequence of partial sum bits and another which is a sequence of carry bits. Consider the sum:

12345678 + 87654322 =100000000.

Using basic arithmetic, we calculate right to left, "8+2=0, carry 1", "7+2+1=0, carry 1", "6+3+1=0, carry 1", and so on to the end of the sum. Although we know the last digit of the result at once, we cannot know the first digit until we have gone through every digit in the calculation, passing the carry from each digit to the one on its left. Thus adding two n-digit numbers has to take a time proportional, even if the machinery we are using would otherwise be capable of performing many calculations simultaneously.

In electronic terms, using bits (binary digits), this means that even if we have n one-bit adders at our disposal, we still have to allow a time proportional to n to allow a possible carry to propagate from one end of the number to the other. Until we have done this,

1. We do not know the result of the addition.
2. We do not know whether the result of the addition is larger or smaller than a given number (for instance, we do not know whether it is positive or negative).

CSA is used at the addition process to reduce the computation delay. CSA adder shown in below

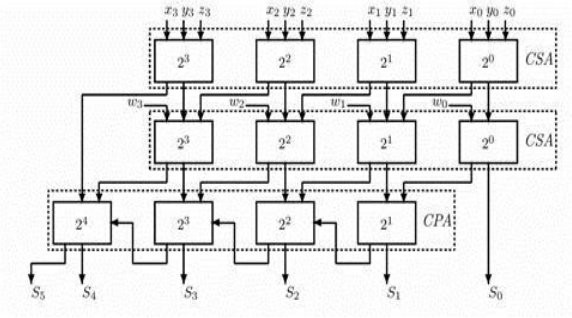

Fig:5 carry save adder architecture.

V. Results

These results are observed in Model SIM simulator and Xilinx synthesis tool.

Table I: Timing delays comparison

Type	RCA	CSA
2X2ffa	23.134	19.329
3x3ffa	28.87	25.101
2x2prffa	27.86	25.66
3x3praffa	27.38	26.852

VI. Conclusion

In this paper, 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. Therefore, we have presented the comparison of parallel FIR filter structure with ripple carry adder and carry save adder. From the results, it shows that Carry save adder performs better than ripple carry adder.

References

[1] Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm Yu-Chi Tsao and Ken Choi

[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.