1 Introduction

THE explosive growth of multimedia application, the demand for high-performance and low-power digital signal processing (DSP) is getting higher and higher. 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. 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. 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. It reduces the required number of multipliers to 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 poly phase 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. In proposed we present the radix-2k feed forward FFT architectures. The proposed designs include radix-22, radix-23 and radix-24 architectures. The paper shows that radix can be used for any number of parallel samples which is a power of two. Accordingly, radix-2k FFT architectures for 2, 4, and 8 parallel samples are presented. These architectures are shown to be more hardware-efficient than previous feed forward and parallel feedback designs in the literature. This makes them very attractive for the computation of the FFT in the most demanding applications. The radix 24 FFT architecture, basic FFT Algorithm is used to design Fig.1. The architecture is designed based on the following three properties in FFT algorithm.

D.ARAVINDARAJ¹, S. SABARINATHAN²

Salem College of Engineering and Technology, Salem

Abstract: Based on a 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. In this work we present the radix- feed forward FFT architectures. The paper shows that radix- can be used for any number of parallel samples which is a power of two. Accordingly, radix-2k FFT architectures for 2, 4, and 8 parallel samples are presented. In feed forward architectures radix-2k can be used for any number of parallel samples which is a power of two. Indeed, the number of parallel samples can be chosen arbitrarily depending of the throughput that is required. The proposed parallel FIR structures exploit the inherent nature of symmetric coefficients reducing half the number of multipliers in subfilter 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, In addition to this, the designs can achieve a very high throughput, which makes them suitable for the most demanding applications.

Keywords: parallel FIR, symmetric convolution.

Reference to this paper should be made as follows: D.Arvindaraj¹, S. Sabarinathan² (2014) ‘Area- Power Efficient Parallel Fir Digital Filter Structures For Symmetric Convolutions Based On Fast Fir Algorithm’, International Journal of Inventions in Computer Science and Engineering, Volume 1 Issue 3 2014
Fig.1 basic FFT Algorithm is used to design

the discrete Fourier transform (DFT) matrix factorization based on the Kronecker product to express the family of radix rk single-path delay commutator/single-path delay feedback (SDC/SDF) pipeline fast Fourier transform (FFT) architectures. The matricial expressions of the radix r, r 2, r 3, and r 4 decimation-in-frequency (DIF) SDC/SDF pipeline architectures are derived. The derived expressions are general in terms of r and the number of points of the FFT N. Expressions are given where it is not necessary that N is a power of r[k].

that method of providers in high throughput rate (T.R.) by applying the eight-data-path pipelined approach for wireless personal area network applications. The hardware costs, including the power consumption and area, increase due to multiple data paths and increased word length along stages. A multi data scaling scheme to reduce word lengths while preserving the signal-to-quantization-noise ratio is also presented. Using UMC 90-nm 1P9M technology [2]. The proposed Radix 22 Parallel Pipeline processor, which employs two parallel data path Radix 22 algorithm and single-path delay feedback (SDF) pipeline architecture, is a small-area and low-power-consomption solution for MB-OFDM UWB system.

Both FPGA Xilinx Virtex4 and ASIC 90 nm technology, 1 V supply voltage targeted synthesis results of this architecture are presented. The required gates are 39000 without testing block and the corresponding area is 181140 Âm2 [3]. The architecture takes advantage of the reduced number of operations of the RFFT with respect to the complex fast Fourier transform (CFFT), and requires less area while achieving higher throughput and lower latency. Of using algorithm can be used for both the decimation in time (DIT) and decimation in frequency (DIF) decompositions of the RFFT and requires the lowest number of operations reported for radix 2 [4]. We consider structures for simultaneous multiplication by a small set of two pair wise coefficients where the coefficients are the real and imaginary part of a limited number of points uniformly spread on the unit circle. Hence, each such multiplier forms half of a complex multiplier suitable for twiddle factor multiplication in FFT architectures. Based on trigonometric identities we propose a multiplier for a unit circle resolution of 32 points [5].

2. Related Work

In existing, feedback architectures are characterized by their feedback loops, i.e., some outputs of the butterflies are fed back to the memories at the same stage. Feedback architectures can be divided into single-path delay feedback (SDF) which processes a continuous flow of one sample per clock cycle, and multi-path delay feedback (MDF) or parallel feedback which process several samples in parallel. On the other hand, feed forward architectures also known as multi-path delay commutator (MDC), do not have feedback loops and each stage passes the processed data to the next stage.

2.1 Designing Radix-22 Fft Architectures

The architectures have been derived using the framework presented. The design is based on analyzing the flow graph of the FFT and extracting the properties of the algorithm. These properties are requirements that any hardware architecture that calculates the algorithm must fulfill.

2.2 Radix-22 Feed Forward FFT Architectures

This section presents the radix-22 feed forward architectures. First, a 16-point 4-parallel radix-22 feed forward FFT architecture is explained in depth in order to clarify the approach and show how to analyze the architectures. Then, radix-22 feedforward architectures for different number of parallel samples are presented. Fig.2 shows a 16-point 4-parallel radix-22 feed forward FFT architecture. The architecture is made up of radix-2 butterflies (R2), non-trivial rotators (®), trivial rotators, which are diamond-shaped, and shuffling structures, which consist of buffers and multiplexers. The lengths of the buffers are indicated by a number.

The Existing method of using architectures has been programmed for the use in field-programmable gate arrays (FPGAs). The designs are parameterizable in the number of points N, word length, and number of samples in parallel P. Table IV shows post-place and route results for different configurations of N and P, using a word length of 16 bits.

3. Our Contributions

The proposed parallel FIR structures exploit the inherent nature of symmetric coefficients reducing half the number of multipliers in subfilter 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 continue 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.

3.1 Fast Fir Algorithm (FFA)

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

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

From this FIR filtering equation, it shows that the traditional FIR filter will require $L^2$ FIR subfilter blocks of length $L/N$ for implementation.

$$y_0 + z^{-1}y_1 = (H_0 + z^{-1}H_1)(X_0 + z^{-1}X_1)$$

$$= H_0X_0 + H_1X_1 + z^{-1}H_0X_1 + z^{-1}H_1X_0$$

(3)

3.2 Proposed Ffa Structures For Symmetric Convolutions:

To utilize the symmetry of coefficients, 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.

**A. 2*2 Proposed FFA (L=2)**

From (3), a two-parallel FIR filter can also be written as

\[
\begin{align*}
Y_0 &= \frac{1}{2} \left[ (H_0 + H_1)(X_0 + X_1) + (H_0 - H_1)(X_0 - X_1) \right] + z^{-1}H_1X_1, \\
Y_1 &= \frac{1}{2} \left[ (H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1) \right].
\end{align*}
\]

(4)

When it comes to a set of even symmetric coefficients, (5) can earn one more subfilter block containing symmetric coefficients than 3), the existing FFA parallel FIR filter. Fig. 5 shows implementation of the proposed two-parallel FIR filter based on (5).

An example is demonstrated here for a clearer perspective.

**Example 1:** Consider a 24-tap FIR filter with a set of symmetric coefficients applying to the proposed two-parallel FIR filter.

Applying to the proposed two-parallel FIR filter structure, and the top two subfilter blocks will be as

\[ h(0), h(1), h(2), h(4), h(5), h(6), h(7), h(8), h(9), \ldots \]

(5)

The (6) 

Applying to the proposed two-parallel FIR filter structure, and the top two subfilter blocks will be as

\[ h(0), h(1), h(2), h(4), h(5), h(6), h(7), h(8), h(9), \ldots \]

As can be seen from the example above, two of three subfilter blocks from the proposed two-parallel FIR filter structure are with symmetric coefficients now, as (7).

**B. 3*3 Proposed FFA (L=3)**

With the similar approach, from (4), a three-parallel FIR filter can also be written as (8). Fig. 6 shows implementation of the proposed three-parallel FIR filter. When the number of symmetric coefficients is the multiple of 3, the proposed three-parallel FIR filter structure presented in (9) enables four subfilter blocks with symmetric coefficients in total, whereas the existing FFA parallel FIR filter structure has only two ones out of six subfilter blocks.

\[
\begin{align*}
Y_0 &= \frac{1}{2} \left[ (H_0 + H_1)(X_0 + X_1) + (H_0 - H_1)(X_0 - X_1) \right] - H_1X_1, \\
Y_1 &= \frac{1}{2} \left[ (H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1) \right].
\end{align*}
\]

(8)

Symmetric coefficients. Therefore, for an N tap three-parallel FIR filter, the proposed structure can save N/3 multipliers from the existing FFA structure.

**C. Proposed Cascading FFA**

The proposed cascading process for the larger block-sized proposed parallel FIR filter is similar to that introduced. However, a small modification is adopted here for lower hardware consumption. As we can see, the proposed parallel FIR structure enables the reuse of multipliers in parts of the subfilter blocks but it also brings more adder cost in preprocessing and post processing blocks. When cascading the proposed FFA parallel FIR structures for larger parallel block factor L the increase of adders can become larger. Therefore, other than applying the proposed FFA FIR filter structure to all the decomposed subfilter blocks, the existing FFA structures which have more compact operations in preprocessing and post processing blocks are employed for those subfilter blocks that contain no symmetric coefficients, whereas the proposed FIR filter structures are still applied to the rest of subfilter blocks with symmetric coefficients.

**3.3 Complexity Analysis And Comparison**

**Table I**

Comparison Of Proposed And The Existing FFA Structures Number Of Required Multipliers (M.), Reduced Multipliers (R.M.), Number Of Required Adders In Subfilter Section (Sub.), Number Of Required Adders In Pre/Post processing Blocks (Pre/Post.), And Number Of The Increased Adders (I.A.)
4 Implementation And Experimental Result

The proposed FFA structures and the existing FFA structures are implemented in VHDL with filter length of 24 and 72, word length 16-bit and 32-bit, respectively. Two sets of the ideal low-pass FIR filter symmetric coefficients of length 24 and 72 are generated by MATLAB using Remez Exchange algorithm.

Simulation Tap (L=2)

Simulation tap (L=3)

5 Conclusions

In this paper, we have presented new parallel FIR filter structures, which are beneficial to symmetric convolutions when the number of taps is the multiple of 2 or 3. 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. 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.

6. Future Work

In this paper, we have provided new parallel FIR structures consisting of advantageous polyphase decompositions dealing with symmetric convolutions comparatively better than the Proposed FFA structures in Modified Adder circuit using CSA in BEC changed reduce the Area and Power in terms of hardware consumption.

References