A Review on Area Efficient Parallel FIR Digital Filter Implementation

Arunadevi A¹, Chitra K., GunaNandhini S., Raghupathi T., Rejusha M²

KSR College of Technology, Tiruchengode

ABSTRACT: Digital signal processing (DSP) applications use various types of filters among which, digital parallel FIR filters are very widely used in various applications. Now a day’s implementation of digital FIR filter using DSP technique has several practical difficulties such as high delay and low speed. To overcome these practical difficulties, the parallel FIR digital filters are implemented using multipliers in VLSI. But implementation using multipliers in VLSI increases the hardware cost. So, to provide low power consumption, low area, high speed and low delay, the multipliers are replaced using adders. Exchanging of multipliers with additional adders is more beneficial because adders are less in weight in terms of silicon area and thus hardware implementation can made simpler.

Keywords: Digital Signal Processing (DSP), Fast Finite Impulse Response (FIR) Algorithms (FFA), Parallel FIR, Very Large Scale Integration (VLSI).

I. INTRODUCTION

Due to the explosive growth of multimedia application, the demand for high performance and low power DSP is getting higher and higher. Most widely used fundamental device performed in DSP system is FIR digital filter [3]. The techniques involved in DSP are filtering, convolution and transformations. In this paper we have presented a new parallel FIR filter based on FFA algorithm in which the multipliers are replaced by adders. FFA can reduce the amount of multiplication in the sub-filter section. In FIR filters, multiplier plays a main role. Many methods have to be done to reduce the power dissipation in FIR filter. The power consumption and hardware cost is very high in VLSI [1]. For this reason we now provide the adder instead of multiplier. The advantage of exchanging multiplier with adder is less weight in terms of silicon layer. 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.

Now we are implementing the parallel FIR filter using VHDL in FPGA kit. In VLSI, the design of an integrated circuit in terms of power, area and speed has become a very challenging problem. Our project is about the improvement and optimization of the algorithm aiming at the problems of the configuration in the coefficient of FIR filter, the storage resource and the calculating speed, which make the memory size smaller and the operation speed faster, to improve the computational performance [3].

II. OVERVIEW OF THE FFA ALGORITHM

Consider an N-tap FIR filter that can be expressed in the general form as [1].

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

Where \( n \) is an infinite length input sequence and \( h(i) \) represents the length of \( N \)-FIR filter coefficients. Then, the traditional L-parallel FIR filter can be derived using polyphase decomposition [5].

\[ \sum_{P=0}^{L-1} Y_P(Z^L) (Z^L)^P = \sum_{q=0}^{N-1} X_q(Z^L) \sum_{r=0}^{L-1} H_r(Z^L) Z^r \ldots \ldots (2) \]

A1. 2*2 TRADITIONAL:

From this fig (1), the equation is expressed as [3]
\[ Y_0 = X_0H_0 + X_1H_1 \]
\[ Y_1 = X_0H_1 + X_1H_0 \] ........... (3)

Fig. 1 Traditional two parallel FIR filter

**A2.3*3 TRADITIONAL:**

From this fig (2), the equation is expressed as

\[ Y_0 = X_0H_0 + X_1H_2 + X_2H_1 \]
\[ Y_1 = X_1H_0 + X_0H_1 + X_2H_2 \]
\[ Y_2 = X_2H_0 + X_1H_1 + X_0H_2 \] ........... (4)

Fig. 2 Traditional three parallel FIR filter

**B1.2*2 FFA (L = 2):**

A two-parallel FIR filter can be expressed as [1]

\[ Y_0 = H_0X_0 + H_1X_1 \]
\[ Y_1 = (H_0+H_1)(X_0+X_1)-H_0X_0H_1X_1 \] ........... (5)

Fig. 3 Two parallel FIR filter using FFA

The implementation of this equation will require three FIR sub filter blocks of length N/2, one pre-processing 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.

**B2.3*3 FFA (L = 3):**

By the similar approach, a three-parallel FIR filter using the FFA can be expressed as [1]

\[ Y_0 = H_0X_0 + H_1X_2 + X_1^2(H_1+H_2)(X_1+X_2)-H_1X_1 \]
\[ Y_1 = [(H_0+H_1)(X_0+X_1)-H_1X_1][(H_0X_0+H_2X_2)]] \]
\[ Y_2 = [(H_0+H_1+H_2)(X_0X_1+X_2)]-[(H_0H_2)(X_0+X_1)+H_1X_1]-[(H_1+H_2)(X_1+X_2)-H_1X_1] \] ........... (6)

The hardware implementation of (6) requires six length N/3 FIR sub-filter blocks, three pre-processing 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. 4.
III. PROPOSED FFA ALGORITHM

The main objective of the proposed structures is to earn the 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 both odd and even 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 sub-filter blocks that contain symmetric coefficient times half the number of multiplications in a single sub-filter block (N/2L). Now we are implementing the parallel FIR filter using VHDL in FPGA kit. The advantage of VHDL, when used for systems design, it allows the behaviour of the required system to be described and verified before synthesis tools translate the design into real hardware (gates and wires). A VHDL project is multipurpose. Being created once, a calculation block can be used in many other projects. A VHDL project is portable. Being created for one element base, a computing device project can be ported on another element base, for example VLSI with various technologies.

C1.2*2 PROPOSED FFA (L=2):

When it comes to a set of even symmetric coefficients, this can earn one more sub filter block containing symmetric coefficients then the existing FFA parallel FIR filter [3]. Fig.5 shows implementation of the proposed two-parallel FIR filter.

\[ Y_1 = \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] \] …(7)

C.3*3 PROPOSED FFA (L=3):

With the similar approach, from (6), a three-parallel FIR filter can also be written as (8). When the number of symmetric coefficients N is the multiple of 3, the proposed three-parallel FIR filter structure presented in (8) 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 [1]. Fig.6 shows implementation of the proposed three-parallel FIR filter.

\[ Y_0 = \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) + (H_0 - H_1)(X_0 - X_1)] - H_1X_1 + \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] \] \[ Y_1 = \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] \] \[ Y_2 = \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) + (H_0 - H_1)(X_0 - X_1)] + H_1X_1 \] \[ Y_3 = \frac{1}{2} [(H_0 + H_1)(X_0 + X_1) - (H_0 - H_1)(X_0 - X_1)] + H_1X_1 \] …… (8)

IV. COMPLEXITY ANALYSIS AND COMPARISON

When the number of symmetric coefficients N is the multiple of 3, the proposed three-parallel FIR filter structure presented 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. A comparison figure is shown in Fig. 6, where the shadow blocks stand for the sub filter blocks which contain symmetric coefficients.

<table>
<thead>
<tr>
<th>Existing FFA</th>
<th>Proposed FFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_0$</td>
<td>$\frac{1}{2} (H_0+H_1)$</td>
</tr>
<tr>
<td>$H_1$</td>
<td>$\frac{1}{2} (H_0-H_1)$</td>
</tr>
<tr>
<td>$H_2$</td>
<td>$\frac{1}{2} (H_0+H_2)$</td>
</tr>
<tr>
<td>$H_0+H_1$</td>
<td>$\frac{1}{2} (H_0+H_2)$</td>
</tr>
<tr>
<td>$H_1+H_2$</td>
<td>$H_1$</td>
</tr>
<tr>
<td>$H_0+H_1+H_2$</td>
<td>$H_0+H_1+H_2$</td>
</tr>
</tbody>
</table>

Fig.7 comparison of sub filter blocks between existing FFA and proposed FFA 3-Parallel FIR structures

Therefore, for an N-tap three-parallel FIR filter, the proposed structure can save N/3 multipliers from the existing FFA structure. However, again, the proposed three-parallel FIR structure also brings an overhead of seven additional adders in pre-processing and post processing blocks. The number of sub filter blocks didn’t increase in our new approach.

**V. EXPERIMENTAL RESULT**

We are implementing the proposed FFA structures and the existing FFA structures using in VHDL. Table shows the result in terms of area and power using SYNOPSYS – Design Compiler by 90nm technology.

<table>
<thead>
<tr>
<th>Length</th>
<th>Structure</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>L=2</td>
<td>Traditional</td>
<td>15238</td>
</tr>
<tr>
<td></td>
<td>FFA</td>
<td>13680</td>
</tr>
<tr>
<td></td>
<td>FFA with additional adders</td>
<td>4929</td>
</tr>
<tr>
<td>L=3</td>
<td>Traditional</td>
<td>15925</td>
</tr>
<tr>
<td></td>
<td>FFA</td>
<td>15035</td>
</tr>
<tr>
<td></td>
<td>FFA with additional adders</td>
<td>5135</td>
</tr>
</tbody>
</table>

Table1. Comparison of area

<table>
<thead>
<tr>
<th>Length</th>
<th>Structure</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L=2</td>
<td>Traditional</td>
<td>18.37</td>
</tr>
<tr>
<td></td>
<td>FFA</td>
<td>8.9</td>
</tr>
<tr>
<td></td>
<td>FFA with additional adders</td>
<td>5.4</td>
</tr>
<tr>
<td>L=3</td>
<td>Traditional</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>FFA</td>
<td>12.4</td>
</tr>
<tr>
<td></td>
<td>FFA with additional adders</td>
<td>7.2</td>
</tr>
</tbody>
</table>

Table2. Comparison of power

In these comparison tables, the area and the power will be reduced from traditional filter structures. The number of taps will increase means the more amount of area and power will decrease. We done this using at very basic taps like 2-taps and 3-taps so less amount of values were reduced. Also we had the table for the numbers of multipliers were reduced and the numbers of adders were increased from previous filter structures.

By using SYNOPSYS tool
Table 3 Comparison of no of multipliers and adders

<table>
<thead>
<tr>
<th>TAP</th>
<th>2TAP</th>
<th>3TAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRADITIONAL</td>
<td>4M</td>
<td>2A</td>
</tr>
<tr>
<td>FFA</td>
<td>3M</td>
<td>4A</td>
</tr>
<tr>
<td>FFA WITH ADDITIONAL ADDERS</td>
<td>2M</td>
<td>6A</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

In this paper, we have presented a parallel FIR filter structure, which are beneficial to symmetric convolutions when the number of taps is the multiple of 2 or 3 [3]. Multipliers are the major portions in hardware consumption for the parallel FIR filter implementation. The proposed 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. Overall, in this paper, we have provided a parallel FIR structures consisting of advantageous polyphase decompositions dealing with symmetric convolutions comparatively better than the existing FFA structures in terms of hardware consumption [1]. The area and the power of these filters were analysed using Synopsys-Design Compiler. The number of taps will increase means the reduction of multipliers will increase and also the area and power will reduce.

REFERENCES


