Adaptive Filter Architecture for FPGA Implementations

Kalaiselvi, Vasuki

Abstract: Adaptive filters play a significant role in digital signal processing but their implementation in real-time consumes high area and power. Several architectures have been proposed for their real-time implementation such as Distributed Arithmetic, CORDIC, Systolic, etc., which reduces the area and improves the speed. All these algorithms are multiplier less and among these, the CORDIC structure is simple and gives reduction in area at the cost of speed. To overcome this drawback, it is modified by implementing it along with Karatsuba algorithm (KA). The combination of KA algorithm and CORDIC structure gives better performance in terms of area and speed. The proposed work is implemented using Xilinx System Generator. The structure is tested for different bit representations and the results show that the proposed structure has better performance compared to the existing structures. The proposed structure can be used in applications such as RADAR, Channel Equalizers and Noise Cancellers.

Keywords: Adaptive filter, FPGA, CORDIC, KA algorithm

I. INTRODUCTION

An adaptive filter is defined as a system consisting of linear filter with its transfer function being controlled by a variable parameter using an optimization algorithm. It is considered as a computational device that models the relation between the two signals iteratively. Adaptive filter are usually realized using a set of instructions that runs on devices like microprocessor, DSP chip or a field programmable array (FPGA). The different classes of adaptive filters are LMS and RLS filters. LMS adaptive filters, have various advantages such as flexibility and programmability, good convergence behavior, and computational simplicity, hence it may be easier for it to be implemented using Field Programmable Gate Arrays (FPGAs).

Adaptive filters [1] are used in wide applications in areas like Digital signal processing (DSP), Communications, system identification, noise cancellation and echo cancellation. This paper proposes a new architecture for implementing the adaptive filter using FPGAs. The proposed architecture uses CORDIC structures and KA algorithm for its implementation. CORDIC (Coordinate Rotational Digital Computer) Structures are used widely in computing several mathematical and digital signal processing operations. Few such operations include computation Trigonometric, hyperbolic exponential, multiplication operations and DSP operations like [17][18] Discrete Cosine Transform, Fourier Transform. It is efficient because of its simple shifter and adder architecture [1].

Due to the above reasons CORDIC occupies less space and has less computation time. KA algorithm is known for its high speed operation with reduced chip area [2], [14]. In the proposed architecture the CORDIC structure is adopted in KA algorithm, thus improving the system throughput and reducing the chip area. This proposed architecture is implemented using VHDL.

Adaptive filter algorithms are widely used in linear filters. The two adaptive algorithms for adaptive filter are Least Mean Square (LMS) and Recursive Least square (RLS). The output signal \( z(n) \) obtained from the filter is given by

\[
z(n) = x(n) * w(n)
\]

Where \( n = 0, 1, 2, 3 \ldots N \). \( x(n) \) refers to the input signal, \( w(n) \) refers to the filter co-efficient, and \( z(n) \) refers to the output signal.

The error signal of the filter is given by

\[
e(n) = z(n) - d(n)
\]

Where \( e(n) \) refers to the error signal and \( d(n) \) refers to the desired signal.

The filter coefficients are updated by the

\[
w(n+1) = w(n+1) + \alpha E(n) * x(n)
\]

The LMS algorithm has less number of operations compared to the RLS algorithm.

II. CORDIC STRUCTURES

CORDIC structure can be used to compute different signal processing operations like DCT, FFT, and mathematical operations such as calculation of sine and cosine values using only adders and shifters. CORDIC can be implemented in three different structures namely Sequential, parallel and pipelined architecture. Among the three architecture, Parallel architecture has high speed. Hence it is widely used in RADAR and Bio medical applications.

Using CORDIC the multiplication operation is carried out in a multiplierless environment. The product is obtained from the CORDIC multiplier by shifting one of the operands and the other operand to zero [1],[4], [14]. The CORDIC multiplication is carried out as

\[
C = A * B
\]
Where \( C \) is the product and A, B are the operands of the multiplier. Then \( C \) can be represented as

\[
C = B \ast \sum_{i=1}^{n} a_i \ast 2^{-i}
\]

(5)

\[
= \sum_{i=1}^{n} B \ast a_i \ast 2^{-i}
\]

(6)

\[
= \sum_{i=1}^{n} a_i \ast B \ast 2^{-i}
\]

(7)

From above equation it is seen that the product \( C \) comprises of shifted versions of the operand \( B \). The coefficients \( a_i \) is determined by making \( x \) to 0 one bit at a time. If the \( i^{th} \) bit position of \( A \) is non-zero then \( B \) is right shifted by \( i \) bits and adding the Value of \( K \) to it. The \( i^{th} \) bit is then removed from \( A \) by subtracting \( 2^{i} \) from \( A \). On the other hand if the value of \( A \) is negative then the \( i^{th} \) bit in the two's complement format is removed by adding \( 2^{i} \).

A. Ka Algorithm

The KA algorithm is a recursive algorithm which is based on the divide and conquer method. The larger operands are divided into K-small number and product is calculated. The KA algorithm can be represented using two ways. They are 2-way and 3-way algorithm.

ALGORITHM – KA 2 Way Structure:

Let \( X \) and \( Y \) represent the two operands of the multiplier. They are expressed as

\[
X = X_0 + X_1 M^{N/2}
\]

(8)

\[
Y = Y_0 + Y_1 M^{N/2}
\]

(9)

Let \( Z \) be the product of \( X \) and \( Y \). Then \( Z \) computed by \( KA \) equation as

\[
Z = K'_0 + (K_{01} - K_0 - K_1) D^{N/2} + K_1 D^M
\]

(10)

Where

\[ M \] represents the number of bits
\[ D \] represents the base

\[
K_0 = X_0 Y_0,
\]

(11)

\[
K_1 = X_1 Y_1,
\]

(12)

\[
K_{01} = (X_0 + X_1)(Y_0 + Y_1)
\]

(13)

The above multiplication is done by CORDIC structures. The entire operation is done without multiplier. The multiplication is performed for smaller operand. Hence the multiplication complexity is reduced from \( O(n^2) \). Also the overall critical delay is improved.

ALGORITHM – KA 3 Way Structure:

The multiplication is carried out by dividing the operand into three parts. Let the two multiplication operands be \( A \) and \( B \). The two operands are then expressed as

\[
A = A_0 + X^{n/3} A_1 + X^{2N/3} A_2
\]

(14)

\[
B = B_0 + X^{n/3} B_1 + X^{2N/3} B_2
\]

(15)

Let \( C \) be the product by multiplying \( A \) and \( B \). Then \( C \) is expressed using 3-way as

\[
C = C_0 + C_1 X^{N/3} + C_2 X^{2N/3} + C_3 X^N + C_4 X^{4N/3}
\]

(16)

Where

\[
C_0 = P_0
\]

\[
C_1 = P_{01} - P_0 - P_1
\]

(17)

\[
C_2 = P_{02} - (P_0 + P_2) - P_2
\]

(18)

\[
C_3 = P_{12} - (P_1 + P_2)
\]

(19)

\[
C_4 = P_2
\]

(20)

\[
P_0 = A_0 B_0
\]

(21)

\[
P_1 = A_1 B_1
\]

(22)

\[
P_{01} = (A_0 + A_1)(B_0 + B_1)
\]

(23)

\[
P_{02} = (A_0 + A_2)(B_0 + B_2)
\]

(24)

\[
P_{12} = (A_1 + A_2)(B_1 + B_2)
\]

(25)

III. ARCHITECTURE

The Adaptive algorithm is used in situations when it is necessary to get the desired signal inspite of the error signal. The weights are updated as per equation

\[
w(n+1) = W(n) + \mu[d(n) - z(n)]\times x(n)
\]

(26)

If MSE gradient is positive, then weights are updated as per equation

\[
w(n+1) = W(n) - \mu[d(n) - z(n)]\times x(n)
\]

(27)

The negative sign indicates that the weight have to be updated in the direction opposite to the gradient slope.

The overall block diagram of the proposed architecture is shown below. The difference between the actual response and the desired response is computed. The difference is then multiplied by the input signal. The multiplications in the above update equations are carried out using the KA algorithm described in section 1. Further the inner multiplication in the KA decomposition is done using the CORDIC structure. Since the CORDIC structure is based on the shift and add operations the proposed architecture doesn’t have multipliers. The KA algorithm using 2-way decomposition is shown in fig.2. The structure consists only of adders and shift registers. The error signal and the input signal are the two operands (inputs) of the proposed architecture. The two operands are split into two equal parts as per equations 8-9. The partial products are calculated as per equations 10 – 13.
Fig. 1. Architecture of the proposed structure

The partial products resulting out of the multiplications (equations 11 – 13) are carried out using CORDIC structures. The multiplication using CORDIC is done wide equations 4-7. By this process the large valued operands are decomposed into small values and the product is computed. Hence the computation complexity is reduced. Also the proposed architecture involves only adders and shifters, hence the hardware complexity is also less.

Fig. 2. Architecture of CORDIC incorporated KA 2-way Multiplier structure

Similarly the KA 3-way structure is framed. The major advantages is that the higher valued operands are decomposed into very small values compared to the KA 2-way structure. The architecture of KA 3-way structure is shown in fig 3. Similar to the KA 2-way decomposition the KA 3 way is also done. First the two operands are divided into two equal parts (equations 14,15). Each part is decomposed into three parts and then the partial products are computed (equations 16 – 25). The multiplication operation involved in computing the partial products (21–25) is done with the help of CORDIC structures. Since the decomposed values are small the computation complexity of this structure is less compared to the @-way decomposition.

IV. SIMULATION RESULT

A CORDIC incorporated KA FIR adaptive filter is implemented using VHDL. On reset signal the output is set to zero. The initial weights are obtained and fed to the filter. The weight is convoluted with the input signal and the output is produced. The actual output is compared with the desired output and the weights are updated according to the proposed architectures for each clock cycle.
Adaptive Filter Architecture for Fpga Implementations

The results are obtained for three different structures CORDIC, KA 2-way with CORDIC, KA 3-way with CORDIC. The three structures are simulated for different bit lengths such as 16 bit, 24 bit, 32 bit, 48 bits. Table 1 and 2 shows the various parameters obtained for different bit combinations. Table 1 shows the performance comparison between CORDIC structure and the KA 2-way with CORDIC structure. The later structure occupies less number of slices and number of LUTs. The delay produced by the KA 2-way structure is less compared to the CORDIC structure because the value of the operands are decomposed and computed. Hence this structure is efficient in terms of hardware and computation time.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Architecture of CORDIC Adaptive filter</th>
<th>Architecture of CORDIC Incorporated KA 2-way Adaptive FIR filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices Utilized</td>
<td>28 33 80</td>
<td>20 28 76</td>
</tr>
<tr>
<td>four input LUT Utilized</td>
<td>50 61 158</td>
<td>39 51 146</td>
</tr>
<tr>
<td>Bounded IOB Utilized</td>
<td>56 74 110</td>
<td>56 74 110</td>
</tr>
</tbody>
</table>

**TABLE 1**
Performance Comparison of CORDIC and KA 2-way Structure

**Fig. 4. Performance Comparison of CORDIC Architecture and KA 2-way with CORDIC architecture**

The Table 2 shows the performance analysis of KA-2 way with CORDIC and KA 3-way with CORDIC structures. The KA 3-way with CORDIC shows better simulation results compared to the other structures. Also this structure better results when the input data has very large values. This structure is also best in terms of hardware utilization and computation time.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Two way Architecture</th>
<th>Three way Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices Utilized</td>
<td>20 28 76</td>
<td>53 62</td>
</tr>
<tr>
<td>four input LUT Utilized</td>
<td>39 51 146</td>
<td>100 108</td>
</tr>
<tr>
<td>Bounded IOB Utilized</td>
<td>56 74 110</td>
<td>110 220</td>
</tr>
<tr>
<td>Time Delay (ns)</td>
<td>9.954 9.993 17.762</td>
<td>11.384 14.357</td>
</tr>
</tbody>
</table>

**TABLE 2**
Performance Comparison of KA 2-way and KA 3-way Structure
Fig.5. Performance Comparison of CORDIC with KA 2 way and 3 way structures.

A. RTL Schematic Of The Proposed Strucutre

The RTL schematic of the structure proposed is shown in figure 6. X1,X2,X3 represent the input signal, d0,d1,d2,represents the desired signal and Y1,Y2,Y3,represents the output signal.

Fig. 6. RTL schematics of FIR adaptive filter

V. CONCLUSION

In this paper multiplier less architecture for implementing adaptive filter was presented. From the simulation results it is found that the KA 2 way and KA 3 way with CORDIC algorithm provides a better performance in terms of hardware complexities and time delay. Also the KA 3 way with CORDIC algorithm shows better performance for high value of input data. Thus this architecture could be used in various applications like noise cancellers, equalizers, echo cancellers.

REFERENCES