Memory Optimization in Adaptive FIR Filter Using APC-OMS

E. Suruthi1, D. Sowmiya2

1P.G Scholar, Department of Electronics and communication Engineering
Sri Muthukumaran Institute of Technology, Chennai, Tamilnadu, India

2Assistant Professor, Department of Electronics and communication Engineering
Sri Muthukumaran Institute of Technology, Chennai, Tamilnadu, India

Abstract: An efficient architecture for the implementation of Adaptive FIR Filter for achieving optimized memory and increased accuracy. Besides, we have described the Adaptive FIR filter where filter coefficients are frequently updated in order to minimize the error out. The combined Anti-symmetric Product Coding (APC) and Modified Odd Multiple Storage (OMS) technique to achieve the reduction of LUT size to Third-fourth of the conventional technique. The APC-OMS based Filter design to reduce the LUT size and the errors are eliminated at the output of the adaptive fir filter. Simulation results show good accuracy of the input signal.

Keywords: Adaptive Filters, Memory optimization, APC, OMS

1. Introduction

ADAPTIVE FIR filters have a wide range of communication and DSP applications such as adaptive equalization, system identification and image restoration[10]-[4]. The direct-form LMS [2], adaptive filter involves a long critical path due to an inner-product computation to obtain the filter output. The critical path is required to be reduced by pipelined implementation when it exceeds the desired sample period. The conventional LMS algorithm does not support pipelined frequency but, they involve implementation because of its recursive behaviour, it is modified to a form called the delayed LMS (DLMS) algorithm [5], which allows pipelined implementation of the filter.

A lot of work has been done to implement the DLMS algorithm in systolic architectures to increase the maximum usable frequency [3], but they involve an adaptation delay. A systolic architecture, where they have used relatively large processing elements (PEs) [3], for achieving a lower adaptation delay with the critical path of one MAC operation. Memory-based computing systems are more regular than the multiply-accumulate structures, and well suited for many digital signal processing (DSP) algorithms, which involve multiplication with fixed set of coefficients [7].

To reduce the pipelined delays[6]. In adaptive fir filter coefficients are not fixed. The existing work on the fixed point LMS adaptive filter [9], does not discuss the APC-OMS approach. In this fixed point implementation using a novel partial product generator (PPG).

So it can increase the area. We have referred to this as odd-multiple-storage (OMS) scheme and anti-symmetric product coding [8]. In this paper, we propose a combined APC-OMS technique, it could be reduced the LUT to Third-Fourth of the conventional LUT size. Since the approach area, time, delay and power reduced compare the canonical sign digit (CSD) [1], multiplier.

2. Review of Fixed Point Implementation

In this Existing system we will use the implementation of a delayed least mean square (DLMS) adaptive filter [5]. For achieving lower adaptation-delay and area-delay-power efficient implementation, use a novel Partial Product Generator (PPG) [9]. The Fixed point LMS Adaptive filter implementation is used to reduce the number of pipeline delays along with area, sampling period and energy consumption. The design more efficient in terms of the Power Delay Product (PDP) and Energy Delay Product (EDP).

![Figure 1: Existing system error computation block](image_url)
b digit ($u_1u_0$) as input and produces three outputs. Each AOC consists of three AND cells and two OR cells. It provides nearly 20% saving in the ADP and 9% saving in EDP.

3. Proposed Architecture

The block diagram of the Adaptive FIR filter and its building blocks are shown in Figure 2. In addition, various circuits have been proposed for each module.

![Figure 2: Proposed block diagram](image)

In this block diagram the APC-OMS plays the major role. The input signal is applied to the filter, it removes the unwanted signal and the output send to the comparator. The comparator compares the desired signal and the input signal. Then the error output send to the APC-OMS. The output of APC-OMS again send to the filter and the output send them out. The frequency response realized in the time domain is of more interest for FIR filter realization (both hardware and software).

3.1 Adder Tree

In the RCA (Ripple Carry Adder) method, two inputs are added from LSB to MSB where each carry is added with forthcoming bits. It increases propagation delay. In the parallel adder method, Both sum and Carry are generated in same time cycle using XOR and AND gates. The carry zero(0) to be stored in the Parallel Adder(PA) and the carry to be one(1) to be stored in the BEC(Binary Excess Code).Multiplexer is used for multiplication operation. In this adder tree is used to reduce the computation time. So the power is saved.

![Figure 3: Adder block diagram](image)

3.2 Shift/Add Tree

The shift/add tree is placed on the filter convolution using mux. The convolution operation means it can perform the multiplication operation. If the two addresses are multiplied number of operation increased but in the shift/add tree quickly perform the operation.

Example:

\[
\begin{array}{c}
0010(2) \\
0100(4) = 1000(8)
\end{array}
\]

![Figure 4: Shift/Add Tree block diagram](image)

3.3 Coefficient Calculation

Then the output \(y(n)\) is the combination of input signal \(x(n)\) and the coefficient \(b(k)\).Both multiplied and the next iteration delay is added for the coefficient calculation.

![Figure 5: Coefficient Calculation diagram](image)
4. The Apc-Oms Technique and Implementation

In the APC-OMS technique, the LUT tables size is reduced to third-fourth of the conventional LUT. The APC-OMS block diagram shown in the Fig. A conventional lookup-table (LUT)-based multiplier is shown in Fig. In this $A$ is a fixed coefficient, and $X$ is an input word to be multiplied with $A$. Assuming $X$ to be a positive binary number of word length $L$, it can be $2^L$ possible values of $X$, and there can be $2^L$ possible values of product $C = A \cdot X$ or $A = C/X$. Then the memory-based multiplication, $2^L$ words of Look Up Table, consisting of pre-computed product values corresponding to all possible values of $X$, is conventionally used.

A. APC for Lut Optimization

For simplicity of presentation, we assume the positive integers $X$ and $A$. The APC words for the different values of $X$ for $L = 5$ are shown in Table I. In this table the two’s complement of the third column of the same row input word $x$ on the first column of each row. Then also the sum of product values corresponding to these two input values $32A$ is on the same row. Then the product values on the 2nd and 4th columns of a row be $u$ and $v$ values. Then the product values on the second and fourth columns of Table I therefore have a negative mirror symmetry. This behaviour of the product words can be used to reduce the Look Up Table size, instead of storing $u$ and $v$, only $[(v - u)/2]$ is stored for a pair of input on a given row. Since one can write $u = [(u + v)/2 - (v - u)/2]$ and $v = [(u + v)/2 + (v - u)/2]$, for $(u + v) = 32A$, we can have

$$u = 16A - \left[\frac{v - u}{2}\right]$$

$$v = 16A + \left[\frac{v - u}{2}\right].$$

The 4-bit LUT addresses and corresponding coded words are listed on the fifth and sixth columns. Then the representation of the product is derived from the antisymmetric [8], behaviour of the products, we can name it as antisymmetric product code. The 4-bit address $X_\prime = (x_3x_2x_1x_0)$ of the APC word is given by

$$X_\prime = \begin{cases} 
X_L, & \text{if } x_A = 1 \\
X_L, & \text{if } x_A = 0
\end{cases}$$

where $X_L = (x_3x_2x_1x_0)$ is the four less significant bits of $X$, and $X_L$ is the two’s complement of $x_3x_2x_1x_0$. The desired APC could be obtained by adding or subtracting the stored value $(v - u)$ to or from the fixed value $16A$ when $x_4$ is 1 or 0 in the MSB.

Figure 6: APC-OMS block diagram

Figure 7: Proposed APC-OMS combined LUT

Product word = $16A + (\text{sign value}) \times (\text{APC word})$, where $\text{sign value} = 1$ for $x_4 = 1$ and sign value = $-1$ for $x_4 = 0$. The product value for $X = (10000)$ corresponds to APC value “zero,” which could be derived by resetting the Look Up Table output and also stored on the Look Up Table.

Figure 8: LUT-based multiplier for $L = 5$ using the APC technique
B. Modified OMS for LUT Optimization

Then the multiplication of size \( L \), of any binary word \( X \), with a fixed coefficient \( A \), instead of storing all the \( 2^L \) possible values of \( C = A \cdot X \), the odd multiples of \( A \) may be stored in the LUT only \( (2^L/2) \) words, while the left-shift operations of one of those odd multiples [8], by all the even multiplies of \( A \) could be derived.

In Table II, we have shown that, at eight memory locations, the eight odd multiples, \( A \times (2i+1) \) are stored as \( P_i \), for \( i = 0, 1, 2, \ldots, 7 \). The left-shift operations of \( A \), then the even multiples \( 2A, 4A, \) and \( 8A \) are derived. Similarly, left shifting \( 3A \) then the \( 6A \) and \( 12A \) are derived, while left shifting \( 5A \) and \( 7A \), then the \( 10A \) and \( 14A \) are derived. A barrel shifter for producing a maximum of three left shifts could be used to derive all the even multiples of \( A \). It may be seen from Tables II and III that the 5-bit input word \( X \) can be mapped into a 4-bit LUT address \( (d_3d_2d_1d_0) \), by a simple set of mapping relations \( d_i = x \cdot i+1 \), for \( i = 0, 1, 2 \) and \( d_3 = x_0 \) 'where \( X'' = (x''3x''2x''1x''0) \) is generated by shifting-out all the leading zeros of \( X_0 \) by an arithmetic right shift followed by address mapping.

\[
X'' = \begin{cases} 
Y_L, & \text{if } x_1 = 1 \\
Y'_L, & \text{if } x_1 = 0 
\end{cases}
\]

where \( Y_L \) and \( Y'_L \) are derived by circularly shifting-out all the leading zeros of \( X_L \) and \( X'_L \), respectively.

Table 3: Reduced Apc-Oms Address

In Figure 9, we have shown that, at eight memory locations, the eight odd multiples, \( A \times (2i+1) \) are stored as \( P_i \), for \( i = 0, 1, 2, \ldots, 7 \). The left-shift operations of \( A \), then the even multiples \( 2A, 4A, \) and \( 8A \) are derived. Similarly, left shifting \( 3A \) then the \( 6A \) and \( 12A \) are derived, while left shifting \( 5A \) and \( 7A \), then the \( 10A \) and \( 14A \) are derived. A barrel shifter for producing a maximum of three left shifts could be used to derive all the even multiples of \( A \). It may be seen from Tables II and III that the 5-bit input word \( X \) can be mapped into a 4-bit LUT address \( (d_3d_2d_1d_0) \), by a simple set of mapping relations \( d_i = x \cdot i+1 \), for \( i = 0, 1, 2 \) and \( d_3 = x_0 \) 'where \( X'' = (x''3x''2x''1x''0) \) is generated by shifting-out all the leading zeros of \( X_0 \) by an arithmetic right shift followed by address mapping.

\[
X'' = \begin{cases} 
Y_L, & \text{if } x_1 = 1 \\
Y'_L, & \text{if } x_1 = 0 
\end{cases}
\]

where \( Y_L \) and \( Y'_L \) are derived by circularly shifting-out all the leading zeros of \( X_L \) and \( X'_L \), respectively.
4. Simulation Results

Simulations have been performed using Modelsim 6.4a Simulation tool technology in. Fig.10 shows the input and output waveform results for proposed APC-OMS technique. Fig.9. shows the area analyzer summary. Proposed APC-OMS provide good accuracy of the input signal. The results of this proposed design reduced the LUT size into third-fourth of the conventional LUT size. By this we can clearly decide that the proposed circuit can have lower area overhead than the other conventional circuits. From the results, it is clear that the proposed circuit can have very less power.

5. Conclusion and Future Work

Analyzing the APC-OMS FIR architecture and performance, this paper presents a new architecture for the LUT. The proposed architecture applies the main concept of the basic APC-OMS technique in implementing the MAC unit and at the same time has many advantages over its basic architecture. The results obtained in the proposed architecture, the LUT size is reduced to Third-Fourth. The proposed architecture can be easily used to implement high order FIR filters e.g. 20-tap with different coefficients wordlength without suffering from large LUT construction and with low hardware complexity needed for the design. The future work is to perform a VLSI implementation of FIR filter for ultra wideband communications.

References