Memory Optimization in Adaptive FIR Filter Using APC-OMS

E. Suruthi[1], D. Sowmiya[2]

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

[2]Assistent 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).

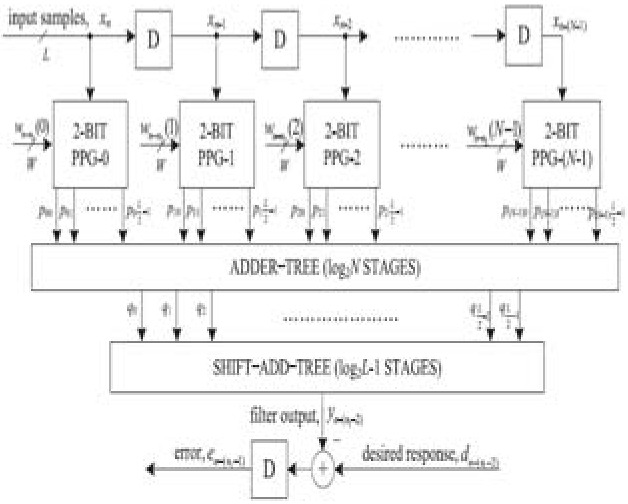


Figure 1: Existing system error computation block

This structure to minimize the adaptation delay in the error-computation block, followed by the weight-update block. The structure for error-computation unit of an *N*-tap DLMS [4], adaptive filter shown in fig.1. It consists of *N* number of 2-b partial product generators (PPG) corresponding to *N* multipliers and a cluster of *L*/2 binary adder trees, followed by a single shift–add tree. The structure of each PPG consists of *L*/2 number of 2-to-3 decoders and the same number of AND/OR cells (AOC).Each of the 2-to-3 decoders takes a 2-

b digit *(u*1*u*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.

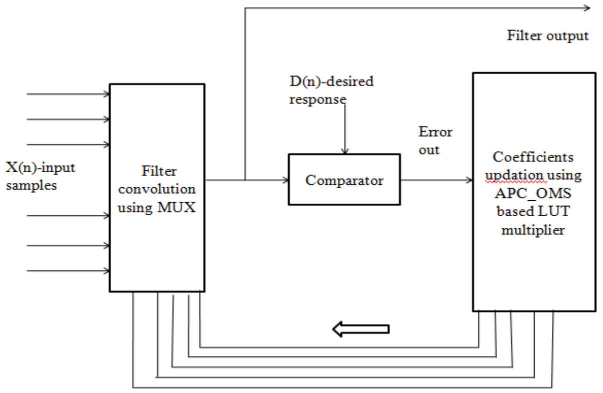

Filter output
D(n)-desired response
Error out
X(n)-input samples
Filter convolution using MUX
Comparator
Coefficients updation using APC_OMS based LUT multiplier

Figure 2: Proposed block diagram

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.

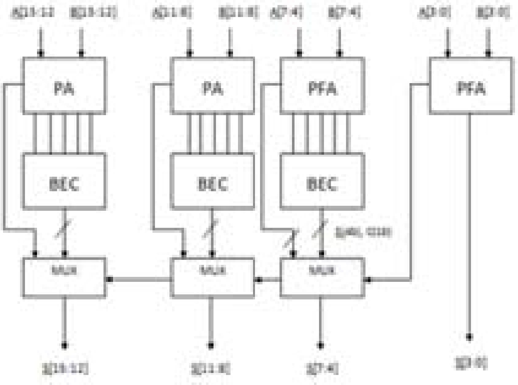


Figure 3: Adder block diagram

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:

```
0010(2)*0100(4)=1000(8)
     0010*0100          0010(2)→ 2 shifts
      0000              1000(8)
     0000
    0010
   0000
  0001000
```

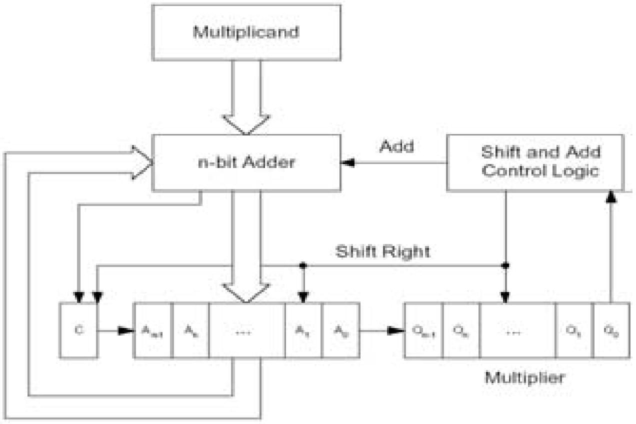


Figure 4: Shift/Add Tree block diagram

3.3 Coefficient Calculation

$$y[n]=\sum_{k=0}^{N-1} b_k \cdot x[n-k]$$

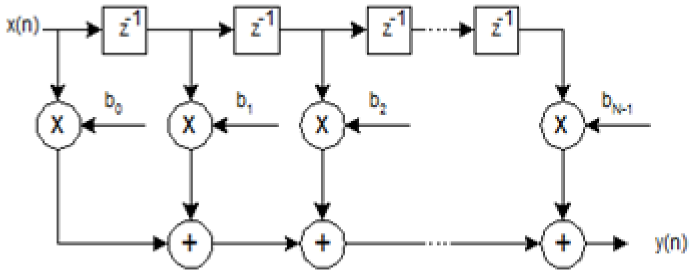


Figure 5: Coefficient Calculation diagram

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.

4. The Apc-Oms Technique and Implementation

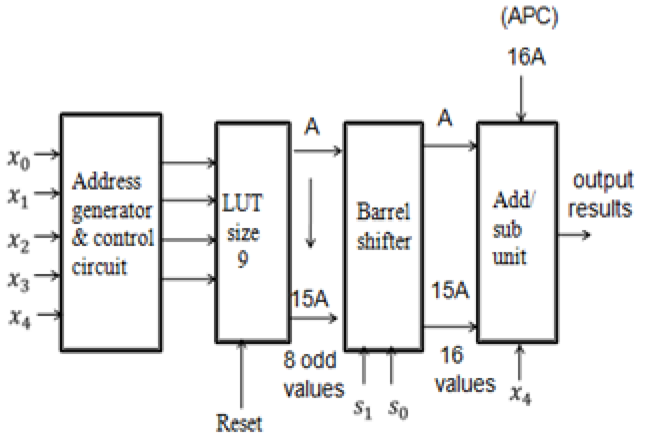


Figure 6: APC-OMS block diagram

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* · *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 32*A* 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*) = 32*A*, we can have

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

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*_ = (*x*_3*x*_2*x*_1*x*_0) of the APC word is given by

$$X' = \begin{cases} X_L, & \text{if } x_4 = 1 \\ X'_L, & \text{if } x_4 = 0 \end{cases}$$

where *XL* = (*x*3*x*2*x*1*x*0) is the four less significant bits of *X*, and *X_L* is the two's complement of *x3x2x1x0*.The desired APC could be obtained by adding or subtracting the stored value (*v* − *u*) to or from the fixed value 16*A* when *x*4 is 1 or 0 in the MSB.

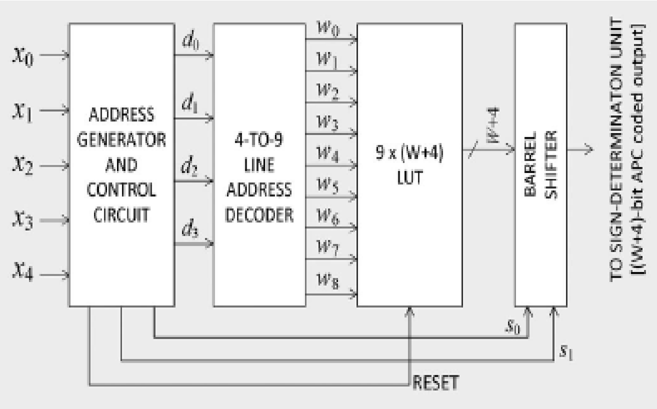


Figure 7: Proposed APC-OMS combined LUT

Product word = 16*A* + (sign value) × (APC word). where 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**.**

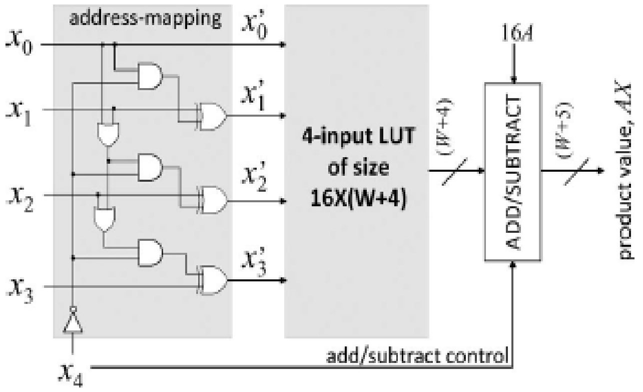
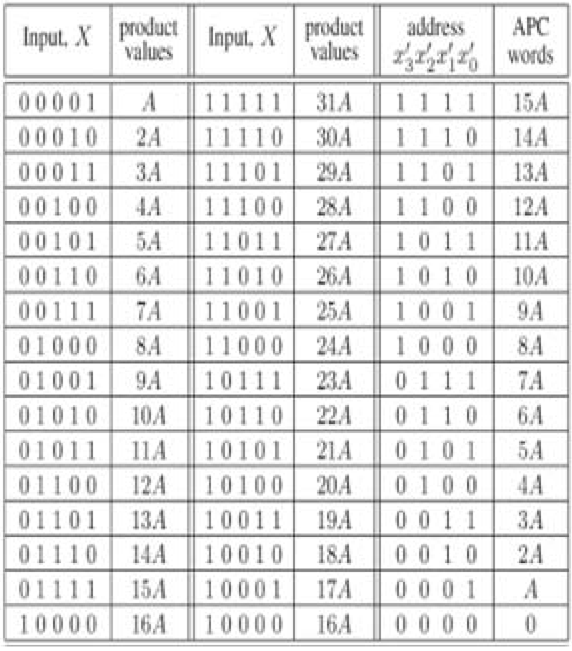


Figure 8: LUT-based multiplier for *L* = 5 using the APC technique

Table 1: Apc Words For Different Input Values

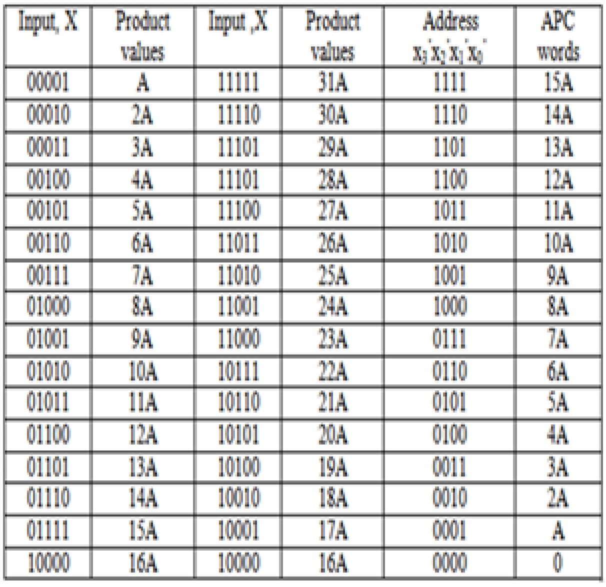


Input, X	product values	Input, X	product values	address $x'_3x'_2x'_1x'_0$	APC words
0 0 0 0 1	A	1 1 1 1 1	31A	1 1 1 1	15A
0 0 0 1 0	2A	1 1 1 1 0	30A	1 1 1 0	14A
0 0 0 1 1	3A	1 1 1 0 1	29A	1 1 0 1	13A
0 0 1 0 0	4A	1 1 1 0 0	28A	1 1 0 0	12A
0 0 1 0 1	5A	1 1 0 1 1	27A	1 0 1 1	11A
0 0 1 1 0	6A	1 1 0 1 0	26A	1 0 1 0	10A
0 0 1 1 1	7A	1 1 0 0 1	25A	1 0 0 1	9A
0 1 0 0 0	8A	1 1 0 0 0	24A	1 0 0 0	8A
0 1 0 0 1	9A	1 0 1 1 1	23A	0 1 1 1	7A
0 1 0 1 0	10A	1 0 1 1 0	22A	0 1 1 0	6A
0 1 0 1 1	11A	1 0 1 0 1	21A	0 1 0 1	5A
0 1 1 0 0	12A	1 0 1 0 0	20A	0 1 0 0	4A
0 1 1 0 1	13A	1 0 0 1 1	19A	0 0 1 1	3A
0 1 1 1 0	14A	1 0 0 1 0	18A	0 0 1 0	2A
0 1 1 1 1	15A	1 0 0 0 1	17A	0 0 0 1	A
1 0 0 0 0	16A	1 0 0 0 0	16A	0 0 0 0	0

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* ・ *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 multiples of A could be derived.

Table 2: Oms-Based Design Of The Lut Of Apc Words



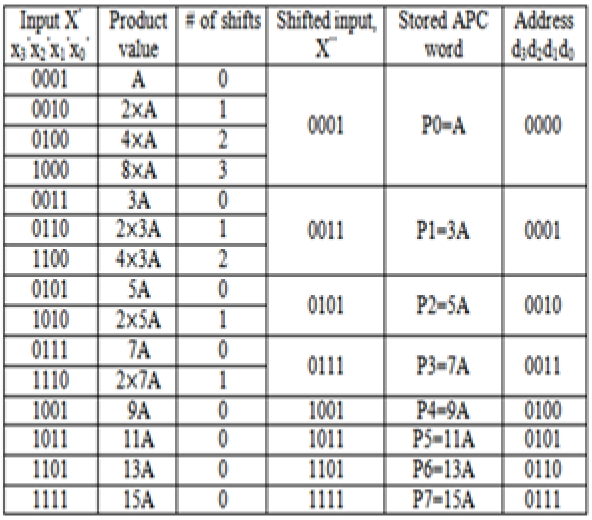
Input, X	Product values	Input ,X	Product values	Address $x_3' x_2' x_1' x_0'$	APC words
00001	A	11111	31A	1111	15A
00010	2A	11110	30A	1110	14A
00011	3A	11101	29A	1101	13A
00100	4A	11101	28A	1100	12A
00101	5A	11100	27A	1011	11A
00110	6A	11011	26A	1010	10A
00111	7A	11010	25A	1001	9A
01000	8A	11001	24A	1000	8A
01001	9A	11000	23A	0111	7A
01010	10A	10111	22A	0110	6A
01011	11A	10110	21A	0101	5A
01100	12A	10101	20A	0100	4A
01101	13A	10100	19A	0011	3A
01110	14A	10010	18A	0010	2A
01111	15A	10001	17A	0001	A
10000	16A	10000	16A	0000	0

In Table II, we have shown that, at eight memory locations, the eight odd multiples, *A* × (2*i* + 1) are stored as *Pi*, for *i* = 0, 1, 2, . . . , 7. The left-shift operations of *A*,then the even multiples 2*A*, 4*A*, and 8*A* are derived.Similarly, left shifting 3*A* then the, 6*A* and 12*A* are derived, while left shifting 5*A* and 7*A*, then the 10*A* and 14*A* 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*3*d*2*d*1*d*0), by a simple set of mapping relations *di* = *x*``*i*+1, for *i* = 0, 1, 2 and *d*3 = *x*0 ‘‘ where *X*’’= (*x*’’3*x*’’2*x*’’1*x*’’0) is generated by shifting-out all the leading zeros of *X*_ by an arithmetic right shift followed by address mapping.

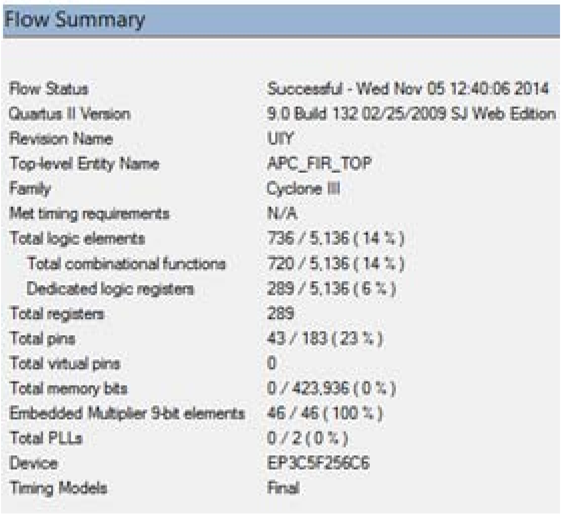
$$X'' = \begin{cases} Y_L, & \text{if } x_4 = 1 \\ Y'_L, & \text{if } x_4 = 0 \end{cases}$$

where *YL* and *Y* _*L* are derived by circularly shifting-out all the leading zeros of *XL* and *X*_*L*, respectively.

Table 3: Reduced Apc-Oms Address



Input X' $x_3' x_2' x_1' x_0'$	Product value	# of shifts	Shifted input, X''	Stored APC word	Address $d_3d_2d_1d_0$
0001	A	0	0001	P0=A	0000
0010	2×A	1			
0100	4×A	2			
1000	8×A	3			
0011	3A	0	0011	P1=3A	0001
0110	2x3A	1			
1100	4x3A	2			
0101	5A	0	0101	P2=5A	0010
1010	2x5A	1			
0111	7A	0	0111	P3=7A	0011
1110	2x7A	1			
1001	9A	0	1001	P4=9A	0100
1011	11A	0	1011	P5=11A	0101
1101	13A	0	1101	P6=13A	0110
1111	15A	0	1111	P7=15A	0111



Flow Summary

Flow Status	Successful - Wed Nov 05 12:40:06 2014
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	UIY
Top-level Entity Name	APC_FIR_TOP
Family	Cyclone III
Met timing requirements	N/A
Total logic elements	736 / 5,136 (14 %)
Total combinational functions	720 / 5,136 (14 %)
Dedicated logic registers	289 / 5,136 (6 %)
Total registers	289
Total pins	43 / 183 (23 %)
Total virtual pins	0
Total memory bits	0 / 423,936 (0 %)
Embedded Multiplier 9-bit elements	46 / 46 (100 %)
Total PLLs	0 / 2 (0 %)
Device	EP3C5F256C6
Timing Models	Final

Figure 9: Area analyzer summary

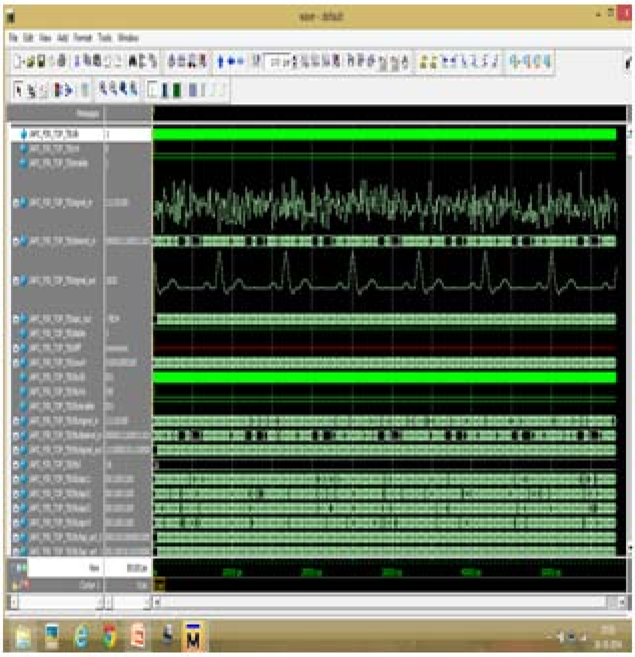


Figure 10: Snapshots of the output waveform

Then the above simulation output is the high accuracy of the input signal.

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

[1] Chang C.H, Jong C.C and Xu.F (2007) ”Design of Low-Complexity FIR Filters Based on Signed-Powers-of-two Coefficients With Reusable Common Subexpression”, IEEE a Transaction on Very Large Scale Integration(VLSI) Syst.,Vol.26,no.10.,pp 1898-1907.

[2] Cowan C.F.N, Ting L.K and Woods.R (2005) “Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers”, IEEE Transaction on Very Large Scale Integration(VLSI) Syst.,Vol.13.no.1.pp.86-99.

[3] Feng W.S and Van L.D (2001) “An efficient systolic architecture for the DLMS adaptive filter and its applications”, IEEE Transaction on Circuits Syst.II,Analog Digital Signal Process.,Vol.48,no.4,pp 359-366.

[4] Haykin.S and Widrow.B(2003) Least –Mean-Square Adaptive Filters., Hoboken, NJ, USA: Wiley.

[5] Ling .F, Long.G and Proakis J.G(1992) “Corrections to ‘The LMS algorithm with delayed coefficient adaptation”,IEEE Transaction on Signal Process., vol.40,no.1, pp. 230–232.

[6] Maheshwari .M and Meher P.K (2011) “A high-speed FIR adaptive filter architecture using a modified delayed LMS algorithm”,IEEE International Symposium on Circuits System., pp 121-124.

[7] Meher P.K and Park S.Y (2011) “Low adaptation-delay LMS adaptive filter part-I: Introducing a novel multiplication cell”, IEEE International Midwest Symposium on Circuits System., pp 1-4.

[8] Pramod Kumar Meher (2010) “Novel Input Coding Technique for High-Precision LUT-Based Multiplication for DSP Applications”, IEEE Transaction on Embedded System, IEEE/IFIP International Conference on Very Large Scale Integration (VLSI) and System-On-Chip (SOC)., pp 201-206.

[9] Pramod Kumar Meher and Sang Yoon Park (2014)”Area-Delay-Power Efficient Fixed-Point LMS Adaptive Filter With Low Adaptation-Delay”, IEEE Transaction on Very Large Scale Integration (VLSI) Syst.,Vol.22,no.2,pp 362-371.

[10] Stearns S.D and Widrow .B (1985) Adaptive Signal Processing ,Englewood Cliffs, NJ, USA: Prentice-Hall.

[11] Tang Bin, Wang Sen and Zhu Jun (2006) ”Distributed Arithmetic for FIR Filter Design on FPGA”, IEEE International Symposium on Signal Processing and Information Technology., pp 248-252.