Memory Optimization in Adaptive Fir Filter using APC-OMS and CSE Method

E. Suruthi
P.G Scholar,
Dept. of Electronics and communication Engineering
Sri Muthukumaran Institute Of Technology
Chennai,Tamilnadu,India

Prof. D. Sowmiya
Associate Professor,
Dept. 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. Besides, we have described the design and implementation of Adaptive FIR filter using APC-OMS and CSE method. The APC-OMS involves reduction in LUT size to third-fourth of the conventional LUT and the error is eliminated at the output. The CSE method is to reduce the number of shifting operation in the multipliers. The area, power is reduced and the speed is increased. The overall design implementation on the ALTRA cyclone III FPGA kit.**

Index Terms: Adaptive FIR Filter, APC-OMS, CSE.

1. INTRODUCTION

ADAPTIVE FIR filters have a wide range of communication and DSP applications such as adaptive equalization,system identification and image restoration. The direct-form LMS 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 behavior, it is modified to a form called the delayed LMS (DLMS) algorithm 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 but they involve an adaptation delay. A systolic architecture, where they have used relatively large processing elements (PEs) 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. In adaptive fir filter coefficients are not fixed. The existing work on the fixed point LMS adaptive filter 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, time, delay and power. We have referred to this as odd-multiple-storage (OMS) scheme and anti-symmetric product coding. 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) multiplier but the multiplications are largely involved. Reduction in the number of multiplication is possible by the CSE method with the shift and add algorithm. The area and power is reduced and the speed is increased. We discuss the synthesis of the proposed architecture and comparison with the existing architectures.

II. REVIEW OF FIXED POINT IMPLEMENTATION

In this Existing adder we will use the implementation of a delayed least mean square(DLMS) adaptive filter. For achieving lower adaptation-delay and area-delay-power efficient implementation, use a novel Partial Product Generator(PPG). 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).

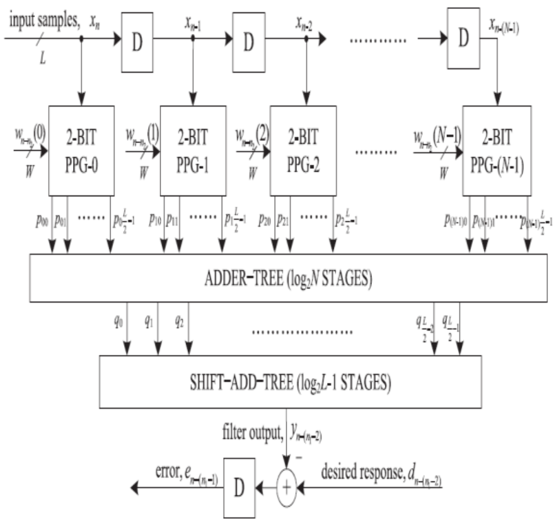


Fig.1. Existing system error computation block

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 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. The design with a clock slower than the maximum usable frequency and a lower operating voltage to reduce the power consumption.

III. PROPOSED ARCHITECTURE

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

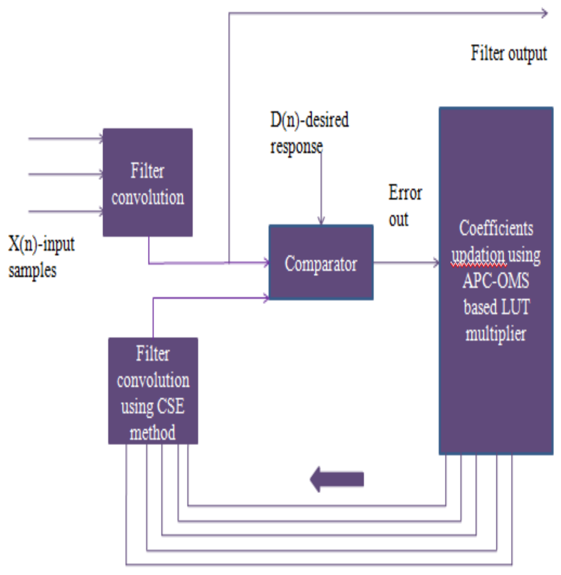

Filter output
D(n)-desired response
Filter convolution
Error out
X(n)-input samples
Comparator
Coefficients updation using APC-OMS based LUT multiplier
Filter convolution using CSE method

Fig.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).

A. 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.

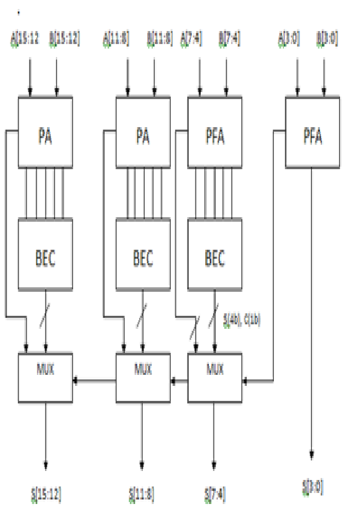


Fig.3. Adder block diagram

B.Shift/Add Tree

The shift/add tree is placed on the filter convolution using multiplier. 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
```

IV. THE APC-OMS AND CSE METHOD

APC-OMS METHOD

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. 1, where *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, there can be $2L$ possible values of X, and accordingly, there can be $2L$ possible values of product $C = A \cdot X$.

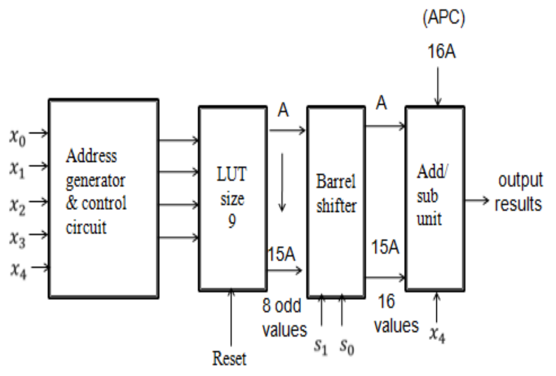


Fig.4.APC-OMS block diagram

Therefore, for memory-based multiplication, an LUT of $2L$ words, 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 both$X$ and A to be positive integers.The product words for different values of X for $L = 5$ are shown in Table I. It may be observed in this table that the input word X on the first column of each row is the two's complement of that on the third column of the same row.In addition, the sum of product values corresponding to these two input values on the same row is $32A$. Let the product values on the second and fourth columns of a row be u and v, respectively. 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 LUT size, where, 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

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

The 4-bit LUT addresses and corresponding coded words are listed on the fifth and sixth columns of the table, respectively. Since the representation of the product is derived from the antisymmetric behavior of the products, we can name it as *antisymmetric product code*. The 4-bit address $X\_ = (x\_3x\_2x\_1x\_0)$ 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 $XL = (x3x2x1x0)$ is the four less significant bits of X, and X_L is the two's complement of XL.The desired product could be obtained by adding or subtracting the stored value $(v - u)$ to or from the fixed value $16A$ when $x4$ is 1 or 0, respectively.

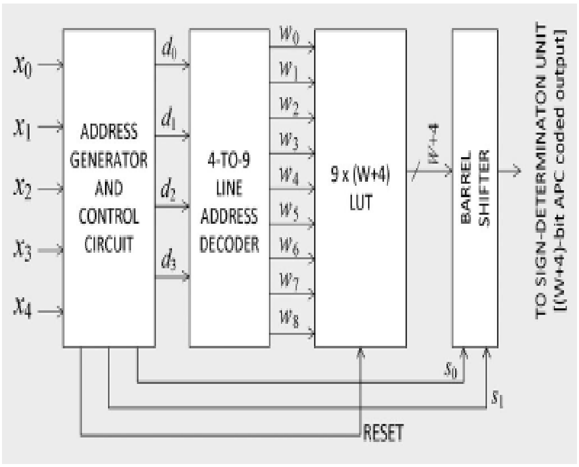


Fig.5.Proposed APC-OMS combined LUT

Product word = $16A$ + (sign value) × (APC word). where sign value = 1 for $x4 = 1$ and sign value = -1 for $x4 = 0$. The product value for $X = (10000)$ corresponds to APC value "zero," which could be derived by resetting the LUT output, instead of storing that in the LUT**.**

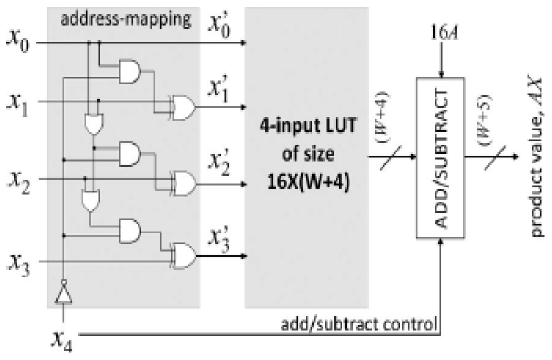


Fig.6. LUT-based multiplier for $L = 5$ using the APC technique

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

TABLE I APC WORDS FOR DIFFERENT INPUT VALUES

B.Modified OMS for LUT optimization

The multiplication of any binary word *X* of size *L*, with a fixed coefficient *A*, instead of storing all the 2*L* possible values of *C* = *A* ・ *X*, only (2*L*/2) words corresponding to the odd multiples of *A* may be stored in the LUT, while all the even multiples of *A* could be derived by left-shift operations of one of those odd multiples.

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 even multiples 2*A*, 4*A*, and 8*A* are derived by left-shift operations of *A*. Similarly, 6*A* and 12*A* are derived by left shifting 3*A*, while 10*A* and 14*A* are derived by left shifting 5*A* and 7*A*, respectively. A barrel shifter for producing a maximum of three left shifts could be used to derive all the even multiples of *A*.

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

TABLE II OMS-BASED DESIGN OF THE LUT OF APC WORDS

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.

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	2xA	1			
0100	4xA	2			
1000	8xA	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

TABLE III REDUCED APC-OMS ADDRESS

CSE METHOD

In adaptive FIR filter, the memory to be reduced based on the architecture level using Common Sub-expression Elimination (CSE) method. When a portion of an expression (sub-expression) occurs more than once ,it can be calculated once and the result can be used further. The Common Sub-expression Elimination method is used for reducing the number of shifting and adding operations and increasing the speed. The use of multiplier is reduced based on the shift and add method. The shift and add algorithm used to reduce the number of multiplications in the APC-OMS output. So the area and power is reduced when compared to the existing method.

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.

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

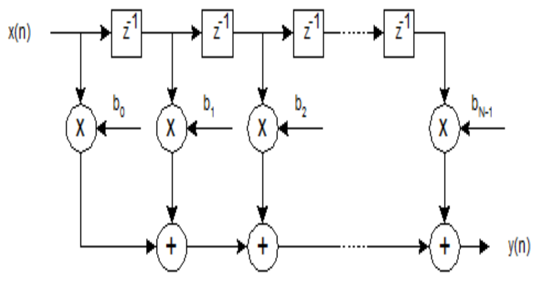
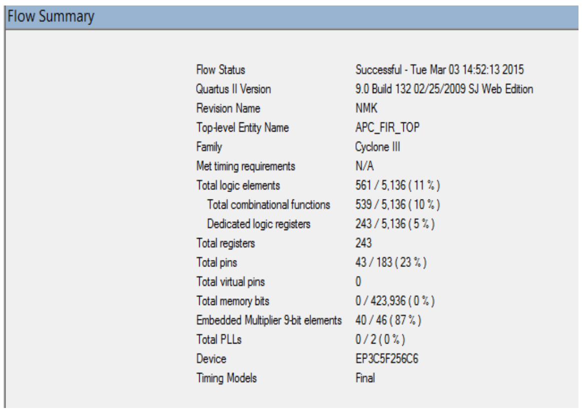


Fig.7.Coefficient Calculation

Example:
A-0000 0000
B-0000 1001
Q-1100

Step	A	Q	B	Operation
0	0000 0000	1100	0000 1001	Initialization
1	0000 0000	1100	0001 0010	Shift left B
	0000 0000	0110	0001 0010	Shift right Q
2	0000 0000	0110	0010 0100	Shift left B
	0000 0000	0011	0010 0100	Shift right Q
3	0010 0100	0011	0010 0100	Add B to A
	0010 0100	0011	0100 1000	Shift left B
	0010 0100	0001	0100 1000	Shift right Q
4	0110 1100	0001	0100 1000	Add B to A
	0110 1100	0001	1001 0000	Shift left B
	0110 1100	0000	1001 0000	Shift right Q

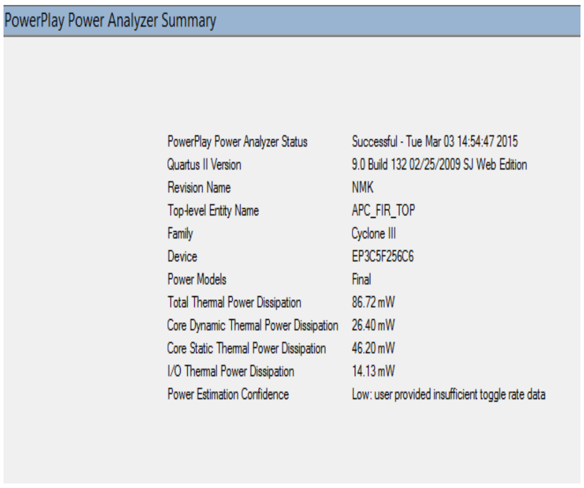
The binary output value is same as the shift and add output and the number of shifting and adding is reduced in the shift and add method. So the area and power is reduced.



Flow Summary

Flow Status	Successful - Tue Mar 03 14:52:13 2015
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	NMK
Top-level Entity Name	APC_FIR_TOP
Family	Cyclone III
Met timing requirements	N/A
Total logic elements	561 / 5,136 (11 %)
Total combinational functions	539 / 5,136 (10 %)
Dedicated logic registers	243 / 5,136 (5 %)
Total registers	243
Total pins	43 / 183 (23 %)
Total virtual pins	0
Total memory bits	0 / 423,936 (0 %)
Embedded Multiplier 9-bit elements	40 / 46 (87 %)
Total PLLs	0 / 2 (0 %)
Device	EP3C5F256C6
Timing Models	Final

Fig.8.Area Report



PowerPlay Power Analyzer Summary

PowerPlay Power Analyzer Status	Successful - Tue Mar 03 14:54:47 2015
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	NMK
Top-level Entity Name	APC_FIR_TOP
Family	Cyclone III
Device	EP3C5F256C6
Power Models	Final
Total Thermal Power Dissipation	86.72 mW
Core Dynamic Thermal Power Dissipation	26.40 mW
Core Static Thermal Power Dissipation	46.20 mW
I/O Thermal Power Dissipation	14.13 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Fig.9.Power Report

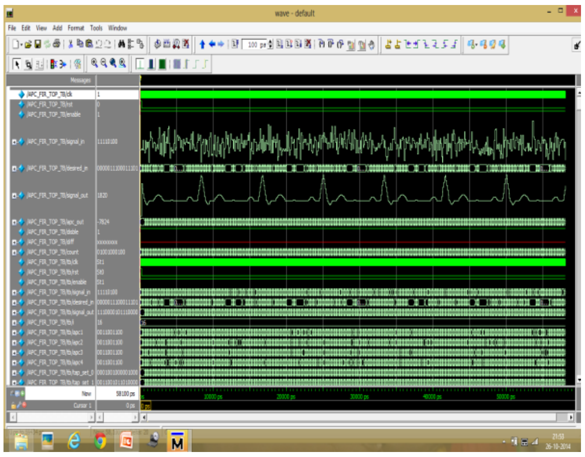


Fig.10.Snapshots of the output waveform

V. SIMULATION RESULTS

Simulations have been performed using Modelsim 6.4a Simulation tool technology Fig.10 shows the input and output waveform results for proposed APC-OMS technique. Fig.9. shows the power analyzer summary.
Fig.8. shows the area report. 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. The same process applied to the CSE method. The area is reduced. From the results, it is clear that the proposed circuit can have very less power.

VI.CONCLUSION AND FUTURE WORK

Analyzing the APC-OMS and CSE FIR architecture and performance, this paper presents a new architecture for DALUT. The proposed architecture applies the main concept of the basic APC-OMS and CSE method implementing the MAC unit and at the same time has many advantages over its basic architecture.The results obtained show that with the proposed architecture, the computation time and the area used is reduced. The overall design implemetations on the ALTRA cyclone III FPGA kit.The error out output is displayed on the system.

VII.REFERENCES

[1] Agrawal D.P and Meyer M.D (1993) “A high sampling rate delayed LMS filter architecture”, IEEE Journal of Circuit Syst, Vol. 40, no 11, pp. 727-729.
[2] 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 Journal of Very Large Scale Integration(VLSI) Syst, Vol. 26, no.10, pp. 1898-1907.
[3] 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 Journal of Very Large Scale Integration(VLSI) Syst, Vol. 13, no.1, pp. 86-99.
[4] Feng W.S and Van L.D (2001) “An efficient systolic architecture for the DLMS adaptive filter and its applications”, IEEE Journal of Analog Digital Signal Process, Vol. 48, no.4, pp. 359-366.
[5] Ho.H and Szwarc.V (2007) “Hardware optimization for a reconfigurable polyphase-FFT design using common sub-expression elimination”, IEEE Conf., on Circuits and Systems,pp. 650-653.
[6] Ling F, Long G and Proakis J.G (1992) “Corrections to the LMS algorithm with delayed coefficient adaptation”, IEEE Journal on Signal Process, Vol. 40, no.1, pp. 230–232.
[7] Menard D, Rocher R, Scalart P and Sentieys O (2004) “Accuracy evaluation of fixed-point LMS algorithm”, Proc. Inter. Conf. Acoust., on Speech, Signal Process, pp. 34-41.
[8] Pramod Kumar Meher (2010) “Novel Input Coding Technique for High-Precision LUT-Based Multiplication for DSP Applications”, Proc. Inter. Conf., on Very Large Scale Integration (VLSI) and System-On-Chip (SOC), pp. 201-206.
[9] Pramod Kumar Meher and Yu Pan (2014) “Bit-Level Optimization of Adder-Trees for Multiple Constant Multiplications for Efficient FIR Filter Implementation”,IEEE Journal of Circuits and Syst,Vol. 61,no.2 ,pp. 455-462.
[10] Qiu-zhong Wu and Yi-he Sun (2005) “An integrated CAD tool for ASIC implementation of multiplierless FIR filters with common sub-expression eliminatiom potimization”,IEEE Conf., on ,pp. 67-72.