FPGA Based Adujsted Step Size LMS Algorithm for Adaptive Noise Cancellation

Associate Prof. Thamer Muhammed Jamel
Associate Lecturer Haider Abd Al-Latif Mohamed
University of Technology / Department of Communications Engineering
Baghdad / Iraq
Email: 30024@uotechnology.edu.iq

Abstract:

In this paper an FPGA based Adaptive Noise Canceller (ANC) with two inputs was designed and built using our previously proposed algorithm in 2014, which was called Absolute Average Error Adjusted Step Size LMS (AAE-ASSLMS). This algorithm is synthesized and simulated on Xilinx Spartan-3E platform. Then a connection between MATLAB and FPGA was made in order to solve the problem of entering the speech signal to FPGA for testing the FPGA performance with perfect appearance. A comparison between hardware and MATLAB software implementation is then made and the results were approximately equal. The suggested AAE-ASSLMS adaptive filter that implemented on FPGA for ANC application works well. The final hardware design can be used for the embedded applications as System on Chip (SoC) for ANC in many portable applications like mobile phone and hearing aids.

Keywords: LMS Algorithm; Adaptive noise canceller; Adjusted Step Size; FPGA.

1. Introduction:

Hardware design of adaptive filter for FPGA implementation became very important because different applications require different needs of area and speed. As an example the LMS adaptive filter used in mobile phone need areas to be as low as possible to gain lower weight of the mobile device, but the speed in this application, has less importance because it used for speech processing [9].

Field's programmable gate arrays (FPGAs) are commonly used in digital signal processing applications such as Adaptive Noise Cancellation (ANC) [1,3-6,8-11]. In this paper we tried to implement ANC on an FPGA using Xilinx Spartan-3E platform . The ANC will use our previously proposed algorithm called the AAE-ASSLMS algorithm as the adaptive filtering algorithm [7-8]. Then try to make a

connection between MATLAB and FPGA in order to solve the problem of entering the speech signal to FPGA for testing the FPGA performance with perfect appearance. This connection will overcome the problem of entering the overall samples by hands which is a hard task or impossible.

Adaptive Noise Cancellation is a an adaptive system for removing an unwanted noise from a desired signal. As shown in **Fig.1**, the ANC has two inputs called primary and reference inputs.

The reference input is filtered and subtracted from a primary input, which is contained both signal and noise. The primary noise will be reduced by cancellation as shown later. As shown in **Fig.1**, a noise signal n_1 from reference input is passed over a noise path channel (H (z)) to a primary input that receives both signal (s) and an uncorrelated noise n_0. The noise n_1 is filtered to produce an output signal y that is a close replica of n_0 and is subtracted from the primary input (s+n_0) to produce the system output signal called the error signal e which is equal to [1] :-

$$e = s + n_0 - y \qquad (1)$$

The error signal is used to adjust the adaptive filter's weights coefficients using an adaptive algorithm such that the error signal is minimized.

2.(AAE-ASSLMS) Algorithm

As explained previously this paper uses previously proposed and published algorithm called Absolute Average Error Adjusted Step Size LMS (AAE-ASSLMS) [9 and 10] as an adaptive algorithm for adaptive noise canceller. AAE-ASSLMS algorithm used adjusted step size that will be time varying according to the absolute average value of the current and the previous estimator errors as follows:-

$$\mu_{n+1} = \mu_n - abs\left(\left(\sum_{c=0}^{L} e(n-c)\right) * \beta\right) \qquad (5)$$

Where L is the filter length, μ_{n+1} and μ_n are next and current time varying step size respectively, and β is constant with bound interval limits $0<\beta<1$. To ensure stability, the variable step size μ_n is constrained to the pre-determined maximum μ_{MAX} and minimum step size values μ_{min} :-

$$\left.\begin{array}{l} \mu_{n+1} = \mu_{MAX} \ \ if \mu_{n+1} > \mu_{Max} \\ \text{or} \\ \mu_{n+1} = \mu_{min} \ \ if \mu_{n+1} < \mu_{min} \end{array}\right\} \qquad (6)$$

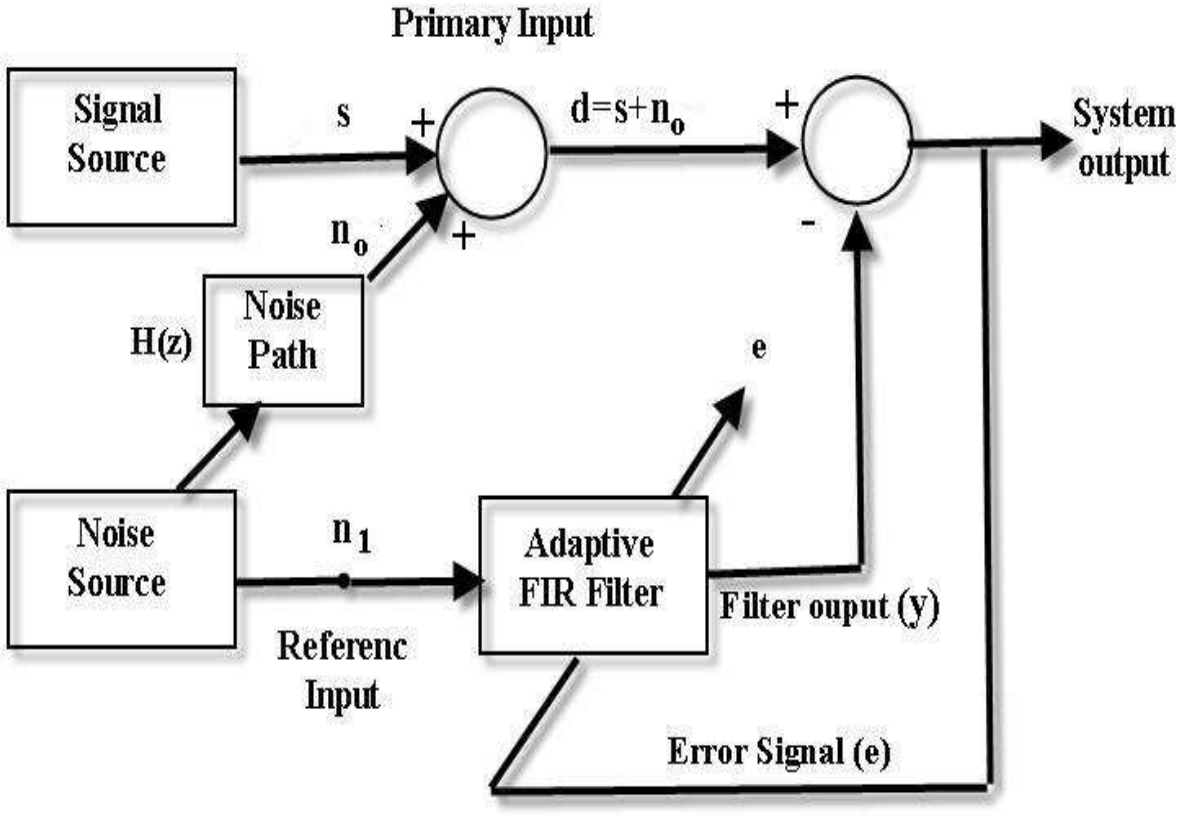

Fig. 1 ANC Model

The filter weights of the adaptive filter **Fig.2** are updated in each iteration according to the following formula [7-8].

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + 2\mu(n)e(n)\boldsymbol{n_1}(n) \quad (7)$$

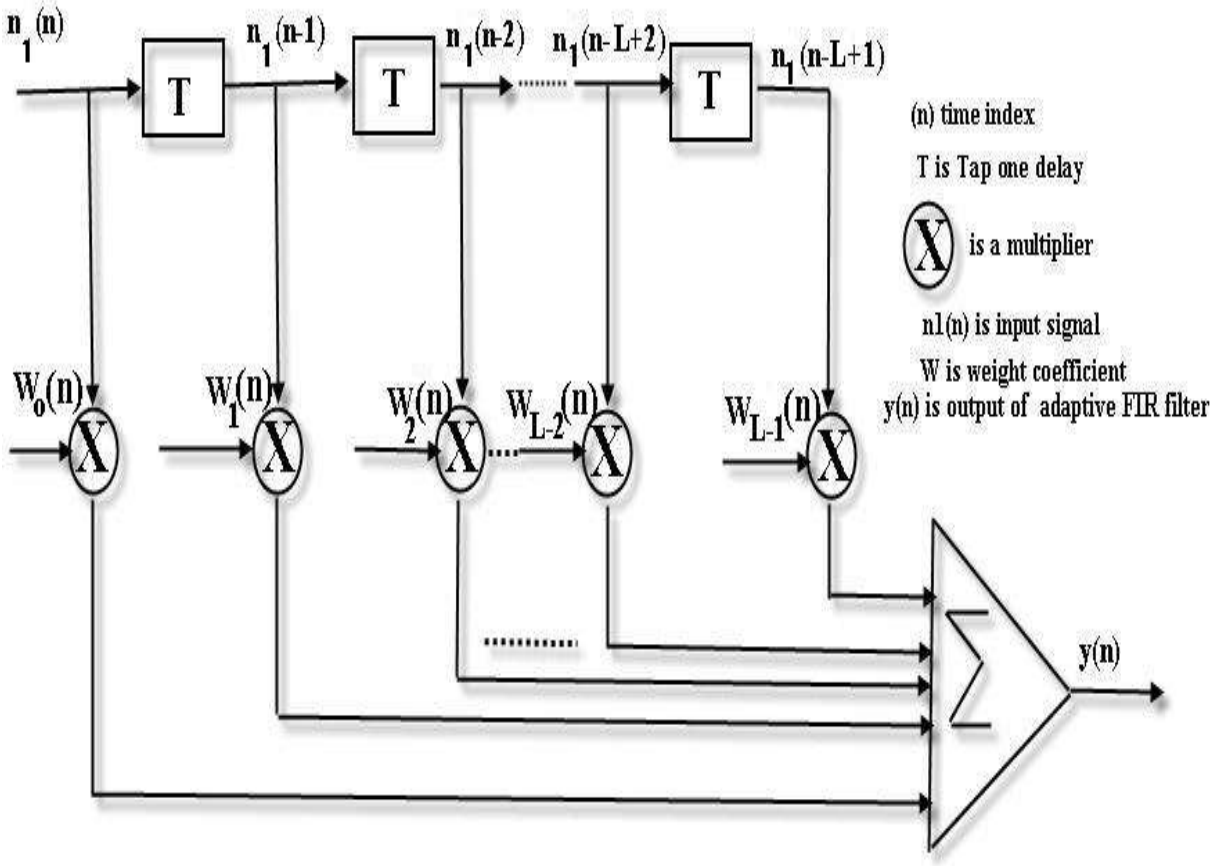

Fig. 2 Structure of Adaptive filter

3. Structure of AAE-ASSLMS Algorithm

The external input/output of the proposed algorithm is shown in **Fig. 3**. As shown in this Figure the external I/Os are: desired signal $d(n)$, input signal $x(n)$, maximum step size (μ_{max}) , minimum step size (μ_{min}) , constant (β) and the error signal $e(n)$.

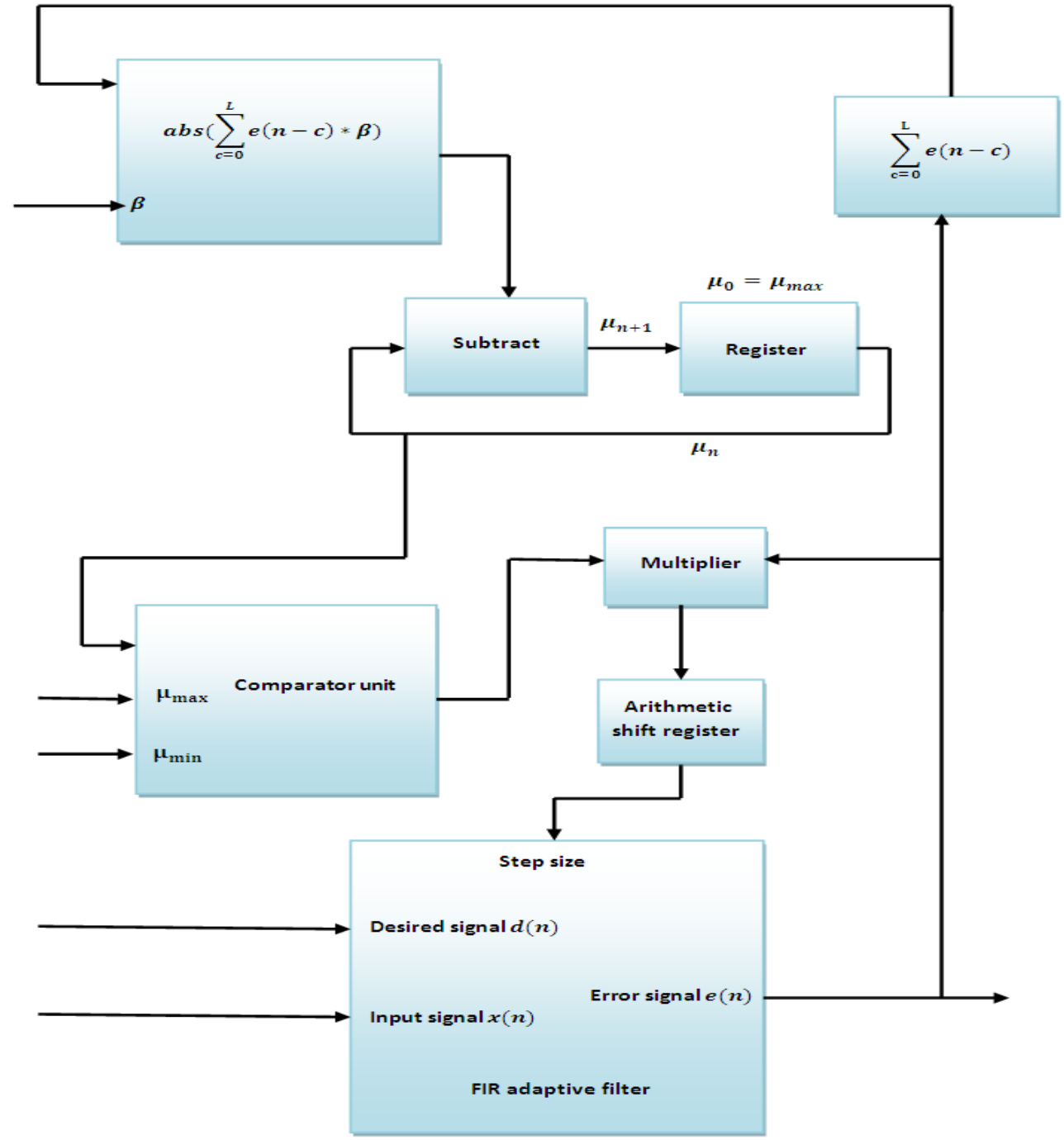

Fig. 3 Structure of the AAE-ASSLMS algorithm

This Figure contains on the comparator, summation of the current and previous estimation errors for a whole filter length (in this paper L=8) plus one, absolute value, multiplier to get the final step size, arithmetic shift

register (in order to get $2\mu_n$) and the FIR adaptive filter block. The internal structure of the FIR adaptive filter block can be explained clearly in **Fig. 4** to show the internal multipliers and adders. Also to show the registers that are used in order to store the previous weight coefficients. Each tap in FIR adaptive filter contains on two multipliers, one adder and two registers except the last tap contains on one register only.

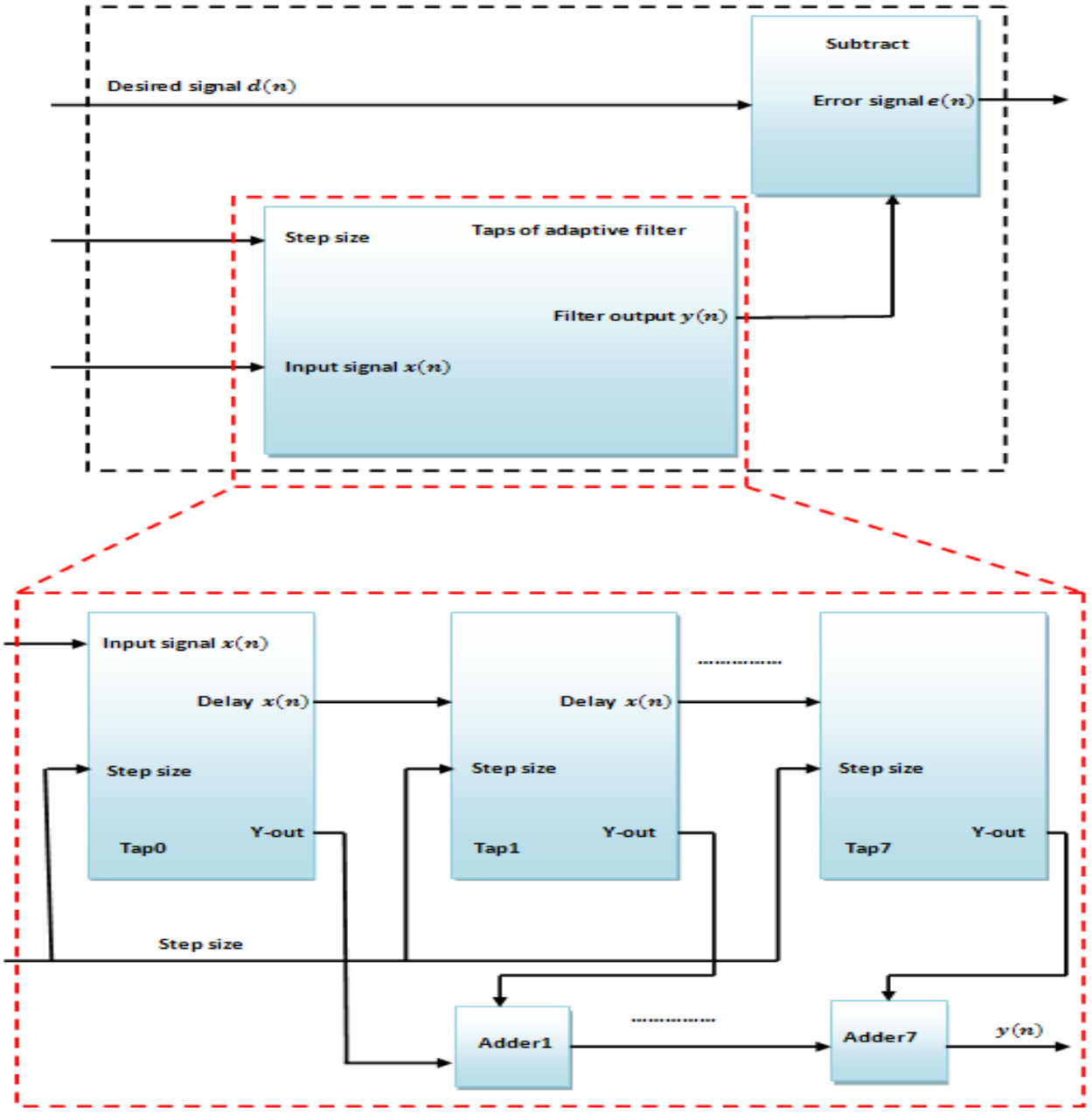

Fig. 4 Structure of FIR adaptive filter

4. FPGA Implementation of AAE-ASSLMS Algorithm

In this paper, the ANC system is designed and hardware implemented in the Xilinx Spartan-3E Starter Kit board .

A. Processing Number

The data format might be positive or negative. Therefore , the highest bit is used as a sign bit.

B. Fixed-point Data Format for AAE-ASSLMS Adaptive Filter

The maximum ranges of the input signals, output signal and tap weights obtained by MATLAB-simulation are shown in **Table.1** For the input signal $x(n)$ range from $(-4\,,4)$; thus, its whole scale may use 2-bit, but in order to guarantee that no overflow will be happening and to give the design some flexibility, this paper assigned it 3-bit. Then the decimal fraction is located between the 13^{th} bit and 14^{th} bit and assign the sign bit location 17^{th} bit.

Table. 1 Numerical Ranges of Signals and Registers Using Matlab Simulation

Signal or Register name	Input signal $x(n)$	Desired signal $d(n)$	Filtered signal $e(n)$	Weight $w(n)$
Data range	$(-4\,,4)$	$(-4\,,4)$	$(-1\,,1)$	$(-0.15,1)$

C. The Design of the Complete Circuit

The complete design of the circuit of the AAE-ASSLMS adaptive filter which contains the eight tap unit, comparator unit, absolute unit and the input/output of the circuit is as shown in **Fig. 5**. The internal structure of FIR adaptive filter (eight taps unit) is shown in the **Fig.6**. Moreover , **Fig. 7** shows the internal structure of one tap. The detail's description of the units of **Fig. 5** is as follows:-

- Delay unit: this unit contains on the register to get the previous estimation errors for a whole of filter length.

- Average unit: this unit contains on adders to get the summation of the current and the previous estimation errors for a whole filter length plus one.

- Absolute unit: this unit contains on the logic relational, negate (i.e. two's complement) and two multiplexer in order to get the absolute value of $\left((\sum_{c=0}^{L} e(n-c)) * \beta\right)$.
- Accumulator unit: this unit contains on the register and subtraction in order to get the $\mu_{n+1} = \mu_n - abs\left((\sum_{c=0}^{L} e(n-c)) * \beta\right)$.
- Comparator unit: this unit contains on two relational and two multiplexer in order to get the boundary condition of the step size (i.e. Between the μ_{max} and μ_{min}).
- Arithmetic shift register: this unit is used to get the value of the term $(2\mu_n)$.

. Moreover, the same method was used for other signals. The selected data formats are shown in **Table.2** and **Table.3.**

Table. 2 Data Format of $x(n), d(n), w(n)$ and $e(n)$

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sign	integer			Decimal fraction													

Table. 3 Data Format of $y(n)$

35	34	33	32	31	30	29	28	...	19	18	17	16	...	3	2	1	0
sign	integer			Decimal fraction							Quantization (Rounding)						

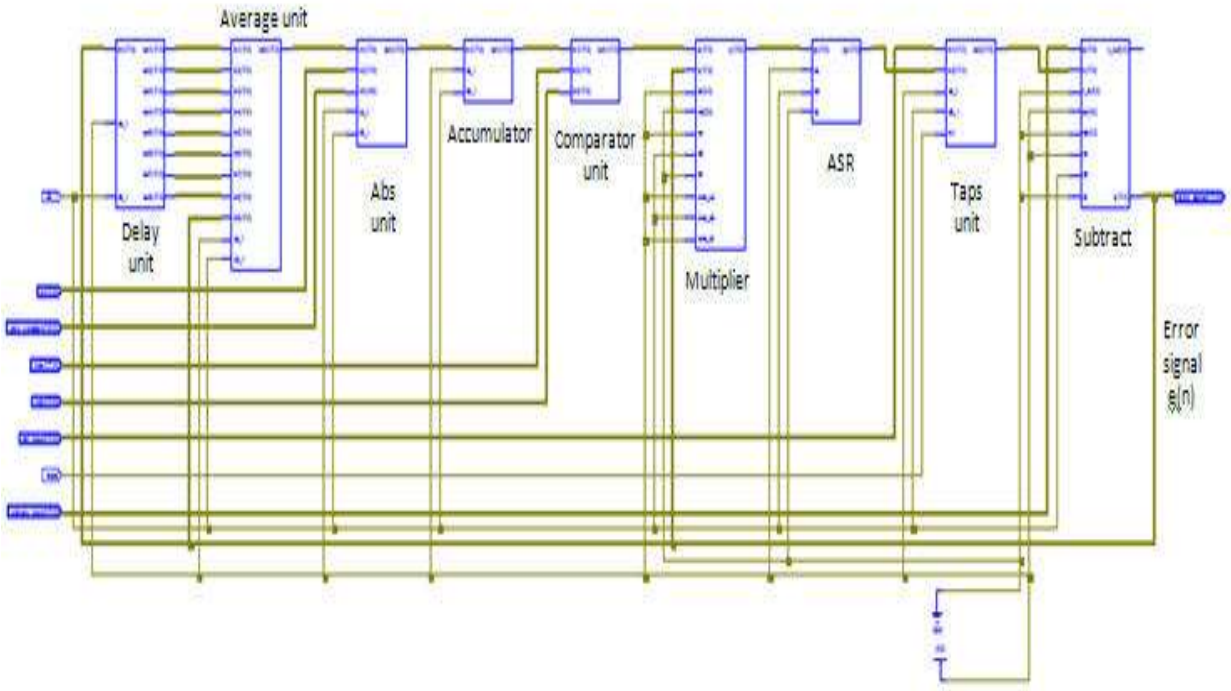

Fig. 5 The complete AAE-ASSLMS adaptive filter circuit.

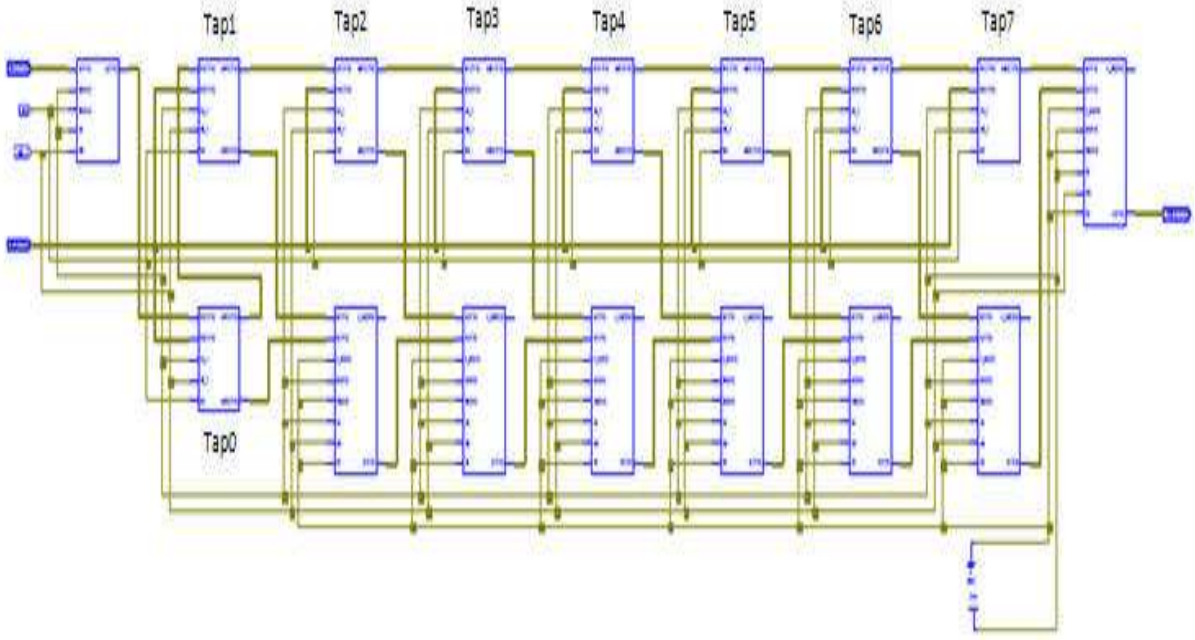

Fig. 6 The internal structure of FIR adaptive filter (eight tap unit).

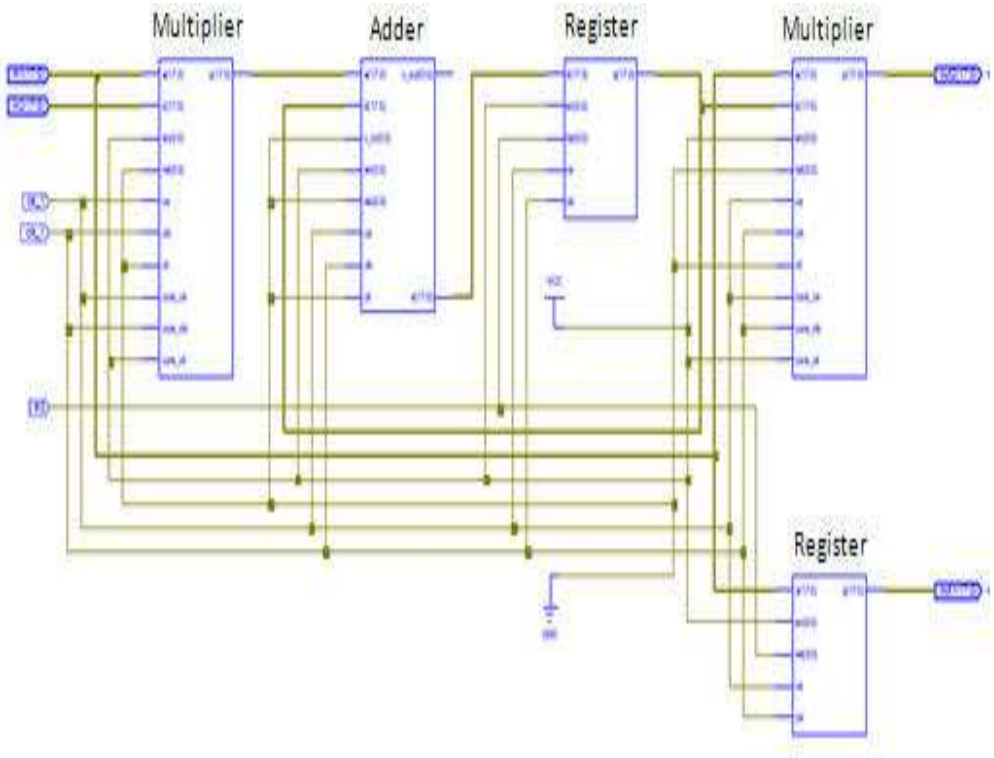

Fig. 7 Internal structure of one tap.

5. Simulation results

In this paper, the number of samples for primary and reference inputs of ANC is equal to (28000) samples. **Fig.8** shows the original speech signal and **Fig.9** shows the primary input signal. The process of entering these samples to Xilinx ISE 10 1i.

Simulator by hand is a very difficult task. In order to overcome this problem a new proposed method is used. This proposed method makes linking between MATLAB and Xilinx ISE as shown in **Fig. 10** In addition, **Fig. 11** shows the MATLAB-Simulink blocks of AAE-ASSLMS.

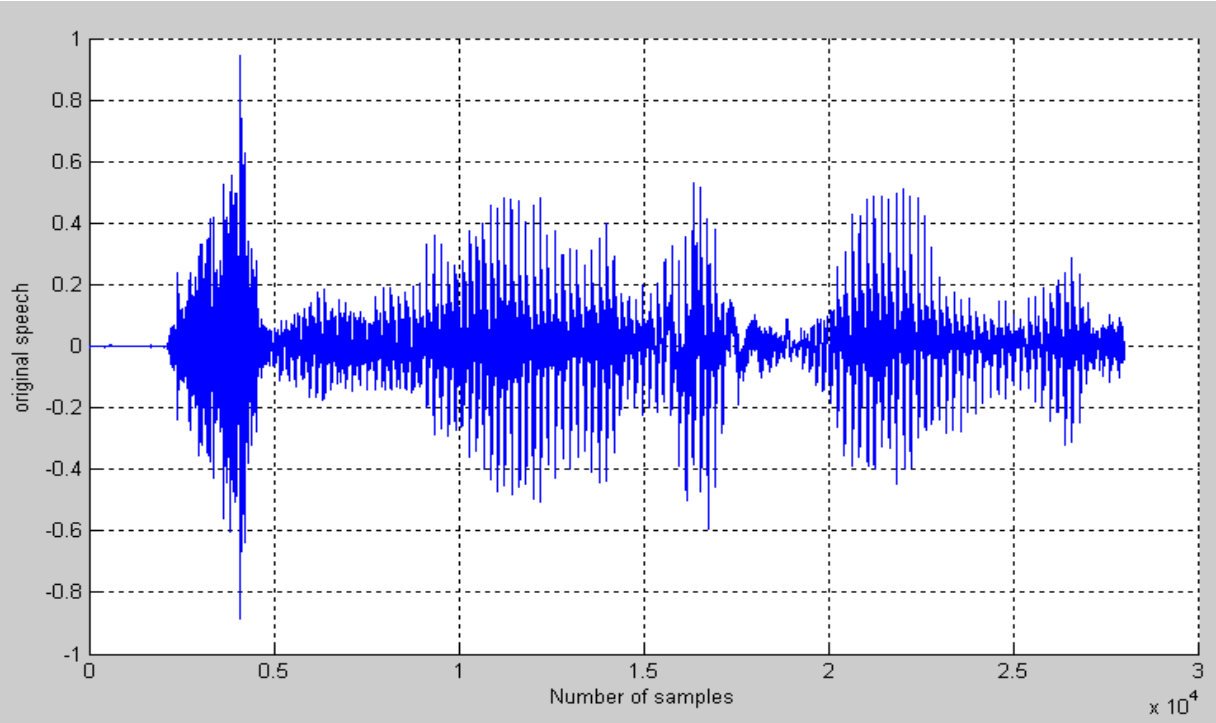

Fig. 8 Original speech Signal

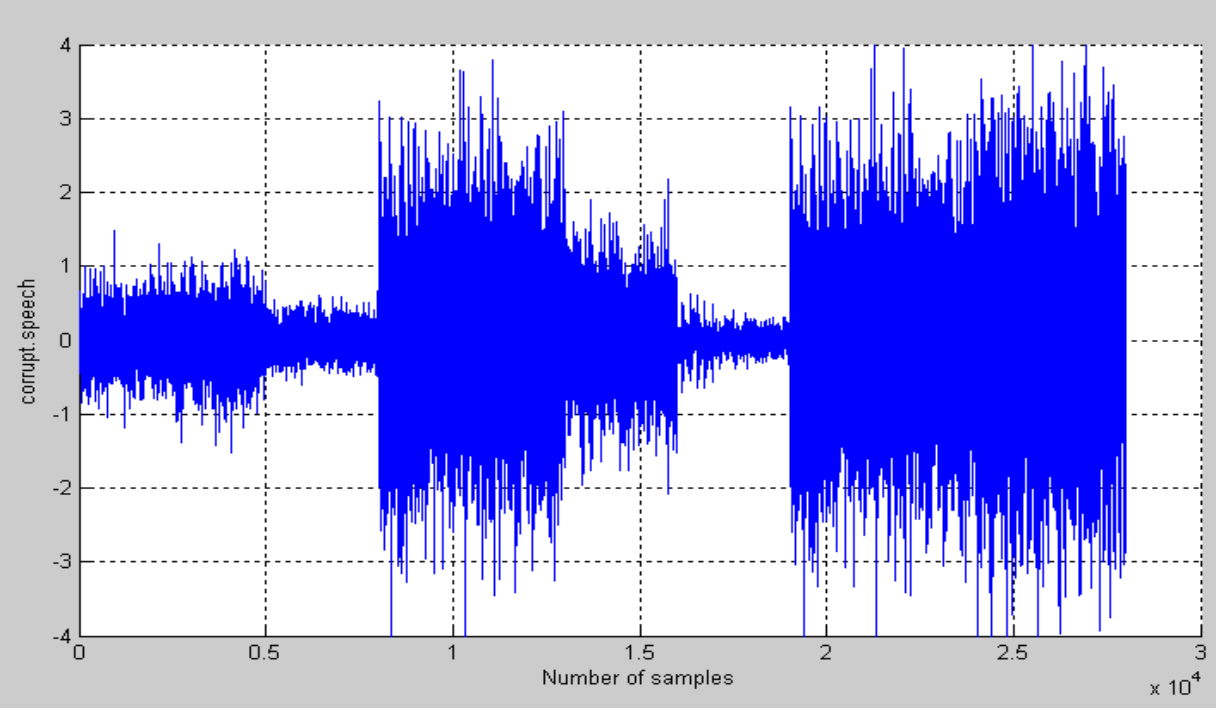

Fig. 9 Corrupted speech signal from time varying noise

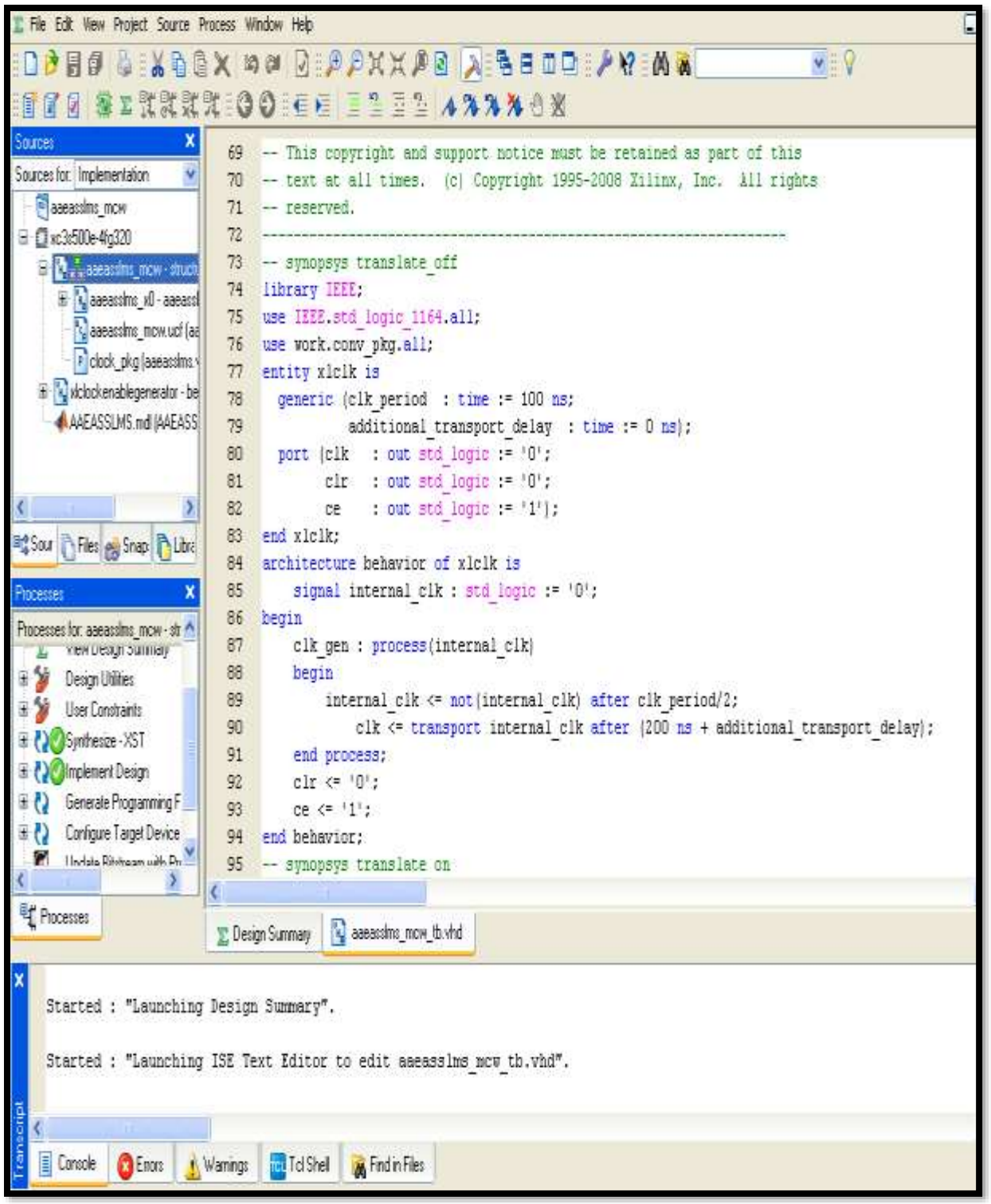

Fig. 10 A connection between MATLAB and Xilinx ISE 10.1i

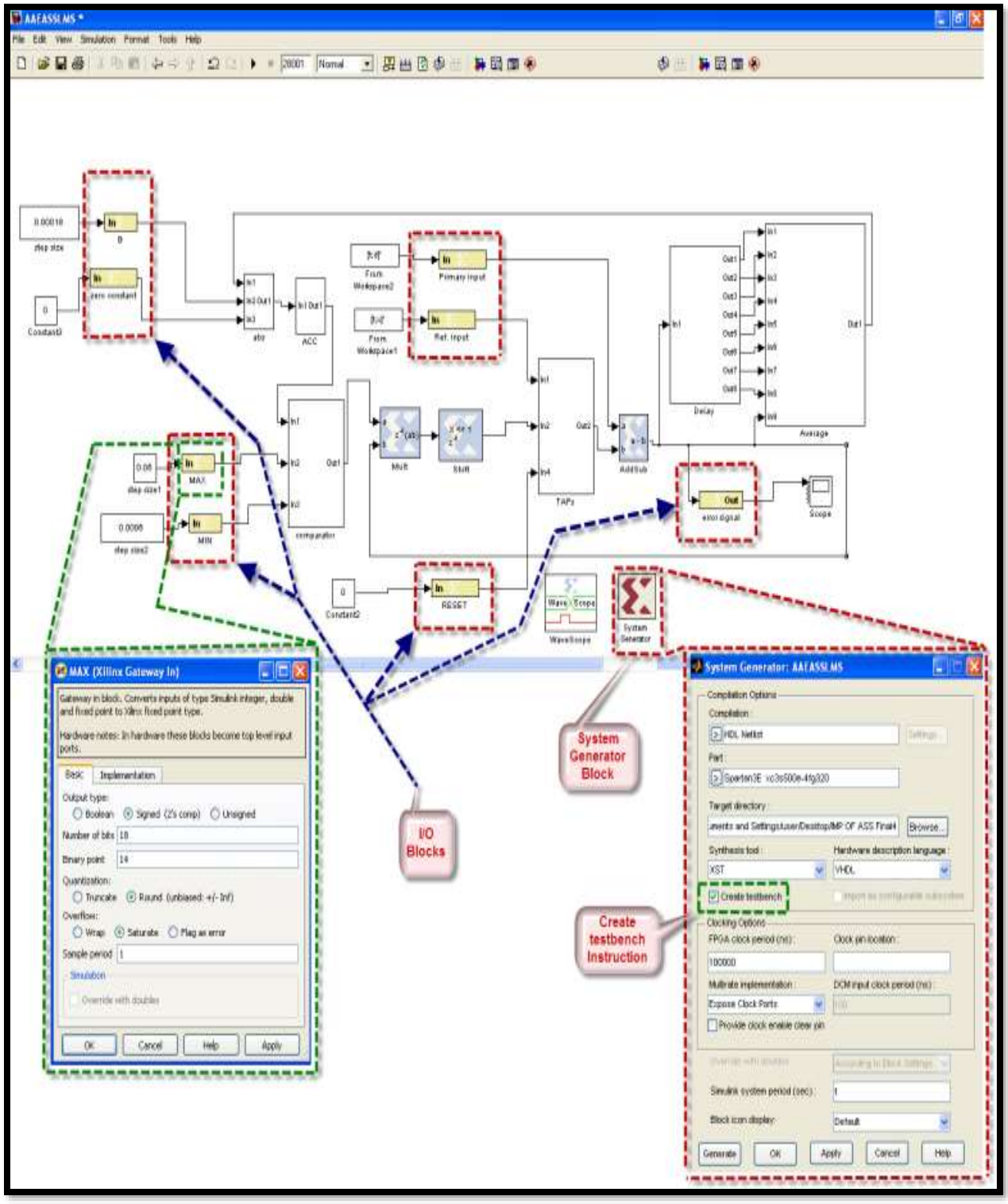

Fig. 11 MATLAB-Simulink blocks of AAE-ASSLMS.

After the simulation of Xilinx ISE (real time implementation) is made on the input data , the output of Xilinx ISE is taken and returns back to MATLAB in order to find the difference between the MATLAB simulation and real time implementation (i.e. Quantization error) . The sampling rate of speech signal in this paper equal to $10KHz$ in order to prevent the antializing. **Fig. 12**, shows the timing diagram of the complete circuit from (0-640) µs. while **Fig. 13** shows an overall timing diagram of the complete circuit design.

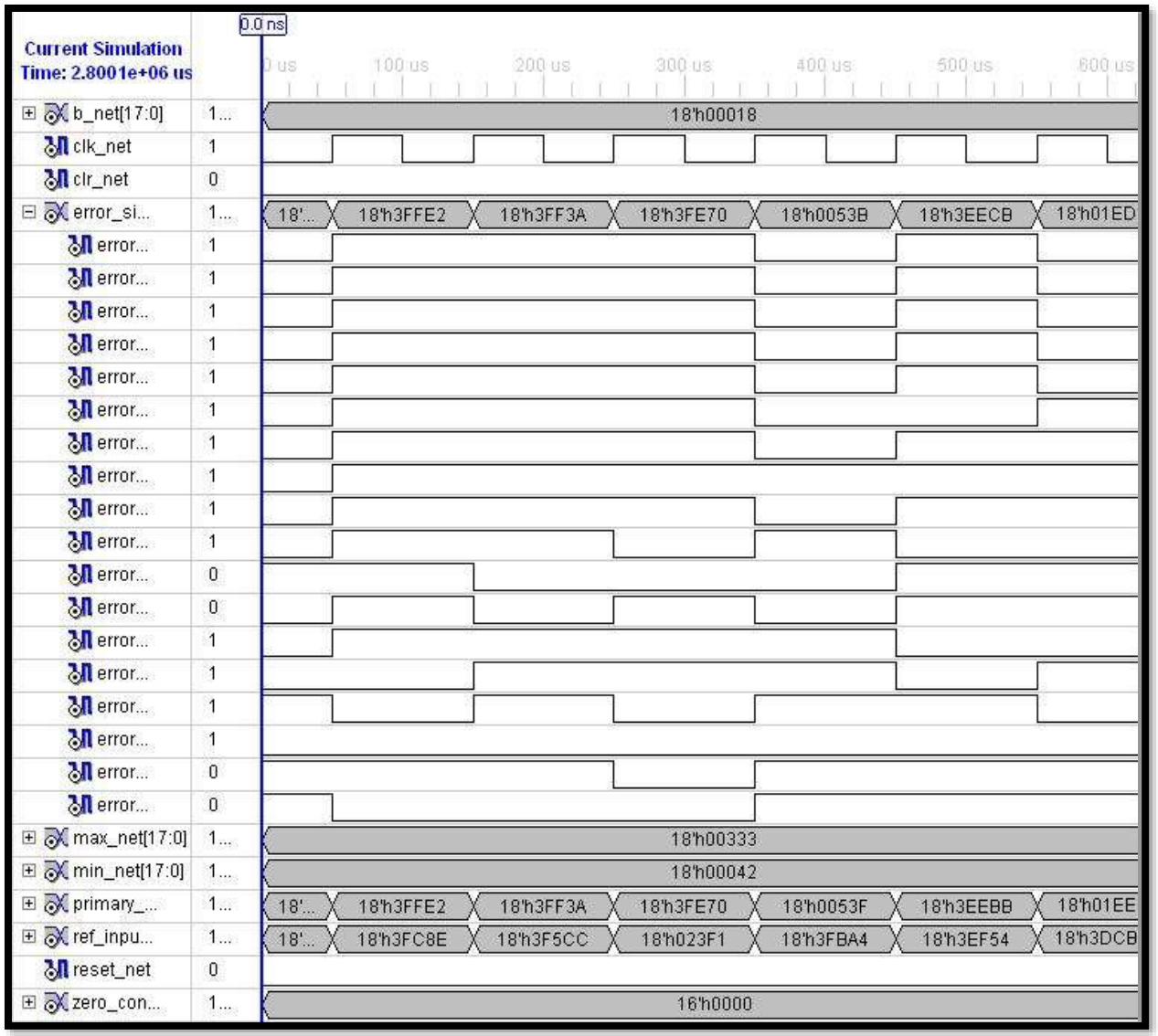

Fig. 12 Timing diagram of the complete circuit design.

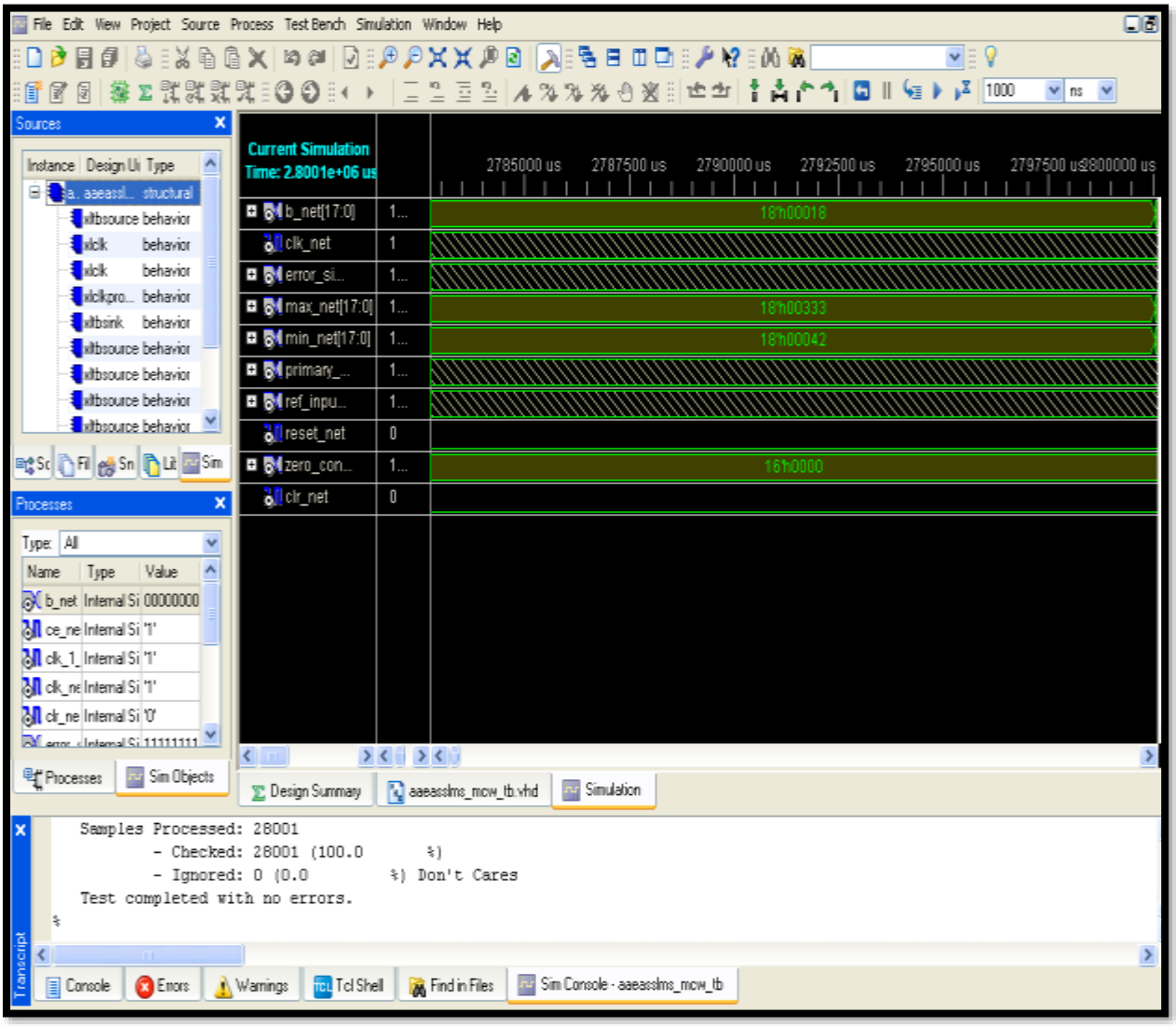

Fig. 13 Overall timing diagram of the complete circuit design

Fig. 14, shows the clean speech signal e(n) that is calculated by MATLAB and Xilinx ISE simulation . It is clear that from this figure the difference between them (quantization error) is not too much. After implementing the AAE-ASSLMS FIR adaptive filter hardware design in the Spartan 3E (XC3S500E-4FG320) device, FPGA; the results for speed and resources used are obtained. **Table .4** gives in details, the resource and the maximum operating frequency requirement of the FPGA implementation . From **Table .4,** the total used number of 4-input LUTs is equal to (10%). The number of 4-input LUTs used as logic equals (7%) and used as a route-thru

equals (3%). The total used number of 4-input LUTs may achieve to the maximum value if the multipliers are built using LUTs. Furthermore, the AAE-ASSLMS adaptive noise cancellation design takes about (90%) of the embedded multipliers and (13%) of the occupied Slices; therefore, the overall delay system is overcome and the system can be operated in frequency larger than the 10KH.

From time summary utilization, the minimum period of the clock system is equal to 59.138 ns (maximum operating frequency 16.910 MHz). Furthermore, the value of maximum combinational path delay is equal to (8.634 ns). The value of combinational path delay will become larger than this value if the multipliers are built using lookup-table instead of the embedded multipliers.

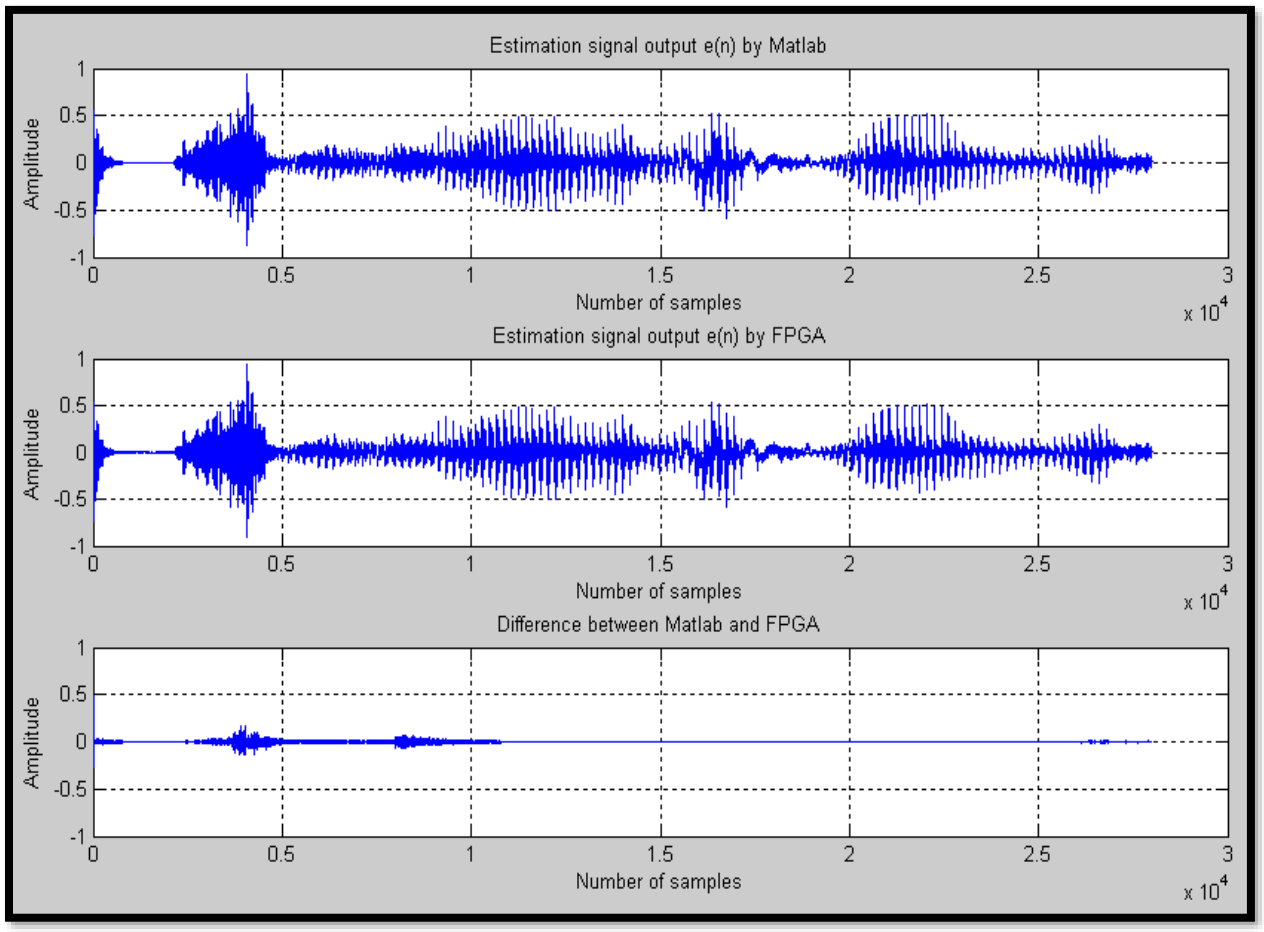

Fig. 14 MATLAB and FPGA Results

Table. 4 The Designed AAE_ASSLMS Implementation r In XC3S500E FPGA

aaeasslms_mcw Project Status			
Project File:	aaeasslms_mcw.ise	**Current State:**	Placed and Routed
Module Name:	aaeasslms_mcw	**• Errors:**	No Errors
Target Device:	xc3s500e-4fg320	**• Warnings:**	259 Warnings
Product Version:	ISE 10.1 - WebPACK	**• Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	**• Timing Constraints:**	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	450	9,312	4%	
Number of 4 input LUTs	674	9,312	7%	
Logic Distribution				
Number of occupied Slices	607	4,656	13%	
Number of Slices containing only related logic	607	607	100%	
Number of Slices containing unrelated logic	0	607	0%	
Total Number of 4 input LUTs	997	9,312	10%	
Number used as logic	674			
Number used as a route-thru	323			
Number of bonded IOBs	123	232	53%	
Number of BUFGMUXs	1	24	4%	
Number of MULT18X18SIOs	18	20	90%	
Number of RPM macros	25			

6. Conclusions

The ANC using the proposed algorithm is designed and implemented using FPGA. The Xilinx Spartan-3E (XC3S500E) is used as a target device and the Xilinx ISE 10.1i is used as a programming tool. The design of AAE-ASSLMS algorithm is implemented using a schematic design entry. The connection between MATLAB and Xilinx ISE is made and all the speech signal samples (with samples equals to 28001) are entered to the Xilinx ISE. This connection will overcome the problem of entering the overall samples by hands which is a hard task or impossible. The difference between the error signal that is obtained by the Xilinx ISE Simulator and MATLAB

simulation is very small. This means that the selecting word length for fixed-point representation of the adaptive filter tap weights is proper, and the result quantization error is very small. The step-size parameter μ(n) plays an essential role in AAEASSLMS algorithm stalling. The 18-bits fixed-point representation of μ(n) is sufficient to prevent the overall system from the stalling. The slow down and saturation phenomenon can be eliminated by proper choice of data and coefficient word length. The 18-bits fixed point word length of weight coefficients eliminated the effects of the slowdown and saturation phenomenon. An embedded multiplier scheme is preferred when the number of embedded multipliers in the used FPGA is sufficient to the hardware design and speed is the main requirement for the application. The final hardware design can be used for the embedded applications as System on Chip (SoC) for ANC in many portable applications like mobile phone and hearing aids.

7. References

[1] A. Elhossini, S. Areibi and R. Dony, : An FPGA implementation of the LMS adaptive filter for audio processing. in Proc. IEEE International Conference on Reconfigurable Computing and FPGAs, pp.1-8 Sept. (2006) .

[2] B. Widrow and S. D. Steams : Adaptive Signal Processing. Englewood Cliffs, Prentice-Hall, (1985).

[3] Daniel J. Allred, Walter Huang, Venkatesh Krishnan, Heejong Yoo, and David V. Anderson: An FPGA Implementation for a High Throughput Adaptive Filter Using Distributed Arithmetic. in the Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'04), USA, 20-23 April (2004).

[4] Joseph Petrone : Adaptive Filter Architectures for FPGA Implementation. M.Sc. Thesis, Florida state university, college of engineering (2004).

[5] Marc Vella, Carl J. Debono : The Implementation of a High Speed Adaptive FIR Filter on a Field Programmable Gate Array. IEEE MELECON, Benalmadena (Malaga), Spain, May 16-19 (2006).

[6] Mohammed Bahoura, and Hassan Ezzaidi : FPGA-Implementation of a Sequential Adaptive Noise Canceller using Xilinx System Generator. IEEE International Conference on Microelectronics, pp. 213- 216, (2009).

[7] Thamer M.Jamel, and Haider Abd Al-Latif Mohamed : Absolute Average Error- Based Adjusted Step Size LMS Algorithm for Adaptive Noise Canceller. Mosharaka International Conference on Wireless Communications and Mobile Computing (MIC-WCMC2011), Istanbul, Turkey, 3-5 June (2011).

[8] Thamer M.Jamel, and Haider Abd Al-Latif Mohamed : Noise Canceller Using a New Modified Adaptive Step Size LMS Algorithm. WSEAS Transaction on signal processing , vol.10 , pp. 637-644,(2014).

[9] T. Lan, and J. Zhang : FPGA Implementation of an Adaptive Noise Canceller. IEEE International Symposiums on Information Processing (ISIP'08), pp. 553–558, Moscow, Russia, (2008).

[10] Wolfgang Fohl, Jörn Matthies, and Bernd Schwarz : A FPGA-based Adaptive Noise Cancelling System. Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09), Como, Italy, September 1-4 (2009).

[11] Waldeck and N. Bergmann : Evaluating software and hardware implementations of signal-processing tasks in an FPGA . in Proc. IEEE International Conference on Field-Programmable Technology. Brisbane, Australia, pp. 299-302. Dec. (2004).

برمجة مصفوفة المجال المنطقية باستخدام خوارزمية اقل معدل للتربيع ذو معامل الخطوة المتغيرة زمنيا لأغراض منظومة الغاء الضوضاء المتكيفة

أ.م. د. ثامر محمد جميل

قسم هندسة الاتصالات/ الجامعة التكنولوجية

بغداد / العراق

م. م. حيدر عبد اللطيف محمد

قسم هندسة الاتصالات/ الجامعة التكنولوجية

بغداد / العراق

الخلاصة :

يتناول هذا البحث بناء منظومة الغاء الضوضاء المتكيفة والتي تستخدم خوارزميتنا المقترحة سنة 2014 (خوارزمية اقل معدل للتربيع ذات معامل الخطوة المتغير زمنيا) باستخدام مصفوفة برمجة المجال المنطقية . كانت النتائج المستخلصة من هذا البناء المادي والمشتخلصة من برنامج الماتلاب متقاربة جدا وهذا يدل على ان البناء المادي كان ناجحا . اهم الاستنتاجات من هذا البحث هي امكانية استخدام البناء المادي المقترح في الاجهزة المحمولة كجهاز الهاتف النقال او اجهزة مساعدة ضعاف السمع.