FPGA IMPLEMENTATION OF MODULAR ACTIVE NOISE CONTROL

By

TAMILARASAN.P

Reg. No. 1020106018

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University, Coimbatore)

COIMBATORE - 641049

A MINI PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF ENGINEERING

IN

APPLIED ELECTRONICS

MAY 2011

BONAFIDE CERTIFICATE

Certified that this project report entitled "**FPGA IMPLEMENTATION OF MODULAR ACTIVE NOISE CONTROL**" is the bonafide work of **TAMILARASAN.P** [Reg. no. 1020106018] who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide

Mr.S.Venkatesh M.E.,

for Head of the Department

Dr. Mrs. Rajeswari Mariappan Ph.D.,

The candidate with university Register no. 1020106018 is examined by us in the project viva-voce examination held on ..23-05-2011.......

23/5/11

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I express my profound gratitude to our chairman **Padmabhusan Arutselvar Dr.N.Mahalingam B.Sc., F.I.E.,** for giving this opportunity to pursue this course.

At this pleasing moment of having successfully completed the project work. I wish to acknowledge my sincere gratitude and heartfelt thanks to our beloved Principal **Dr.S.Ramachandran, Ph.D.,** for having given me the adequate support and opportunity for completing this project work successfully.

I express my sincere thanks to **Dr.Ms.Rajeswari Mariappan, Ph.D.,** Head of the Department of Electronics and Communication Engineering, who rendering us all the time by helps throughout this project

In particular, I wish to thank and everlasting gratitude to the project coordinator **Ms.R.Hemalatha, M.E.,** Asst. professor(SRG). Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success. With her careful supervision and ensured me in the attaining perfection of work.

I extend my heartfelt thanks to my internal guide **Mr S.Venkatesh, M.E.,** Asst.Professor(SRG) for his ideas and suggestion, which have been very helpful for the completion of this project work.

Last, but not the least, I would like to express my gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering department for their encouragement and support throughout the course of this project.

ABSTRACT

This project presents the High-speed field programmable gate array (FPGA) implementation of a modular architecture of the Active Noise Control (ANC) system with online secondary path modelling. The clock frequency that is obtained for this FPGA implementation is 120MHz. A comparision of this implementation to other FPGA and DSP implementations of an ANC system shows a better speed up and convergence time for the proposed design.

This innovative modular implementation of the ANC system results in a fast design and fast convergence with capability to expand the four adaptive filters and data bus when higher speed for ANC system. Considering the modularity of the design and reconfigurability of FPGA, one can expand an ANC system for different accuracies and require convergence times.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The reduction of Acoustic noise has become a serious problem as the number of industrial equipment such as engines, blowers, fans, compressors and transformers are increased. In recent years powerful digital signal processing (DSP) devices have made possible the development of real time ANC systems with a wide range of applications. DSP algorithms have been developed for single and multiple channel ANC systems in branches of broadband, narrowband and adaptive feed-forward control.

1.1. PROJECT GOAL

This project proposes the FPGA implementation of the modular architecture of the ANC system. This new modular implementation of the ANC system results in a fast design and fast convergence with capability to expand the four adaptive filters and data bus when higher speed for ANC system is desired.

1.2 OVERVIEW

However, with its growing die size as well as incorporating the embedded digital signal processing blocks, the FPGA devices have become a serious contender in the signal processing market. The processor in a DSP approach needs some time to execute the software program. The FPGA approach, in the other hand, performs the tasks on a hardware base and it is faster than the DSP approach. Because of the need for much speed in these systems, FPGA devices are good candidates for these applications.

1.3 SOFTWARE USED

- ModelSim XE III 6.2g
- Xilinx ISE 9.2i

1.4 ORGANIZATION OF THE REPORT

- **Chapter 2** discusses about summary of ANC system.
- **Chapter 3** discusses about implemention.
- **Chapter 4** discusses the ANC System.
- **Chapter 5** discusses the LMS Algorithm.
- **Chapter 6** discusses the comparision.
- **Chapter7** discusses the simulation results.
- **Chapter 8** shows the conclusion of the project.

CHAPTER 2

A Summary of Modular Design of an ANC System

This ANC system has three inputs and one output. The inputs of this ANC system are the reference noise signal, *x(n)*, the injected noise signal, *v(n)*, and the error microphone output, *e(n)*, inputted to the ANC system. The output of this system is the anti-noise signal, which is propagated by the loudspeaker and combines with the primary noise signal to reduce the noise pressure around the error microphone.

The modular design for this ANC system consists of four similar modules. The architecture of these modules is the same, but their inputs and outputs are different. These four modules are composed to form an ANC system with online secondary path modeling. The detail of this ANC system using the four M modules is shown in Figure 2.1.

In this Figure, *P(z)* is the primary acoustic path between the reference noise source and the error microphone. The reference noise signal, *x(n)*, is filtered through *P(z)* and appears as a primary noise signal at the error microphone. The output of the primary path filter, *d(n)*, is the desired output of the ANC system which is the convolution of the reference signal, *x(n)* and the tap-weights of the primary path FIR filter, ***P****(n)*, i.e.

$$d(n) = \sum_{k=0}^{N-1} p_k(n) x(n-k)$$

where *N*, is the length of the primary path FIR filter.The objective of the ANC system is to generate an appropriate anti-noise signal, *a(n)*, propagated by the loudspeaker to create a zone of silence in the vicinity of the error microphone. The module M used in this design has six inputs and one output. The block diagram of this module is shown in Figure 2.2.

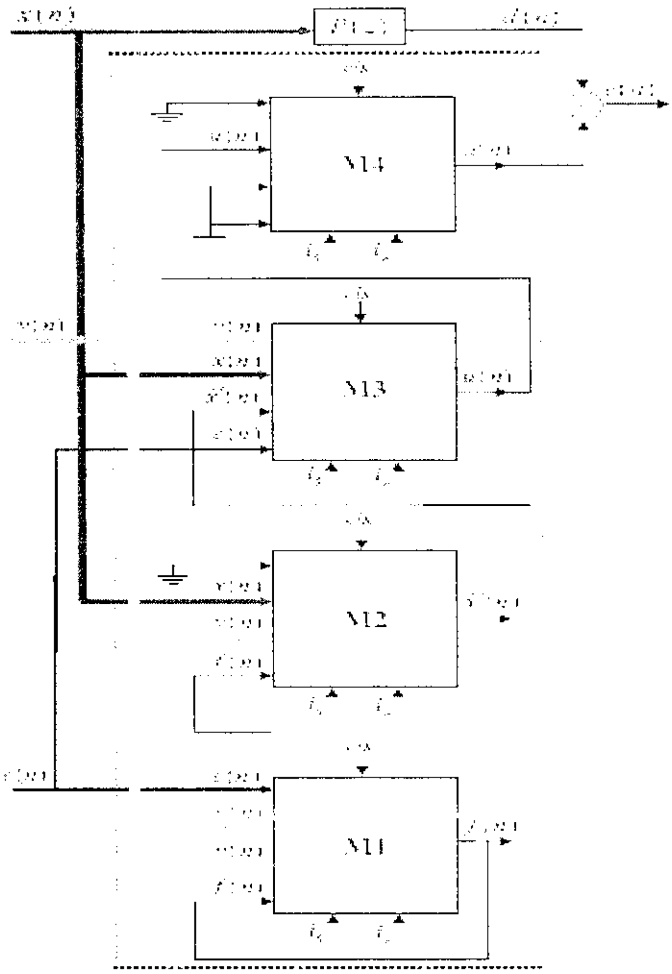


Figure 2.1 Architecture of the ANC system with online secondary path modeling using four M modules

Each module consists of two shift registers, a convolution unit, an adaptive filter controller, a 2's complementer, and an adder.

The first input to the module, *i1(n)*, represents the desired output of the adaptive filter at time *n*. The second input, *i2(n)*, represents the input data to the adaptive filter at time *n*. The third input, *i3(n)*, represents the modified input data to the adaptive filter at time *n*. The forth input, *i4(n)*, represents the error data at time *n*. The fifth input, *i5*, is the input data to the step size parameter register. The sixth input, *i6*, is a control signal to negate the output of the module.

The output of the module, *z(n)*, is the difference between the adaptive filter's output and the desired output at time *n*. Two shift registers with the length of *K* are installed in the inputs of the block to save and shift the input data and the modified input data at each time *n*.

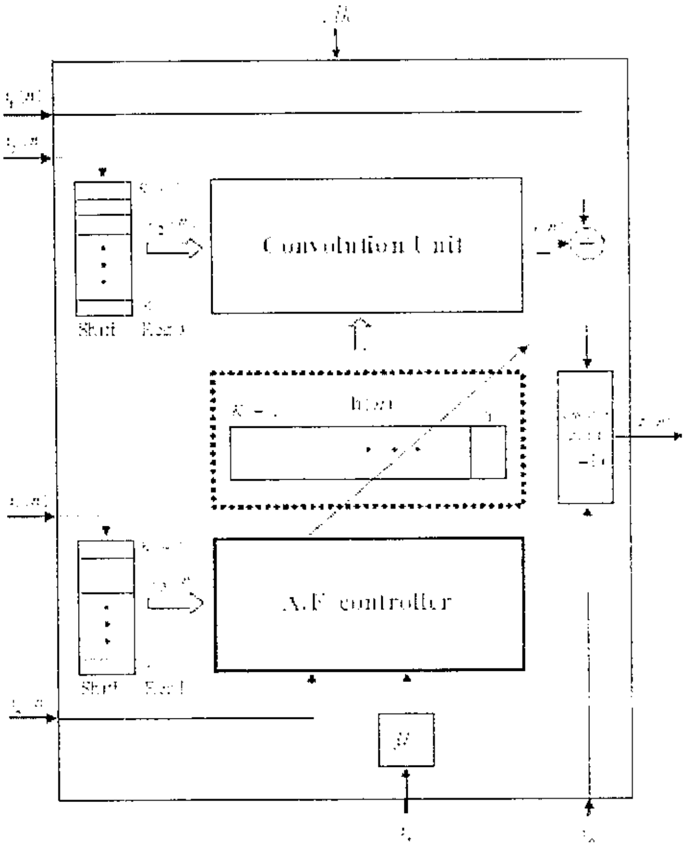


Figure 2.2 Block diagram of the module M

The digital filter *H(z)* contains *K* tap-weights. The output of this unit at time *n* is the convolution of the input vector, *i2(n)* and the tap-weight vector of the filter, *h(n)*.

$$y(n) = \sum_{i=0}^{K-1} h_i(n) i_2(n-i)$$

The tap-weights of this digital filter, *hi(n)*, are updated at each time using the Least Mean Square (LMS) algorithm, i.e.

$$h_k(n+1) = h_k(n) + \mu i_3(n) i_4(n-k), \quad k = 0, \ldots, K-1$$

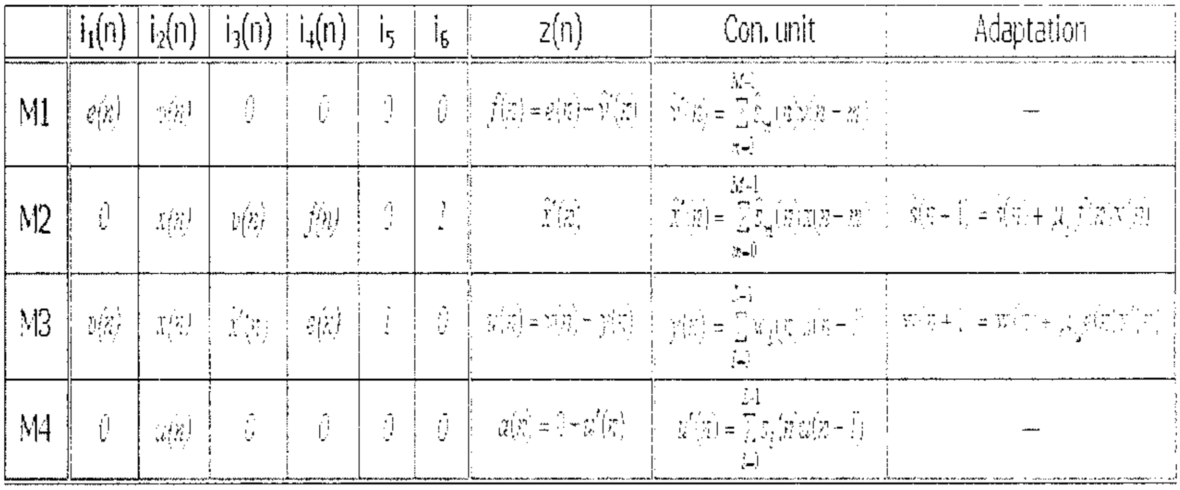
Where μ is the step size in the LMS algorithm, and *i3(n)* and *i4(n)* are the input values to the module at time *n*. If the control input to 2's complementer unit, *i6*, is one, then the 2's complement of the data input is obtained in the output. The adder in the module subtract the value of the output of the adaptive filter and the input *i1(n)*, i.e.

$$z(n) = i_1(n) - y(n)$$

The output of the adder is the output of the module. Four of these modules are used to form an ANC system with online secondary path modeling as shown in Figure 2.1.

The inputs, output and computations of these four modules are depicted in Table 2.1.

TABLE 2.1: Inputs, output and computations of four modules



	$i_1(n)$	$i_2(n)$	$i_3(n)$	$i_4(n)$	i_5	i_6	$z(n)$	Con. unit	Adaptation
M1	$e(n)$	$v(n)$	0	0	0	0	$f(n) = e(n) - \hat{v}'(n)$	$\hat{v}'(n) = \sum_{m=0}^{M-1} \hat{s}_m(n) v(n-m)$	—
M2	0	$x(n)$	$v(n)$	$f(n)$	0	1	$\hat{x}'(n)$	$\hat{x}'(n) = \sum_{m=0}^{M-1} \hat{s}_m(n) x(n-m)$	$\hat{s}(n+1) = \hat{s}(n) + \mu_s f(n) v(n)$
M3	$d(n)$	$x(n)$	$\hat{x}'(n)$	$e(n)$	1	0	$e(n) = d(n) - y(n)$	$y(n) = \sum_{i=0}^{L-1} w_i(n) x(n-i)$	$w(n+1) = w(n) + \mu_w e(n) \hat{x}'(n)$
M4	0	$a(n)$	0	0	0	0	$a(n) = 0 - a'(n)$	$a'(n) = \sum_{i=0}^{L-1} s_i(n) a(n-i)$	—

CHAPTER 3
IMPLEMENTATION

In this section the modular ANC system with online secondary path modeling is implemented on FPGA

These FPGAs have dedicated multiplier, pipeline and accumulator circuitries. With the embedded DSP blocks, these FPGAs can perform high speed calculations, therefore are useful devices for digital signal processing designs, as well as adaptive filters. The step size parameters of the adaptive control filter, μw, and the adaptive modeling filter, $\mu\_$, are considered to be 2-11 and 2-6, respectively. All the signals in ANC processor are considered to be floating point numbers with 32 bit length that comprises 11 accuracy bits. The number of used DSP blocks for implementing the ANC system on FPGA is 736 blocks. The clock frequency is 120MHz. A summary of implementation results is shown in Table 3.1.

TABLE 3.1: The number of elements used on Stratix FPGA, μw=2-11,$\mu\_$=2-6

Resource	Used	Avail	Utilization
IOs	103	1,171	8 %
Registers	4,050	150,386	2 %
ALUT	6,344	143,520	4%
DSP block 9-bit elems	736	768	95%

CHAPTER 4
ANC SYSTEM

ANC is a technique used to remove an unwanted noise from a received signal. the operation is controlled in an adaptive way in order to obtain an improved signal-to-noise ratio (SNR) . As shown in Fig4.1, an ANC is typically a dual-input, closed-loop adaptive feedback system. The two inputsare: the primary input signal d(n) (the desired signal corrupted by the noise) and the reference signal x(n) (an interfering noise supposed to be uncorrelated with the desired signal but correlated with the noise affecting the desired signal in an unknown way). The adaptive filtering operation achieved the best results when the system output is noise free, which means that the output SNR is infinitely large. Therefore, in the application, on obtaining the best result, we should put the reference sensor into the most appropriate place where the signal component of the primary sensor output is undetectable in the reference sensor output and the noise component of the primary sensor output is highly correlated with the reference sensor output.

ANC technique has been successfully applied to many applications, such as acoustic noise reduction, adaptive speech enhancement and channel equalization. In these cases and other related high speed required applications, pure software implementation would bring about long processing time, as shown in the last part of section IV, thus can not meet the requirement. An effective way can be represented by a hardware implementation on FPGA

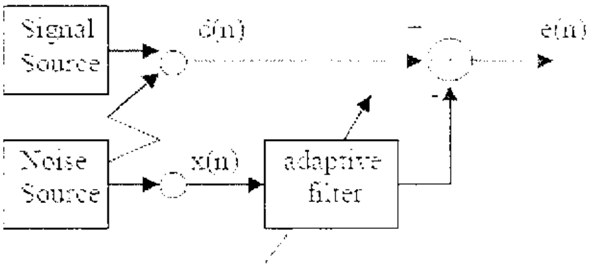


Figure 4.1 ANC system

CHAPTER 5

LMS ALGORITHM

The LMS algorithm is a widely used algorithm for adaptive filtering. The algorithm is described by the following equations:

$$y(n) = \sum_{i=0}^{M-1} wi(n) * x(n-i)$$

$$e(n) = d(n) - y(n)$$

$$wi(n+1) = wi(n) + 2ue(n)x(n-i).$$

In these equations, the tap inputs x(n), x(n-1),......,x(n-M+1) form the elements of the reference signal x(n), where M-1 is the number of delay elements. d(n) denotes the primary input signal, e(n) denotes the error signal and constitutes the overall system output. wi(n) denotes the tap weight at the nth iteration. the tap weights update in accordance with the estimation error. And the scaling factor u is the step-size parameter. u controls the stability and convergence speed of the LMS algorithm.

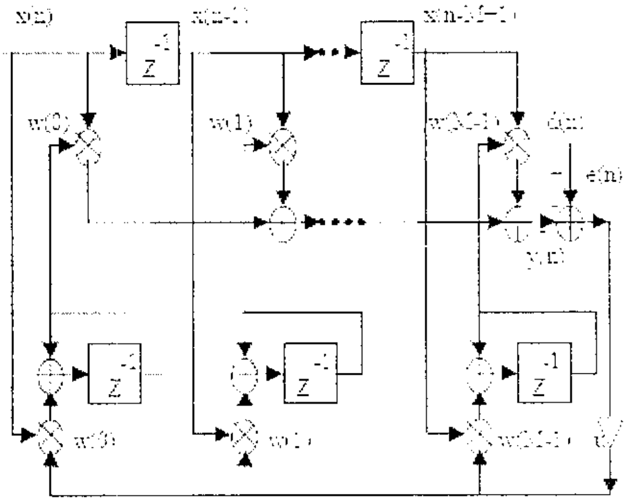


Figure 5.1 Flowchart of the LMS Algorithm

The LMS algorithm is convergent in the mean square if and only if u satisfies the condition:

0 < u < 2 / tap-input power

where tap-input power,

$$= \sum_{k=0}^{M-1} E[|u(n-k)|^2]$$

There are usually two ways to implement the LMS algorithm, hardware implementation and software implementation . The hardware implementation of the algorithm in an FPGA has good real-time ability, but requires large resources. From Fig.5.1, we can see that an N-tap adaptive filter requires at least 2N multipliers and 2N adders. The software implementation consumes trivial amount of resources, however, the low speed of which makes it uncommonly used.

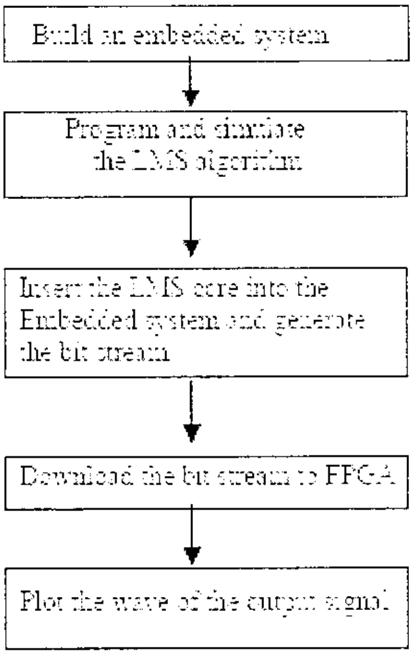

Build an embedded system
Program and simulate the LMS algorithm
Insert the LMS core into the Embedded system and generate the bit stream
Download the bit stream to FPGA
Plot the wave of the output signal

Figure 5.2 Flowchart of the FPGA Implementation Process

CHAPTER 6

COMPARISION

To evaluate the proposed implementation, the results were compared with the results of implementations. The ANC systems in these researches have no secondary path. where, our ANC system has secondary path modelling filter.

The digital signal processors uses a programmable, 16-bit, fixed-point format with the clock frequency of 40MHz. The computational time needed to execute an iteration of the ANC system on this DSP processor is about 78.1 usec. The acceptable convergence, 350 iterations is needed, leads to a total convergence time of about 27.3ms (=350*78.1 us).

The proposed FPGA implementation of the modular ANC system in this paper uses a 32-bit fixed pointformat. The adaptive control filter and the adaptive modeling filter have 32 tap eights. The clock frequency that obtained for this FPGA implementation is 120 MHz. The system needs 66.7 ns to execute an iteration. The design needs 190 iterations for an acceptable convergence. This results lead to a total convergence time of about 12.67us (= 190*66.7ns).

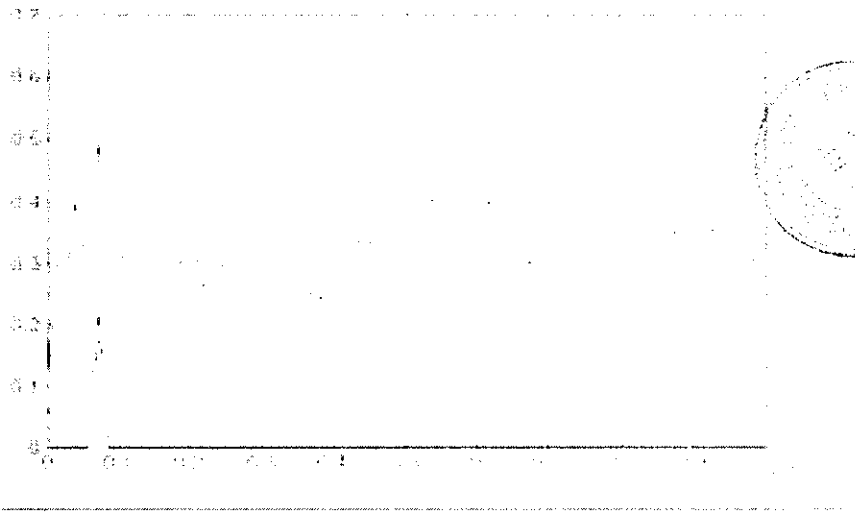


Figure 6.1 Noise Signals

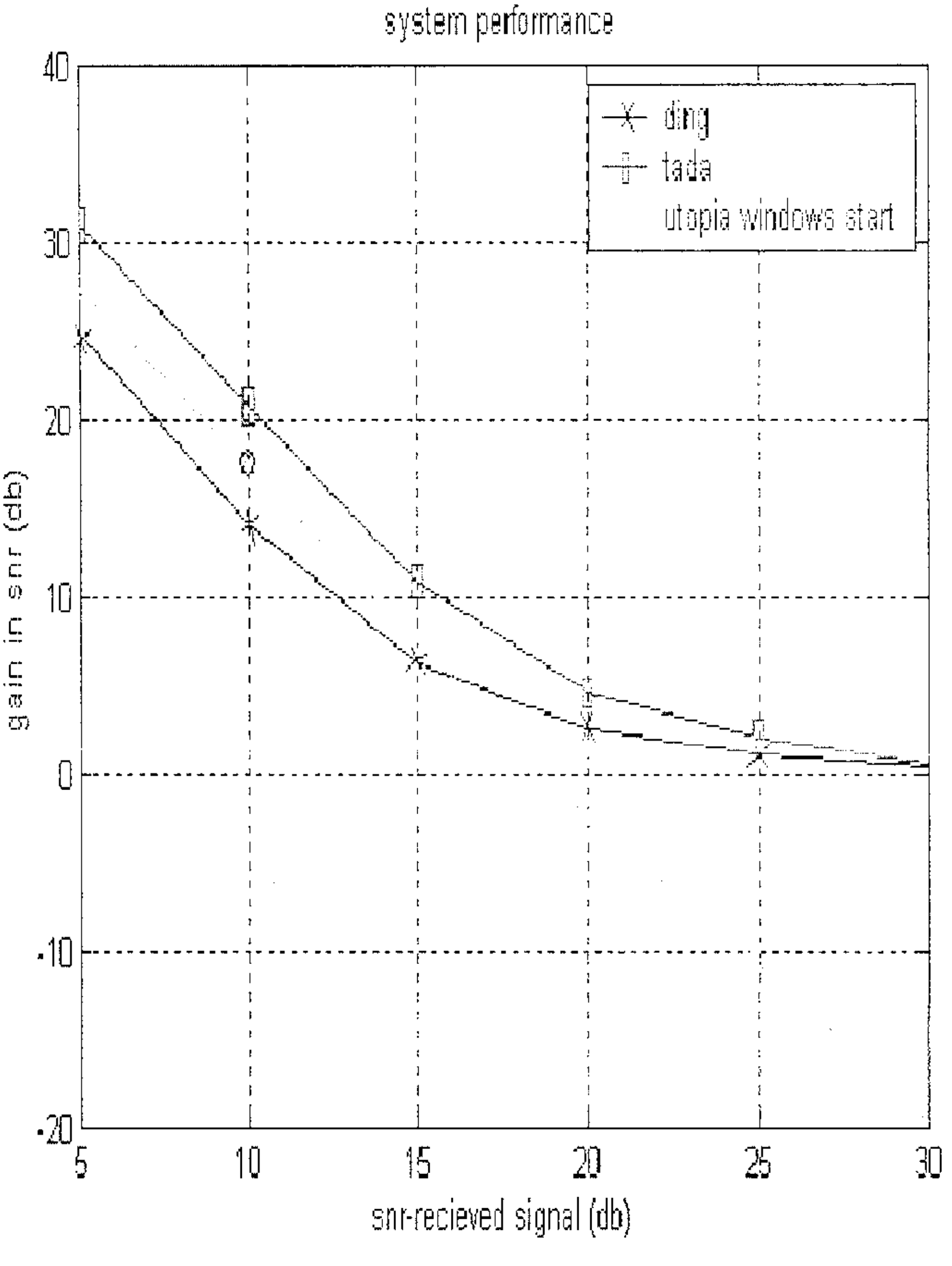


Figure 6.2 System Performance

CHAPTER 7
RESULTS AND DISCUSSION

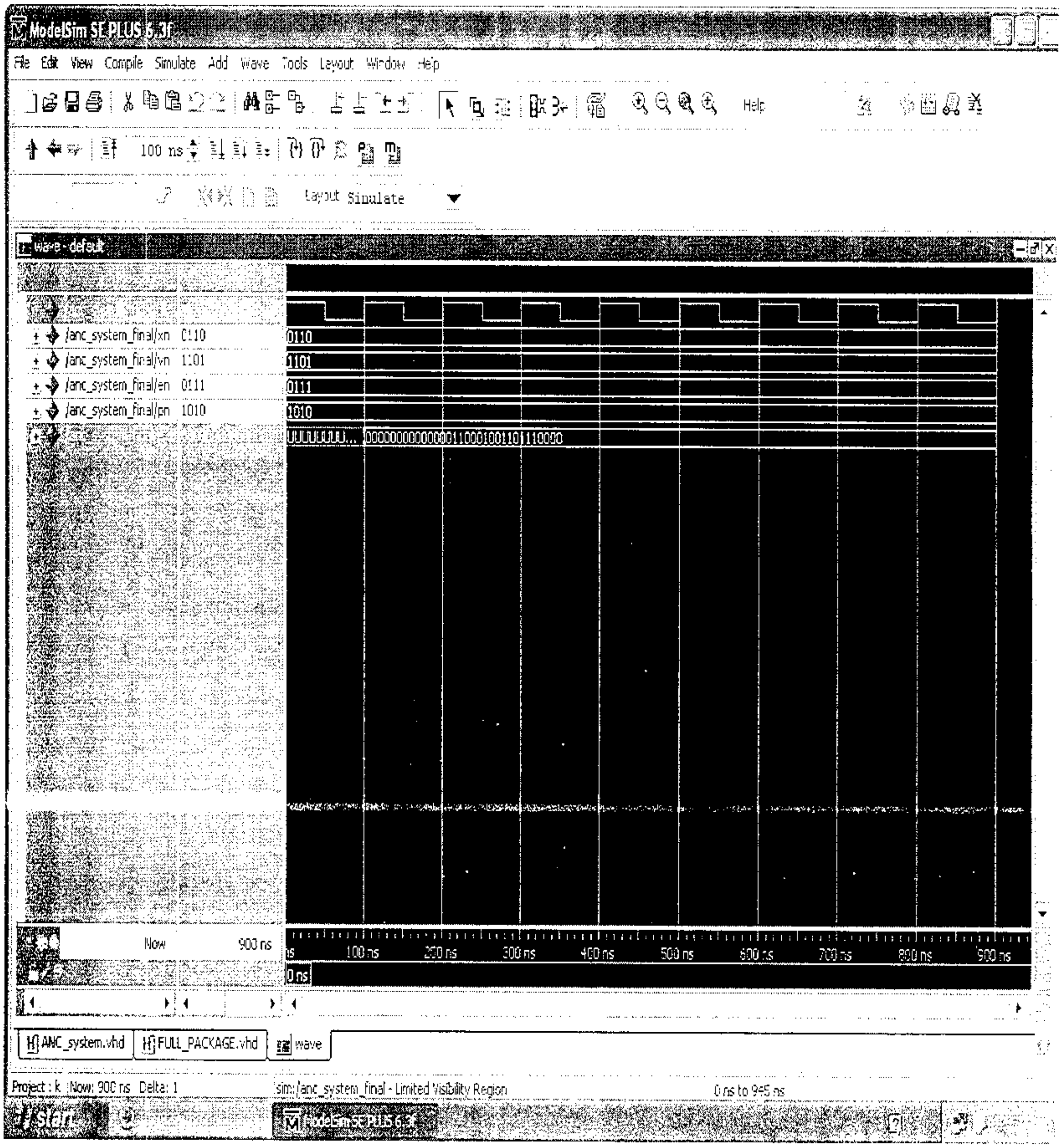
The simulation of this project has been done using MODELSIM XEIII 6.2g and XILINX ISE 9.2i.

Modelsim is a simulation tool for programming {VLSI} {ASIC}s, {FPGA}s, {CPLD}s, and {SoC}s. Modelsim provides a comprehensive simulation and debug environment for complex ASIC and FPGA designs. Support is provided for multiple languages including Verilog, SystemVerilog, VHDL and SystemC.

Xilinx was founded in 1984 by two semiconductor engineers, Ross Freeman and Bernard Vonderschmitt, who were both working for integrated circuit and solid-state device manufacturer Zilog Corp. The Virtex-II Pro, Virtex-4, Virtex-5, and Virtex-6 FPGA families are particularly focused on system-on-chip (SOC) designers because they include up to two embedded IBM PowerPC cores The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx. The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and-route (PAR), completed verification and debug using Chip Scope Pro tools, and creation of the bit files that are used to configure the chip.

The simulation results showing the FPGA implementation of modular active noise control system are attached here.

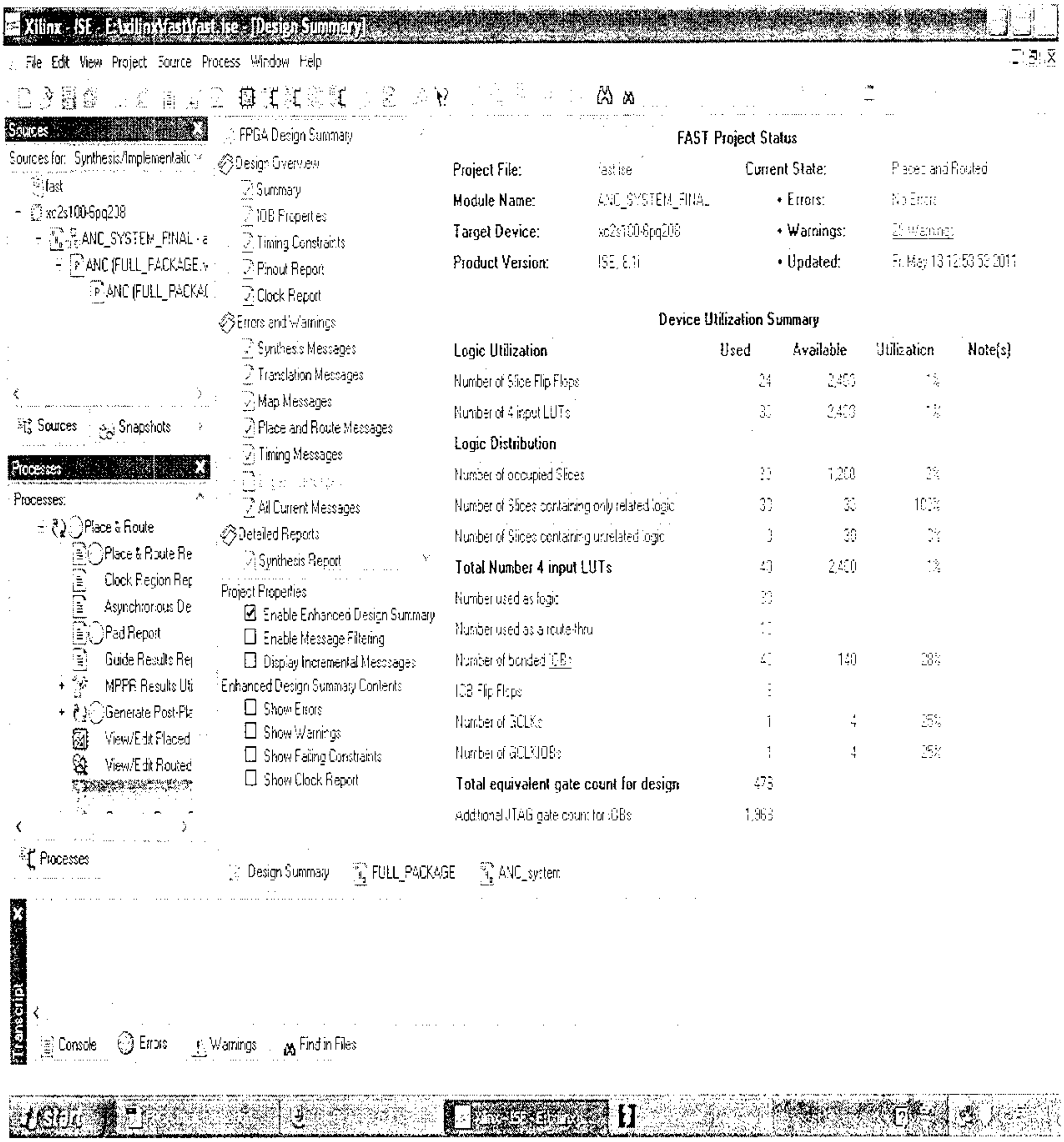
.


ModelSim SE PLUS 6.3f
File Edit View Compile Simulate Add Wave Tools Layout Window Help
100 ns
Layout Simulate
wave - default
Now 900 ns
ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns 700 ns 800 ns 900 ns
0 ns
ANC_system.vhd
FULL_PACKAGE.vhd
wave
Start
ModelSim SE PLUS 6.3f

```
/anc_system_final/xn  C110   0110
/anc_system_final/vn  1101   1101
/anc_system_final/en  0111   0111
/anc_system_final/pn  1010   1010
                             UUUUUUUU... 0000000000000011000100110 1110000
```

```
Project : k  Now: 900 ns  Delta: 1     sim:/anc_system_final - Limited Visibility Region     0 ns to 945 ns
```

Figure 7.1 Simulation result of the Output Waveform



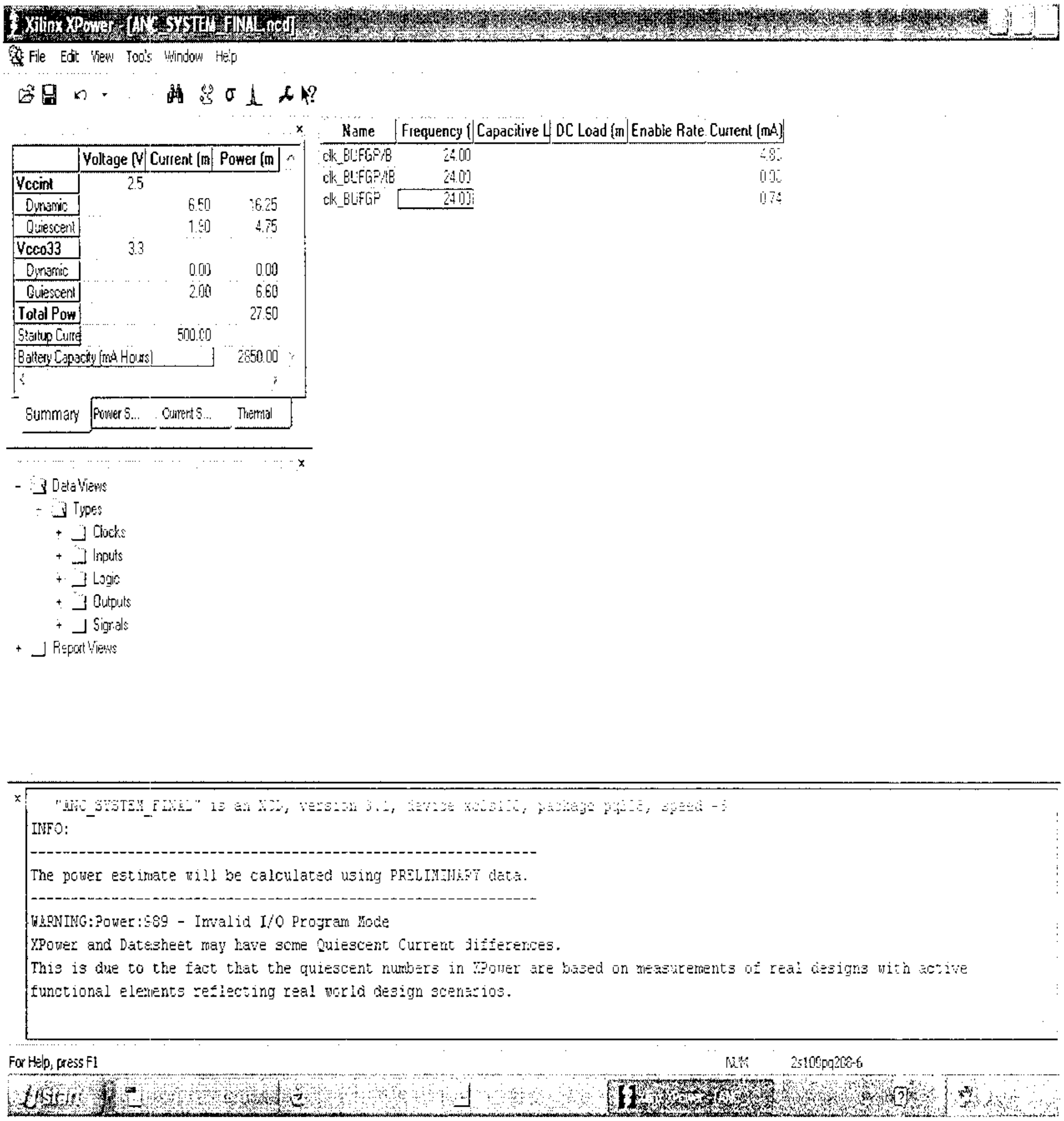
FAST Project Status

Project File:	fast.ise	**Current State:**	Placed and Routed
Module Name:	ANC_SYSTEM_FINAL	• **Errors:**	No Errors
Target Device:	xc2s100-6pq208	• **Warnings:**	25 Warnings
Product Version:	ISE, 8.1i	• **Updated:**	Fri May 13 12:53:53 2011

Device Utilization Summary

Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	24	2,400	1%	
Number of 4 input LUTs	30	2,400	1%	
Logic Distribution				
Number of occupied Slices	30	1,200	2%	
Number of Slices containing only related logic	30	30	100%	
Number of Slices containing unrelated logic	0	30	0%	
Total Number 4 input LUTs	40	2,400	1%	
Number used as logic	30			
Number used as a route-thru	10			
Number of bonded IOBs	40	140	28%	
IOB Flip Flops	3			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	473			
Additional JTAG gate count for IOBs	1,963			

Figure 7.2 Simulation result showing Gate count used



	Voltage (V	Current (m	Power (m
Vccint	2.5		
Dynamic		6.50	16.25
Quiescent		1.90	4.75
Vcco33	3.3		
Dynamic		0.00	0.00
Quiescent		2.00	6.60
Total Pow			27.60
Startup Curre		500.00	
Battery Capacity (mA Hours)			2650.00

Name	Frequency (	Capacitive L	DC Load (m	Enable Rate	Current (mA)
clk_BUFGP/B	24.00				4.80
clk_BUFGP/IB	24.00				0.00
clk_BUFGP	24.00				0.74

```
  "ANC_SYSTEM_FINAL" is an NCD, version 3.1, device xc2s100, package pq208, speed -6
INFO:
--------------------------------------------------------------
The power estimate will be calculated using PRELIMINARY data.
--------------------------------------------------------------
WARNING:Power:989 - Invalid I/O Program Mode
XPower and Datasheet may have some Quiescent Current differences.
This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active
functional elements reflecting real world design scenarios.
```

Figure 7.3 Simulation result of the Power used

CHAPTER 8
CONCLUSION

In this project a hardware implementation of the modular ANC system with online secondary path modeling was presented. This ANC system contains four similar modules to reduce the unwanted primary noise. Each module consists of two shift registers, a convolution unit, an adaptive filter controller, an adder, and a two's complementer. One of these modules performs as the main adaptive control filter that uses an FxLMS algorithm to converge the tap-weights. Two modules work as the adaptive modeling filters, used LMS algorithm to model the secondary path filter. The last module works as a digital filter to produce the anti-noise output signal.

The ANC system was implemented on the Altera FPGA. The current FPGA implementation of the ANC system uses a 32- bit fixed point format. The adaptive control filter and the adaptive modeling filter have 32 tapweights. The clock frequency that obtained for this FPGA implementation is 120 MHz. The system needs 66.7 ns to execute an iteration. The proposed design needs 190 iterations for an acceptable convergence. This results lead to a total convergence time of about 12.67us (=190*66.7ns).

BIBLIOGRAPHY

[1] F. Nagi, J. Low, S. Li , "Active Noise Cancellation with TMS320C5402 DSP Starter Kit," *IEEE Signal processing* , 2004.

[2] A. D. Stefano, A. Scaglione, C. Giaconia, "Efficient FPGA Implementation of an Adaptive Noise Canceller," in *Proc. IEEE 2005, CAMP'05*, vol. 00, 2005, pp. 87-89.

[3] A. Jalali, Sh. Gholami Borouj and M. Eshghi, "Design and Implementation of a Fast Active Noise Control System on FPGA," in *Proc. IEEE MED'07*, Athens, Greece, 2007.

[4] M. Eshghi, Sh. Gholami Boroujeny and A. Jalali, "A Hardware Design for an Online Active Noise Control System," *IEEE MED'07 Conference*, Athens, Greece, 2007.

[5] M. T. Akhtar, M. Abe, and M. Kawamata, "A New Structure for Feedforward Active Noise Control Systems With Improved Online Secondary Path Modeling," IEEE Trans. Speech Audio Process.,vol. 13, 2003, pp. 155-158 .

[6] M. Zhang, H. Lan. and W. Ser, "Cross-updated active noise control system with online secondary path modeling," *IEEE Trans.Speech Audio Process.*, vol. 9, pp. 598-602, July 2001.

[7] A. Carini. S. Malatini and G. Sicuranza, "Optimal Variable Step-Size NLMS Algorithms for Feedforward Active Noise Control." In *Proc EUSIPCO 2008*, Lausanne Switzerland. August 25-29. 2008.

[8] Sh. Gholami Boroujeny, M. Eshghi, "A Modular Design for an Active Noise Control System," *IEEE TENCON Conference*, Hyderabad, India, November 2008.

[9] 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*, Sept. 2006,