Adaptive LMS Filter for Noise Cancellation Based on FPGA

Wang Shenming

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Computer Engineering

Prince of Songkla University

2011

Thesis Title	Adaptive LMS Filter for Noise Cancellation Based on FPGA
Author	Mr. Wang Shenming
Major Program	Computer Engineering

Major Advisor:

.....

(Asst. Prof. Dr. Wannarat Suntiamorntut)

Examining Committee:

.....Chairperson

(Asst. Prof. Dr. Nattha Jindapetch)

.....

(Asst. Prof. Dr. Wannarat Suntiamorntut)

.....

(Dr. Rachaporn Keinprasit)

The Graduate School, Prince of Songkla University, has approved this thesis as partial fulfillment of the requirements for the Master of Engineering Degree in Computer Engineering.

.....

(Prof. Dr. Amornrat Phongdara)

Dean of Graduate School

Thesis Title	Adaptive LMS Filter for Noise Cancellation Based on FPGA
Author	Mr. Wang Shenming
Major Program	Computer Engineering
Academic Year	2010

ABSTRACT

The concept of adaptive filter is used in many applications, especially digital signal processing is demanded, because it can adjust the coefficients to obtain a better performance as possible. In this thesis, we present two types of LMS filters (standard LMS and normalized LMS (NLMS)) which applied in noise cancellation system. Firstly, we certified the availability of LMS algorithms based on MATLAB. We introduce the corresponding concepts of noise cancellation system, the noise cancellation algorithms, low power consumption techniques and resources sharing technique. Then we implemented and synthesized LMS algorithms on Xilinx ISE 10.1. We also simulated and tested the power consumption on Xpower analyzer. In addition, we intensively analyzed the original and modified circuits of the two types of LMS filters. Finally, we analyzed the experiment results. We compared the original circuits and modified circuits, and found that the modified circuits have a better performance on logic resources usage and power consumption.

Keywords: Hearing Aids, Noise Cancellation, LMS Algorithms, Low Power Techniques

ACKNOWLEDGEMENT

The thesis, together with the 2 and a half years study here is coming to an end. At this moment, I would like to extend my sincere thanks to all those who have concerned about or helped the writing.

My deepest gratitude goes first and foremost to Dr. Wannarat Suntiamorntut and Dr. Nattha Jindapetch, they gave instructive advice and useful suggestions to me on my thesis. Their understanding, patience, encouragements have been the most important factors which increase my confidence extremely.

Second, I would like to express my heartfelt gratitude to all lecturers in Department of Computer Engineering, who helped me a lot on my study. Additionally, I would like to thank the graduate School, the staffs of Faculty of Engineering and the staffs of Department of Computer Engineering for giving the convenience for me.

Third, I would like to thank my friends in the lab of WSN for their supports and assistances.

Finally, my thanks would go to my beloved family for their loving considerations and great confidence in me all through these years.

Wang Shenming

CONTENTS

CONTENTS(CONTINUED)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES(CONTINUED)

Figure Page

LIST OF ABBREVIATIONS AND SYMBOLS

ANC	Adaptive Noise Cancellation
ADC	Analog to Digital Converter
BTE	Behind the Ear
CTC	Completely in the Ear
DFG	Data Flow Gragh
IC	Integrated Circuit
ITE	In the Ear
ITC	In the Canal
LMS	Least-Mean-Square
MAC	Multiplier Accumulate
NLMS	Normalized LMS
RTL	Register Transfer Level
SD	Steepest Descent
SSM	Steady State Miss-adjustment
SNR	Signal to Noise Ratio

CHAPTER 1

INTRODUCTION

1.1 Motivation

In our daily life, there are many people who are trouble in some types of hearing loss. According to the investigation, approximately 10% of the world's population suffers from the hearing problem [1]. To design and produce the device to help the people who are trouble in the hearing loss, it is very valuable.

Hearing aids system is the one of the most important contributions for a human being. It is a small electronic instrument which makes sound louder and reduces the unnecessary noise signal to make a good quality of the speech. The hearing aid is designed to pick up sound waves with a tiny microphone, change weaker sounds into louder sounds and pass them to the ear through a tiny speaker. With the microchips available and the integrated circuit (IC) applied today, hearing aids have got smaller and smaller and have significantly improved quality.

In the past, people used the analog circuit to implement the hearing aid function. The typical analog hearing aids are similar to a simple radio. The volume bass and treble can be tuned and adjusted, but hearing loss is not simple. It requires specific technique that is not only adjusted a technical loss of volume. Rather than, hearing loss is very sensitivity of the appeared and unnecessary noise signal. The limited and defect of traditional analog hearing aid is that cannot separate the speech signal from the mixed sound and lead to distortion. Though there is a lot of people trouble in various hearing loss problems, yet only a small percentage of them use a hearing aid.

Digital signal processing has been used in many applications, such as telecommunications, consumer electronics, speech processing, image processing and so on. The hearing aid industry also uses digital signal processing. The difference between speech and background noise can be separated using the digital technology, allowing filtering the noise out. This leads us to develop the hearing aid device using digital signal processing.

Many of the hearing aids are electronically programmable. The programmable parameters are downloaded from a computer-based fitting system and stored in digital registers. The register outputs are used to switch resistor networks that control various analog circuitries. Further advances in hearing aid technology will require a good sophisticated signal processing. It can be argued that only a limited signal processing complexity can be achieved with analog circuitry and that this limit has been reached. The limit is imposed by size constraints and fundamental transistor scaling considerations. It is necessary to reduce the size of the elementary circuits for the complex hearing aid circuits.

Analog hearing aids are already marginally acceptable; another reason is the limited of hearing aid size. The potential for achieving greater complexity with digital processing is more encouraging. The overall size of a digital processor is determined by the number of digital logic used. In digital technology, it can use several techniques to reduce logic element utilization, such as scheduling & resources sharing. There is another problem state of the hearing aid system. In order to extend hearing aid battery life, it is necessary to reduce the power consumption. We use the low power design technique to reduce the dynamic power consume of the circuit.

The hearing aid is used to process the speech signal. It reduce the noise signal to obtain a clearly speech for a patient. The adaptive filter is widely used in this area. Such as the LMS filter.

In this research work, we propose the low power LMS circuits with fixed step size u using resource sharing technique. In the content of this thesis, we firstly describe the noise cancellation system for hearing aid device and various adaptive LMS filter algorithms such as LMS algorithm with fixed step size μ and LMS algorithm with adjust step size μ. Secondly we introduce several types of logic resources sharing and low power design technology. We then translate the LMS algorithm with fixed step size u and LMS algorithm with adjust step size μ to circuit data flow graph (DFG). The scheduling and resources sharing technology have been applied to implement the original LMS circuit and the modified LMS circuit. Those designs have been developed using Verilog and simulated on Xilinx ISE Simulator. Finally, we analyze the design based on the simulation results and provide a conclusion about the research work.

1.2 Objectives

1.2.1 To review and find out a good Least-Mean-Square (LMS) algorithm suitable for adaptive noise canceller (ANC) in hearing aid system.

1.2.2 To select the appropriated step size μ based on Matlab.

1.2.3 To implement the low power LMS algorithm using resources sharing.

1.3 Scope of Work

This thesis is aiming for the design and implementation of the low power LMS adaptive filter on Field Programmable Gate Array (FPGA). The adaptive algorithm is simulated on Matlab. The important parameter step size μ and filter order are firstly evaluated on Matlab to choose the good result with a fast convergence speed and small miss-adjustment error. Then later, the design will apply both resources shearing technology and low power design technology to make a good performance and simulate the circuits on Xilinx ISE.

1.4 Work Plan

1) Review the research problem of noise cancellation, literature review and exercise some problem issues.

2) Identify the problem and write a research proposal.

3) Analyze some good candidates of LMS algorithm to find out a better one for this research.

4) Implement and simulate the selected LMS algorithm based on Matlab. Determine some important parameters. Such as the number of the filter tap and the step size μ.

5) Translate the selected LMS algorithm formulas to data flow graphs (DFGs). Implement the original LMS circuits based on Xilinx IES.

6) Improve the LMS circuit in term of performance and power consumption by using resources sharing technique and low power design technique.

7) Compare the performance between the original circuit and modified LMS circuit.

8) Testing/Debugging & measure the logic resources utilization and power consumption on Xpower Analyzer.

9) Conclusions & collect the results for writing the final report.

1.5 Thesis Outline

This document has been organized into 5 chapters as follows:

Chapter1: Introduction. In this chapter, the motivation, objective and scope of this thesis are presented. It presents the work plan to evaluate and investigate the LMS algorithm and low power consumption technology.

Chapter2: Research background. In this chapter, we introduce the adaptive noise cancellation (ANC) system and the hearing aid system. And then we explain several LMS algorithms for the ANC system. At last, the power dissipation sources, some low power design techniques and resource sharing technique are described.

Chapter3: Implement low power LMS filter. In this chapter, we determine the design statement (such as representation of data, bit-extension and truncation). And then we certify the tracking ability and the convergence rate of LMS algorithm with different step size μ. The several speech signals are simulated based on MATLAB tool. At last we describe the traditional fixed step size μ LMS filter circuit and the NLMS circuit. We also introduce and describe the new modified circuit of LMS filter.

Chapter 4: Experiment Results. In the chapter, we show the logic usage of the traditional LMS circuit and the modified LMS circuit. We also compare the power consumption between the two types of the circuits.

Chapter 5: Conclusion and Discussion. We conclude and discuss this work.

CHAPTER 2

RESEARCH BACKGROUND

2.1 Introduction of Hearing Aids

A hearing aid device was placed in or around the ear. It helps to improve the hearing of people who loss the hearing. It makes sounds louder so that a person with hearing loss can listen, communicate and participate better in daily activities. And it can help people to have a good quality of hearing in noisy situations. The basic components of a hearing aid are microphone, signal processor, a speaker and a battery. The microphone receives and converts the sound into an electric signal. The signal then undergoes processor that can be as simple as amplifying all of the sound equally, to more advanced equalization involving a digital signal processor. The speaker converts the electronic signal back to sound, and the battery powers the electronics.

There are four main styles of hearing aids on the market today. They are shown in Figure 2.1. They are classified as following:

Behind the ear (BTE): The BTE style sits behind the ear with a clear tube going to an ear mold in the ear to deliver the sound. A variation on this style is called an open-fit-behind-the-ear (OTE) where the ear mold is replaced by a small tip, resulting in a more open feeling. Other variations include replacing the tube with wires and moving the receiver from the behind the ear to inside the ear.

In the ear (ITE): The ITE style moves the hearing aid into the outer ear, where it becomes a single unit with the ear mold. This style fills up most of the outer ear and appears as a solid mass.

In the canal (ITC): The ITC style moves some of the hearing aid into the ear canal and reduces the space taken up in the outer ear, but is still plainly visible.

Completely in the canal (CIC): The CIC style is the smallest of them all, as it fits completely inside the ear canal, thus nearly disappearing from view.

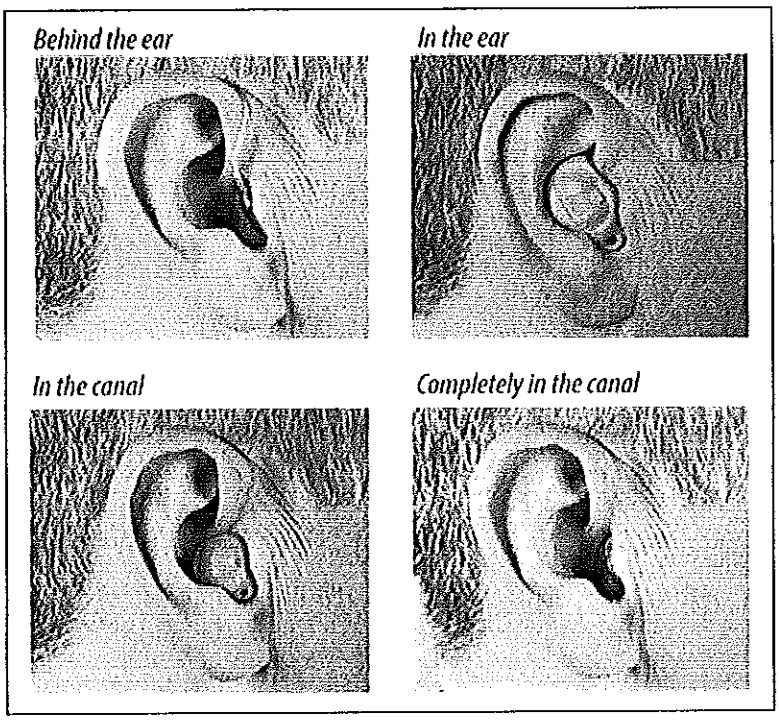


Figure 2.1 Styles of hearing aid

From the types of technology for hearing aids, they are classified into analog and digital.

Analog hearing aids convert sound waves into electrical signals and process electrical signals in the analog domain. The earliest analog hearing aids simply amplified both speech and noise, and are custom built to meet the needs of each user. It is called analog/adjustable hearing aids. Newer analog hearing aids can be programmed during the fitting process, and the user can change the program for different listening environments. It is named analog/programmable hearing aids.

Digital hearing aids convert sound waves into numerical codes, similar to the binary code of a computer, before amplifying them and process the numerical codes in the digital domain. Digital hearing aids are also programmable during the fitting process and have multiple listening profiles that are selectable by the patient. The digitization of sound allows more advanced signal processing such as noise reduction, filtering, and acoustic feedback control. The vast majority of hearing aids used digital circuit because of their increased performance and flexibility over the analog versions.

It is also classified single microphone hearing aid and multiple microphones hearing aid; depend on the number of the microphone usage. In this work, we will use two microphones as the signal input ports.

2.2 Noise Cancellation System

Noise cancellation technique [2] is widely used in all kinds of consumer electronics (EC) market, industrial areas and medical equipment such as headphone, electrocardiogram (ECG) machine and hearing aid. For example, in automatic speech recognition from noise-corrupted speech, the noise cancellation schemes provide an improved quality of speech signal that helps in achieving a better recognition performance. A single sensor noise cancellation system is described in [3], [4]. The general single-microphone noise cancellation system is shown in Figure 2.2.

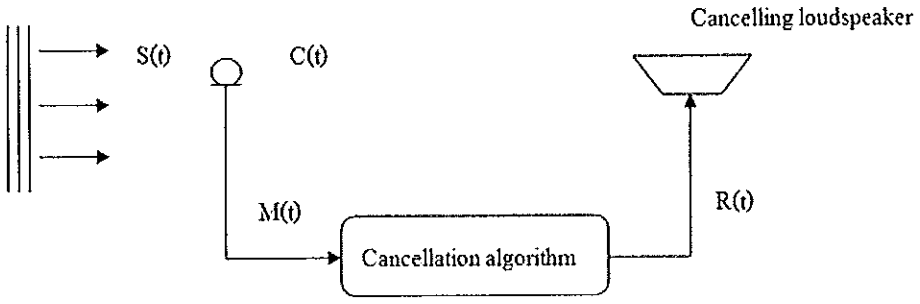


Figure 2.2 General single-microphone noise cancellation system [3].

However, the single sensor technique exhibit performance limitations under real operational scenarios, such as signal distortion. Moreover, their performance strongly depends on noise characteristics and a satisfactory performance is only expected when the observation noise can be assumed as stationary. It is to be noted that in real-life applications, most of the cases, the properties of signal and noise are varying with time, such as noisy environments in cars, factories and markets.

In order to overcome such limitations of the single sensor noise cancellation techniques, (Figure 2.3) the adaptive noise cancellation (ANC) technique are described in [5] [6]. In the ANC system, in addition to the primary sensor capturing the noisy observations, a

secondary estimated noise can be subtracted from the primary sensor resulting in the desired signal.

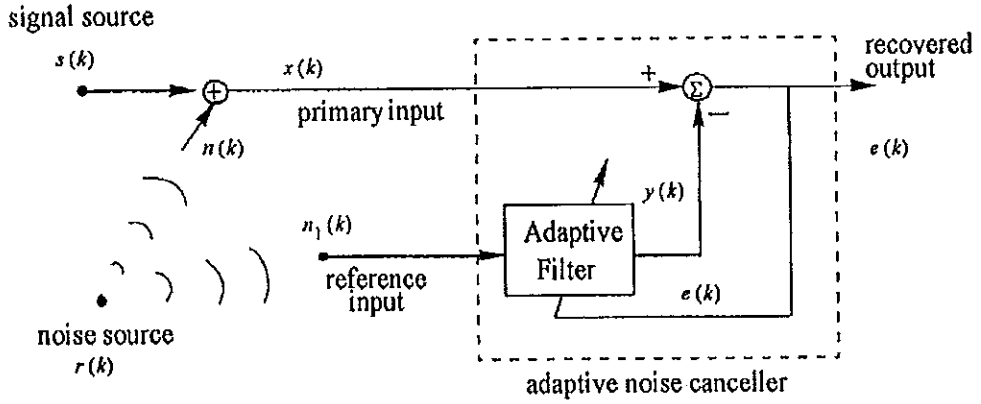


Figure 2.3 The adaptive noise cancellation (ANC) system [6].

The sum and the noise provide the primary and reference inputs for the adaptive canceller using finite impulse response (FIR) as a digital adaptive filter. Normally the adaptive filter weights are updated using least-mean-square (LMS) algorithm. In the next section, the several types of LMS algorithm are described and analyzed.

2.3 Adaptive Noise Cancellation Algorithm

The concept of adaptive noise canceling was proposed [7] by Widrow and Hoof in 1960. The basic idea is to subtract out a filtered version of some signal from the noise corrupted desired signal. The parameter of the filter is continuously modified by some algorithms so as to optimize performance in the result.

2.3.1 Standard LMS Algorithm

The LMS algorithm [8] [9] [10] [11] is one of the most widely used algorithm in the adaptive Noise Cancellation. One reason is its simplicity and robustness to the static signal. It continuously updates the tap-weight vector by the estimates of the recently tap-weight values. The LMS algorithm derivation based on the steepest descent (SD) algorithm [8]. The main formulas of LMS algorithm are shown in equations (2.1 to 2.3).

$$y(n) = \sum_{k=0}^{M-1} w_k(n)u(n-k) = W(n)^T U(n) \qquad (2.1)$$

$$e(n) = d(n) - y(n) \qquad (2.2)$$

$$w_k(n+1) = w_k(n) + 2ue(n)u(n-k) \qquad (2.3)$$

The input vector $d(n)$ includes the speech source signal and other noise signal. The filter output is $y(n)$. And the weight vector is $W(n)$. $e(n)$ is the error between the desired signal and the filter output $y(n)$. Another important parameter is the filter step size is μ.

The input matrix is $U(n) = [u(n), u(n-1), ..., u(n-M+1)]$.
The tap weight matrix is $W(n) = [w_0, w_1, ..., w_{M-1}]$. So the equation 2.3 rewrite to equation 2.4.

$$\begin{bmatrix} w_0(n+1) \\ w_1(n+1) \\ \cdot \\ \cdot \\ \cdot \\ w_{M-1}(n+1) \end{bmatrix} = \begin{bmatrix} w_0(n) \\ w_1(n) \\ \cdot \\ \cdot \\ \cdot \\ w_{M-1}(n) \end{bmatrix} + ue(n) \begin{bmatrix} u(n) \\ u(n-1) \\ \cdot \\ \cdot \\ \cdot \\ u(n-M+1) \end{bmatrix} \qquad (2.4)$$

According to the LMS equations (2.1 to 2.3), the standard LMS algorithm is departed to three blocks. They are FIR filter block (F-Block) with coefficient vector $W(n)^T$ and input sequence $u(n)$, estimate output error block and the weight (coefficient) update block (WUD-Block). The system level diagram of LMS filter is shown in Figure 2.4. And Figure 2.5 shows the detail of LMS filter block diagram [9].

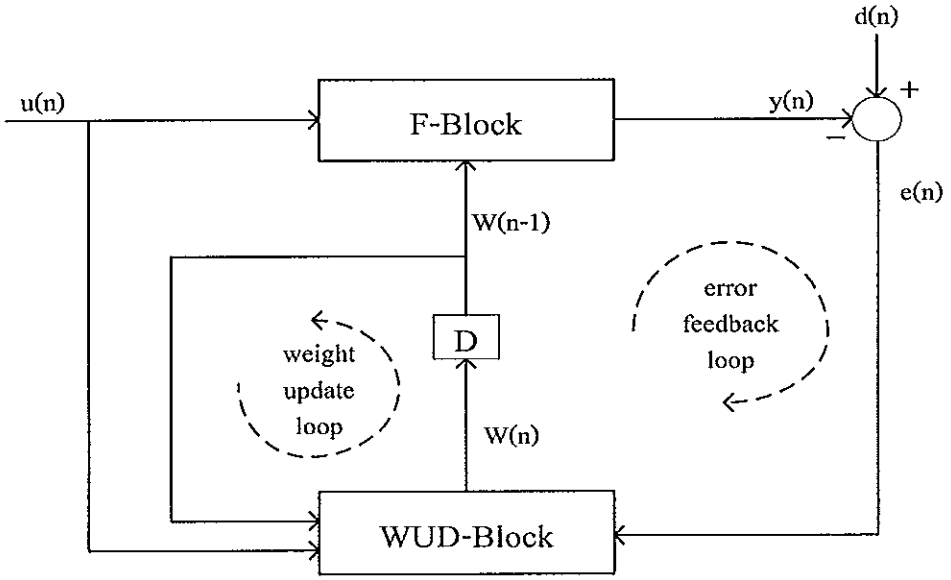


Figure 2.4 Schematic view of LMS algorithm [9]

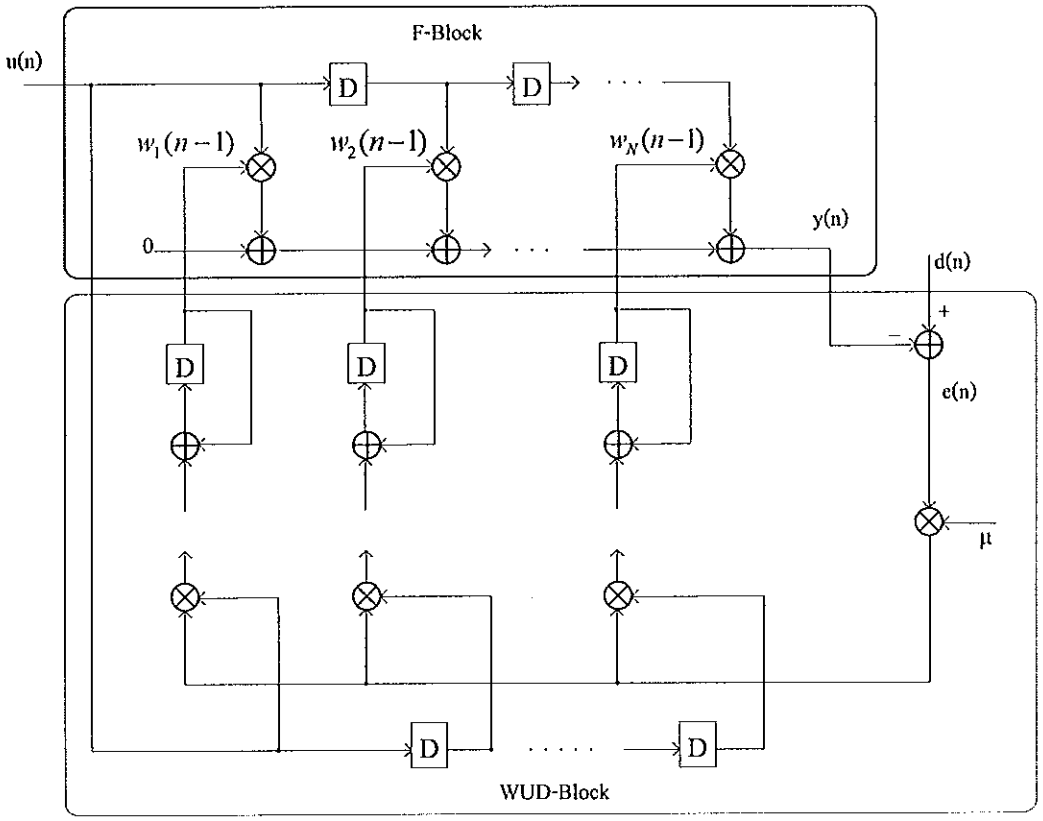


Figure 2.5 The LMS adaptive digital filter [9]

According to the structure of ANC system (Figure 2.3), the error output $e(n)$ is the signal which we expect to get. In order to finding the better results, it must calculate the optimum weight vector W_{opt}. So it is defined the square of the error output:

$$e^2(n) = (d(n) - y(n))^2 = (d(n) - \sum_{k=0}^{M-1} w_k(n)u(n-k))^2 \qquad (2.5)$$

The equation 2.5 is a binary quadratic function. And the surface graph is shown in Figure 2.6 [8]. It is a parabolic. To finding the optimum error signal, compute the gradient of the square error (equation 2.5).

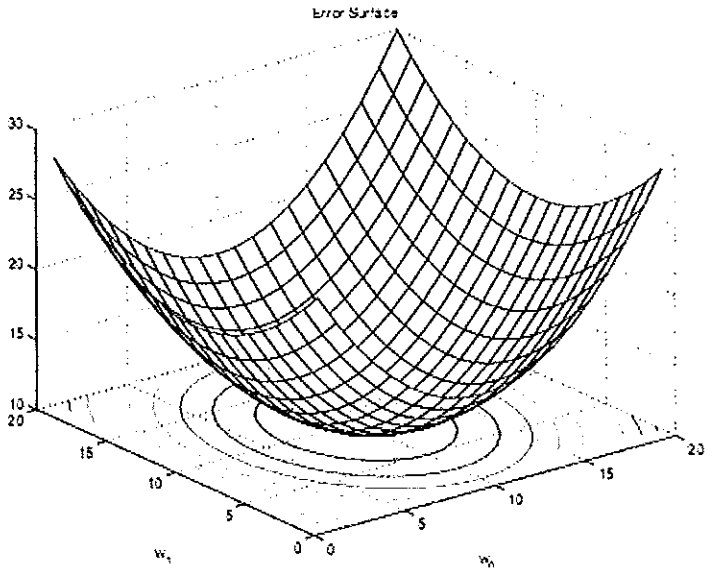


Figure 2.6 The square error surface graph [8]

The equation (2.5) is unfolded to:

$$\begin{aligned} e^2(n) &= d^2(n) - 2d(n)U(n)W(n) + W^T(n)U(n)U^T(n)W(n) \\ &= d^2(n) - 2R_{du}W(n) + W^T(n)R_{uu}W(n) \end{aligned} \qquad (2.6)$$

Compute the gradient of the square error and set it equate to zero, it got equation 2.7.

$$\nabla k = \frac{\partial e^2(n)}{\partial w} = -2R_{du} + 2R_{uu}W(n) = -2e(n)U(n-k) = 0 \qquad (2.7)$$

From the equation 2.7 the optimal W_{opt} will be found. $W_{opt} = R_{xx}^{-1} R_{dx}$. R_{uu} is correlation matrix of the filter input sequence $U(n)$. R_{du} is the cross-correlation matrix of $U(n)$, $d(n)$.

$$R_{uu} = U(n)U^T(n) \qquad R_{du} = d(x)U(n) \qquad (2.8)$$

According to steepest descent (SD) algorithm [8]:

$$W(n+1) = W(n) + u\nabla k \qquad (2.9)$$

Substituting equation (2.7) in (2.9), it got equation (2.3). The LMS algorithm initiated with some arbitrary value for the weight vector is seen to converge and stay stable for μ. Choosing an appropriate step size μ is very important because the step size μ can control the convergence rate of the LMS filter coefficients and determine the final excess mean-square error. The convergence of LMS algorithm is inversely proportional to μ. With a large step size μ, the filter will be fast convergence; however, it results in increased miss-adjustment error. Otherwise it will be slow convergence with smaller miss-adjustment error. Figure 2.7 shows the relationship between the convergence rate and the miss-adjustment error with different μ values.

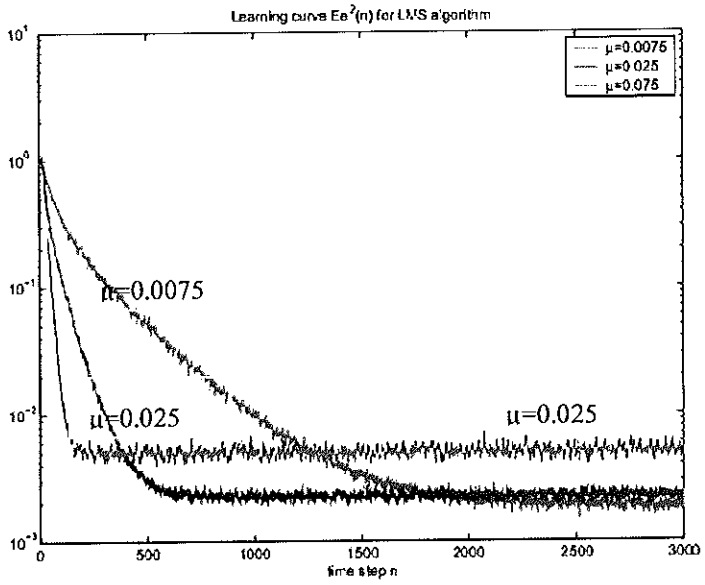


Figure 2.7 Effect of different step size μ value [8]

Normally $0 < u < \frac{1}{\lambda_{\max}}$, $\lambda_{\max}$ is the maximum eigenvalues of the correlation matrix R_{uu}.

Advantages & disadvantages of LMS algorithm:

(1) Simplicity in implementation.

(2) Stable and robust performance against different signal conditions.

(3) Slow convergence (due to eigenvalues spread).

2.3.2 The Normalized LMS Algorithm

The Normalized LMS (NLMS) [12] [13] [14] introduces a variable adaptation rate. It improves the convergence speed in a non-static environment. It just modified the tap-weight vector based on the standard LMS. The tap weight defined as:

$$w_k(n+1) = w_k(n) + \frac{a}{c + U^T U} e(n)u(n-k) \qquad (2.10)$$

Where $U^T U$ is autocorrelation matrix of the filter input sequence. a and c are constants. $\frac{a}{c + U^T U}$ may be thought of as a step size parameter μ which controls the rate of convergence of the algorithm and also its miss-adjustment. The constant c prevents division by the very small number of $U^T U$. So $u = \frac{a}{c + U^T U}$ and μ can adjust depend on the input data. By setting the variable step size μ and comparing the standard LMS and NLMS, there are some characteristics.

(1) When there is a similar average steady state square error, NLMS will be faster.

(2) When there is a similar convergence speed, NLMS achieves a lower average steady state square error.

(3) NLMS offers better trade-offs than the standard LMS.

(4) The computation complexity of NLMS is slightly higher than the standard LMS.

2.3.3 A Variable-step-size LMS Algorithm

To achieve the optimum convergence rate and steady-state miss-adjustment performance, the step size of the update equation is altered. This also improves the trade-off

between speed and the minimum mean-square error. The step size of a variable-step-size LMS algorithm [16] affects to the mean-square error corresponding to the formula below:

$$u(n+1) = \alpha u(n) + \xi e^2(n) \qquad (2.11)$$

Where $0 < \alpha < 1, \xi > 0$ and $u_{\min} < u < u_{\max}$, $u_{\min}$ and $u_{\max}$ are selected to limit the convergence rate as well as the steady-state miss-adjustment (SSM). A large error produces a large the step size to provide faster tracking. A small error produces a small the step size to reduce the miss-adjustment. A typical value of α found to work well in simulations is $\alpha = 0.97$. The parameter ξ is usually very small. In the reference [15], they choose $\xi = 4.8 \times 10^{-4}$. There is a better performance for adjusting the step-size. However, this algorithm results in a noisy step size and leads to a high steady-state miss-adjustment in the case of low signal-to-noise ratio (SNR). Another modified variable-step-size LMS was proposed in [16]. It updates the step size based on the time-averaged estimate of the autocorrelation of $e(n)$ and $e(n-1)$, the formula is :

$$u(n+1) = \alpha u(n) + \gamma p^2(n) \qquad (2.12)$$

$$p(n) = p(n-1) + (1-\beta)e(n)e(n-1) \qquad 0 < \beta < 1 \qquad (2.13)$$

β governs the averaging time constant. $p(n)$ is the time-averaged estimation of the correlation of $e(n)$ and $e(n-1)$. It assumes that the steady-state error $e(n)$ is uncorrelated and therefore results in a small value for the step size after convergence. A new variable-step-size LMS algorithm proposed in [14]. Taking the expectation of $u(n+1)$ then:

$$E[u(n+1)] = \alpha E[u(n)] + \gamma\beta^2 E[p^2(n-1)] + \gamma\beta(1-\beta)E[p^2(n-1)]E[\xi^2(n)] + (1-\beta)^2 E[\xi^2(n)] \qquad (2.14)$$

Where $\xi(n)$ is an irrelevant white noise with zero mean square. It is suitable when the input power is invariable and the irrelevant noise is very strong. On other condition, the computation is much bigger and need more operations.

2.3.4 Using The Signal to Noise Ratio (SNR) to Adjust The Step Size LMS Algorithm

An adjusted step size NLMS algorithm that the step size is controlled by the estimation of signal to noise ratio (SNR) is proposed in [17] and [18]. A small SNR will cause the step size to increase in order to make the tracking fast. When a large SNR will result in decreasing the step size, it could yield a smaller miss-adjustment. Figure 2.8 shows how the step size μ controlled by SNR.

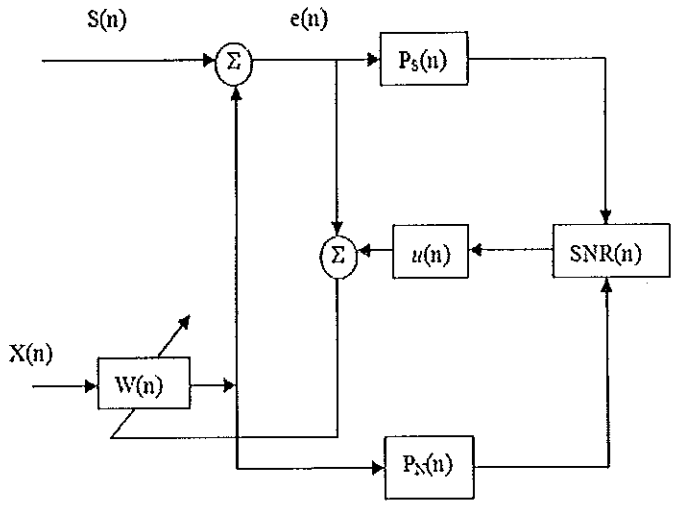


Figure 2.8 Adjust the step size LMS algorithm [18]

To estimate the SNR, the average power of the speech signal $p_s(n)$ and the average power of the noise signal $p_M(n)$ are defined by:

$$p_s(n) = \sum_{j=0}^{M-1}[x(n-j) - y(n-j)]^2 \qquad (2.15)$$

$$p_M(n) = \sum_{j=0}^{M-1} y(n-j)^2 \qquad (2.16)$$

M is the number of samples used for estimating the equation. The SNR defined by:

$$SNR(n) = 10 * \log_{10}(\frac{p_s(n)}{p_M(n)}) \qquad (2.17)$$

The tap-weight updated by:

$$W(n+1)=W(n)+\frac{u}{X^T(n)X(n)}e(n)X(n) \qquad (2.18)$$

Where μ is the step size coefficient. It is controlled by the SNR(n).

$$\mu=\begin{cases}\mu_{min} & \text{if } \mathrm{SNR}(n)>\mathrm{SNR}_{max}\\ \mu_{max} & \text{if } \mathrm{SNR}(n)<\mathrm{SNR}_{min}\\ a\cdot SNR(n)+b & \text{if } \mathrm{SNR}_{min}\le \mathrm{SNR}(n)\le \mathrm{SNR}_{max}.\end{cases} \qquad (2.19)$$

The constant u_{max} is obtained near the point of instability of the conventional LMS providing the maximum possible convergence speed. The value of u_{min} is chosen as a compromise between the desired level of steady state miss-adjustment and the required tracking capabilities of the algorithm. Its implementation requires 5N+4 multiplications and 5N-3 additions. The whole system is shown in Figure 2.9.

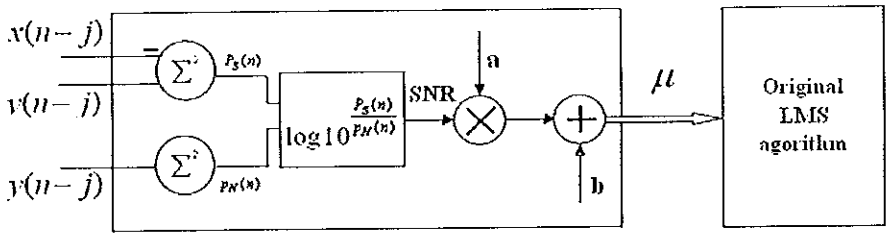


Figure 2.9 The whole system of SNR LMS algorithm

2.3.5 Compare with Those LMS Algorithms

Considering those types of LMS algorithm, the standard LMS algorithm is simple and easy to implement based on Field Programmable Gate Array (FPGA) or digital signal processer (DSP) device. However it cannot find the balance between the convergence rate and the state-steady miss-adjustment error. NLMS offers better trade-offs than the standard LMS. The computation complexity of NLMS is slightly higher than the standard LMS. The others adjust step size LMS algorithm have a better method to adjust the convergence rate and the state-steady miss-adjustment error. But the computation is complexity and use a lot of logic resources. So in

this work we will implement a fixed step size LMS algorithm and NLMS algorithm on Xilinx ISE & XPower Analyzer tool.

2.4 Overview of Low Power Methods Design

Low power is an essential constraint for hearing aids, therefore suitable low power strategy need to be applied. Low power VLSI techniques at different abstraction levels are applied while designing filter. Various types of methodologies are employed for low power design, such as algorithmic level, Structural level and Circuit level. Various filter applications designed at circuit level and FPGA based digital filters also have been studied. In [19], it presented a new hardware efficient approach for low power implementation of FIR digital filters. It described the FIR filter architecture of each implementation and their key components. And the power reduction of up to 34% is reported as compared to a conventional FIR filter implementation [20].

2.4.1 Power Dissipation Sources

The total power for an IC design consists of dynamic power and static power. Dynamic power is the power consumed when the device is active. Static power is the power consumed when the device is powered up but no signals are changing value. In CMOS circuits, the main contributions to the power consumption are from leakage current (static power), short-circuiting current and switching currents (dynamic power).

1. **Leakage Power**

Leakage currents are due to two sources: one from the currents that flow through the reverse biased diodes (reverse biased PN-Junction current), the other from the currents that flow through transistors that are non-conducting (sub-threshold channel conduction current). The leakage currents are proportional to the leakage area and exponential of the threshold voltage. The leakage currents are due to manufacturing technology and cannot be modified by the designers except in some logic styles. Sub-threshold leakage and reverse-biased junction leakage current; both increases dramatically with temperature and are independent of the operating voltage for a given fabrication process. The leakage current is in the order of pico-Ampere, but it increases as the threshold voltage is reduced. Leakage current is difficult to predict, measure or optimized.

Generally, leakage current serves no useful purposes, but some circuits do exploit it for intended operations, such as power-on reset signal generation. The leakage power problem mainly appears in very low frequency circuits or ones with "sleep modes" where dynamic activities are suppressed.

2. Short-Circuit Power

In a static CMOS circuit, there are two complementary networks: p-network (pull-up network) and n-network (pull-down network). The logic functions for the two net-works are complementary to each other. Normally when the input and output state are stable, only one network is turned on and conducts the output either to power supply node or to ground node and the other network is turned off and blocks the current from flowing. Short-circuit current exists during the transitions as one net-work is turned on and the other network is still active. For example, the input signal to an inverter is switching from 0 to V_{dd}. During this transaction, there exists a short time interval where the input voltage is larger than V_{tn} but less than $V_{dd} - |V_{tp}|$ During this time interval, both PMOS-transistor (p-network) and NMOS-transistor (n- network) are turned on and short-circuit current flows through both kinds of transistors from power supply line to the ground. The short-circuit current will consume power.

3. Switching Power

The switching currents are due to the charging and discharging of node capacitances. The node capacitances mainly include gate, overlapping, and interconnection capacitances. The power consumed by switching [21] current can be expressed as:

$$p = \alpha C_L f V_{dd}^{\ 2} \qquad (2.20)$$

Where α is the switching activity factor, C_L is the load capacitance, f is the clock frequency, and V_{dd} is the supply voltage. The above equation (2.20) shows that the switching power depends on a few quantities that are readily observable and measurable in CMOS circuits. It is applicable to almost every digital circuit and hence provides guidelines for the low power design. The power consumed by switching current is the dominant part of the power consumption. Reducing the switching current is the focus of most low power design techniques. For large capacitance circuits, reduction of the frequency is the best way to reduce the switching power.

The use of different coding methods, number representation systems, continuing sequences and data representations can directly alter the switching frequency of the design, which alters the switching power. The best method of reducing switching frequency is to eliminate logic switching that is not necessary for computation.

2.4.2 Low Power Techniques

Low power techniques can be discussed at various levels of abstractions: system level, algorithm and architecture level, logic level and circuit level [22]. Figure 2.10 shows some examples of the low power techniques at the different levels.

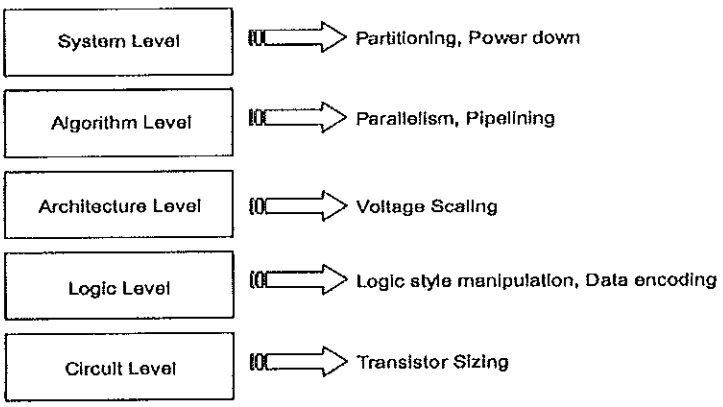


Figure 2.10 Low-power design methodologies at different abstraction levels.

In the following sections, an overview for different low power techniques has been described.

1. System Level

A system typically consists of both hardware and software components, which affect the power consumption. The system design includes the hardware/software design, hardware platform selection, resource sharing (scheduling) strategy, etc. The system design usually has the largest impact on the power consumption and hence the low power techniques applied at this level have the most potential for power reduction. At the system level, it is hard to find the best solution for low power in the large design space and there is a shortage of accurate power analysis tools at this level. However, if, for example, the instruction-level power models for given processors are available; software power optimization can be performed [23].

Clock gating is one of the most used low power techniques at system level. Clock gating [21] [24] is the most common way to reduce the power by turning clocks off when they are not required. Modern design tools support automatic clock gating: they can identify circuits where clock gating can be inserted without changing the function of the logic. Figure 2.11 shows how this works.

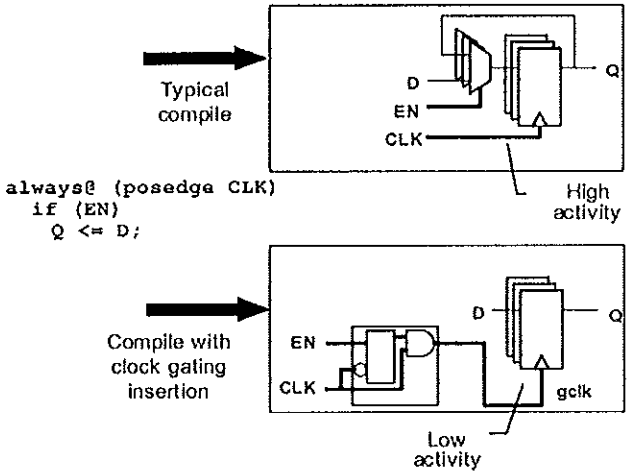


Figure 2.11 Clock gating technique [23]

Asynchronous design of the circuit can also be used as another low power designing technique. The asynchronous designs have many attractive features, like non-global clocking, automatic power-down, no spurious transitions, and low peak current, etc. It is easy to reduce the power consumption further by combining the asynchronous design technique with other low power techniques, for instance, multi-voltage scaling [21] technique as shown in figure 2.12.

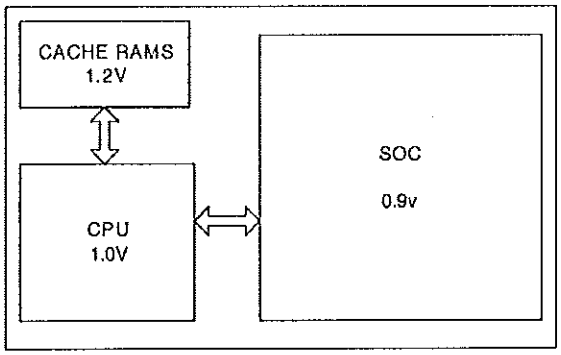


Figure 2.12 Multi-voltage architecture [23]

2. Algorithm Level

The algorithm selection has large impact on the power consumption. The task of algorithm design is to select the most energy-efficient algorithm that just satisfies the constraints. The cost of an algorithm includes the operate computation part and the communication/storage part. Reduction of the number of operations, cost per operation, and long distance communications are key issues to algorithm selection.

One important technique for low power of the algorithmic level is algorithmic transformations [25]. This technique exploits the complexity, concurrency, regularity, and locality of an algorithm. Reducing the complexity of an algorithm reduces the number of operations and hence the power consumption. The possibility of increasing concurrency in an algorithm allows the use of other techniques, e.g., voltage scaling, to reduce the power consumption. The regularity and locality of an algorithm affects the controls and communications in the hardware.

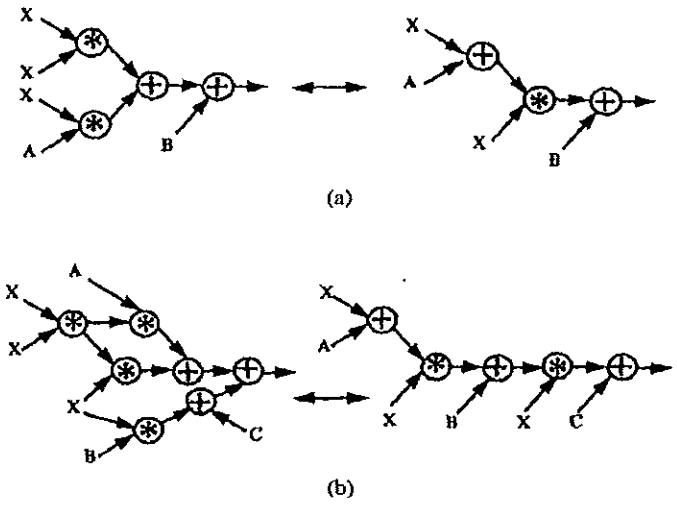


Figure 2.13 Unrolling and reducing the operations [26].

In figure 2.13, the unrolling reduces the critical path and the number of operations. Furthermore, this technique can be combined with other techniques at architectural level, for instance, pipeline and parallel technique [9], to save more power.

3. Architecture Level

According to the selection of the algorithm, the architecture can be determined for the given algorithm. From equation (2.20) we can say that, an efficient way to reduce the dynamic power consumption is the voltage scaling. When supply voltage is reduced, the power

consumption is reduced. However, this increases the gate delay. To compensate the delay, low power techniques like parallelism and pipelining [9] [27] architectures were used. The use of two parallel data path is equivalent to interleaving of two computational tasks. A data path to determine the largest number of C and (A + B) is shown in figure 2.14. It requires an adder and a comparator. The original clock frequency is 40 MHz [27].

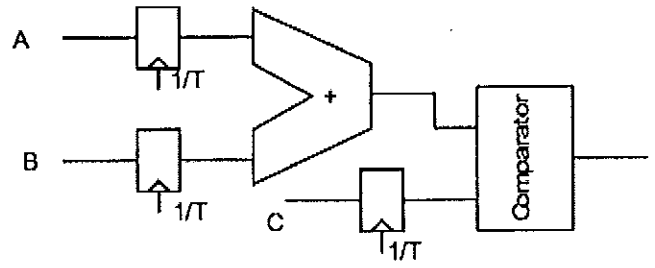


Figure 2.14 Original data path [27].

In order to maintain the throughput while reducing the power supply voltage, we use a parallel architecture. The parallel architecture with twice the amount of resources is shown in figure 2.15. The clock frequency can be reduced to half, from 40 MHz to 20 MHz since two tasks are executed concurrently. This allows the supply voltage to be scaled down from 5 V to 2.9 V.

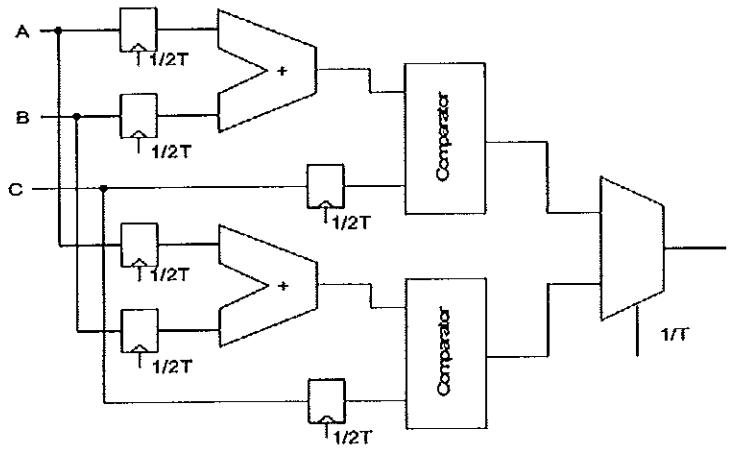


Figure 2.15 The parallel implementation [27].

Pipelining is another method for increasing the throughput. By adding a pipelining buffer/ register after the adder in figure 2.16.

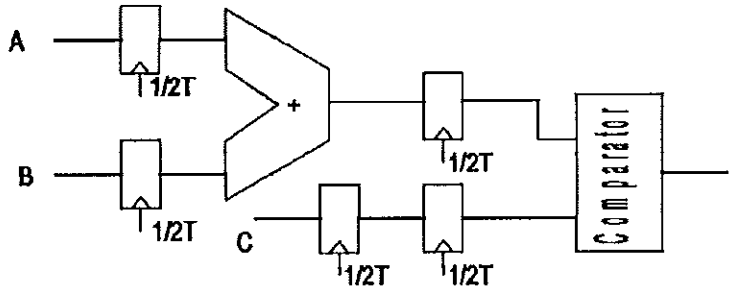


Figure 2.16 The pipelining implementation [27].

Main advantage of pipelining is the low area overhead in comparison with using parallel data paths. Another benefit is that the amount of glitches can be reduced. However, since the delay increases significantly as the voltage approaches the threshold voltage and the capacitance load for routing and/or pipeline registers increases, there exists an optimal power supply voltage. Reduction of supply voltage lower than the optimal voltage increases the power consumption.

4. Logic Level

The power consumption depends on the switching activity factor, which in turn depends on the statistical characteristics of data. The low power techniques at the logic level focus mainly on the reduction of switching activity factor by using the signal correlation and the node capacitances. In case of the gated clocking, the clock input to non-active functional block does not change by gating, hence, reduces the switching of clock network.

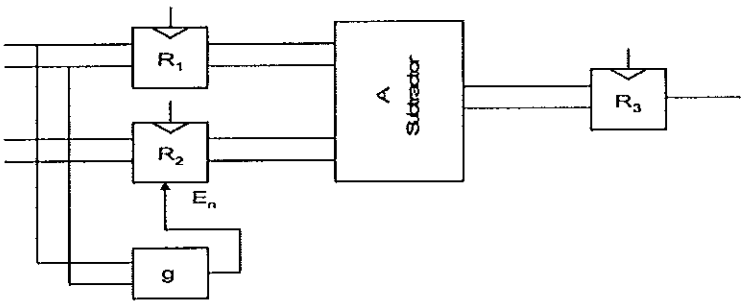


Figure 2.17 A pre-computation structure for low power [26].

Pre-computation [26] uses the same concept to reduce the switching activity factor: a selective pre-computing of the output of a circuit is done before the outputs are required,

and this reduces the switching activity by gating those inputs to the circuit. As shown in figure 2.17, the input data is partitioned into two parts, corresponding to registers R1 and R2. One part, R1, is computed in pre-computation block g, one clock cycle before the computation of A. The result from g decides gating of R2. The power can then be saved by reducing the switching activity factor in A.

Gate reorganization [26] is another technique used to restructure the circuit. This can be decomposition a complex gate to simple gates, or combines simple gates to a complex gate, duplication of a gate, deleting/addition of wires. The decomposition of a complex gate and duplication of a gate help to separate the critical and non-critical path; which reduce the size of gates in the non-critical path, as a result reduces the power consumption. In some cases, the decomposition of a complex gate increases the circuit speed and gives more space for power supply voltage scaling. The composition of simple gates can reduce the power consumption. The complex gate can reduce the charge/discharge of high-frequency switching node. The deleting of wires reduces the circuit size as a result, reduces the load capacitance. The addition of wires helps to provide an additional interconnection for better results.

5. Circuit Level

At the circuit level, the powers saving techniques are quite limited if compared with the other techniques at higher abstract levels. However, this cannot be ignored. The power savings can be significant as the basic cells are frequently used. A few percents improvement for D flip-flop can significantly reduce the power consumption in deep pipelined memory systems.

Many logic gates have inputs that are logically equivalent, i.e., the swapping of inputs does not modify the logic function of the gate. Some examples of gates are NAND, NOR, XOR, etc. However, from the power consumption point of view, the order of inputs does effect the power consumption. In addition to clock gating, there are a number of logic optimizations that the tools can perform to minimize dynamic power. Figure2.18 shows two of these optimizations [21].

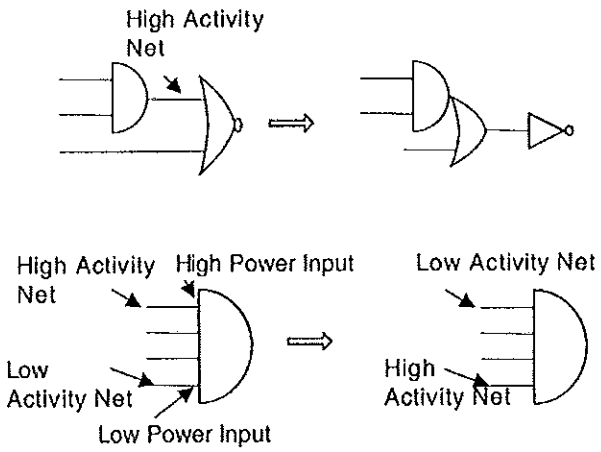


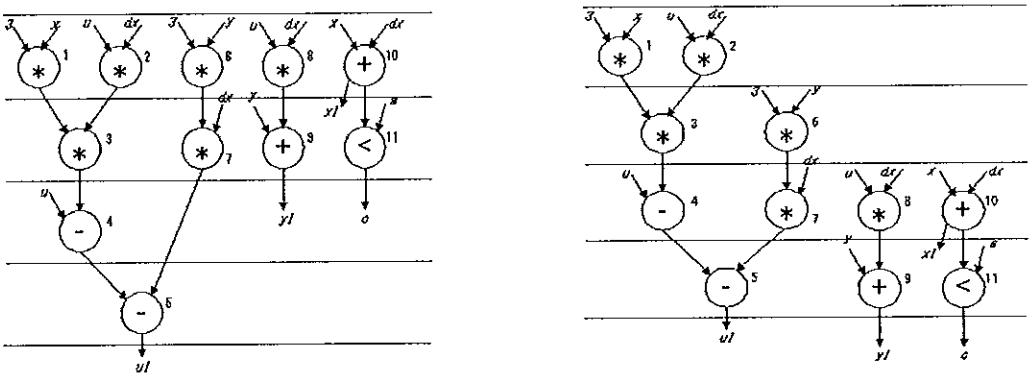
Figure 2.18 Examples of gate level optimizations [21].

At the top of the figure, an AND gate output has a particularly high activity. Because it is followed by a NOR gate, it is possible to remap the two gates to an AND-OR gate plus an inverter, so the high activity net becomes internal to the cell. Now the high activity node (the output of the AND gate) is driving a much smaller capacitance, reducing dynamic power.

At the bottom of the figure, an AND gate has been initially mapped so that a high activity net is connected to a high power input pin, and a low activity net has been mapped to a low power pin. For multiple input gates there can be a significant difference in the input capacitance and hence the power for different pins. By remapping the inputs so the high activity net is connected to the low power input, the optimization tool can reduce dynamic power.

2.4.3 Resources Sharing Technique

Resource sharing enables EDA tools to generate one hardware component for multiple operations, which typically reduces the hardware required to implement the design. Several resource sharing methods are offered to eliminate the redundant logic circuit. Using automatic resource sharing by the EDA tools is the simplest way to share components and reduce the design area. However, the optimization performance is not the best for the design. Change the order of an operation if does not change the data flow graph (DFG) critical path. Like a differential equation implement [28] shown in Figure 2.19.



(a) (b)

Figure 2.19 A differential equation DFG [28].

Figure 2.19 was used pipeline circuit design. It departed 4 steps to implement the equation. In figure 2.19 (a) requires four multipliers and one adder, one subtract and one comparison, though changing the order of some operation (b) only use two multipliers and one adder, one subtract and one comparison. From this figure, we can know that assign the operations in each time step appropriately is very efficiency for reducing resources usage.

Another resource sharing technique is described in [29]. It presented a special approach to solve combinational feedback loop and unnecessary long critical path because of operation units' resource sharing in RTL (Register Transfer Level) coding. Figure 2.20 described three different resources sharing methods.

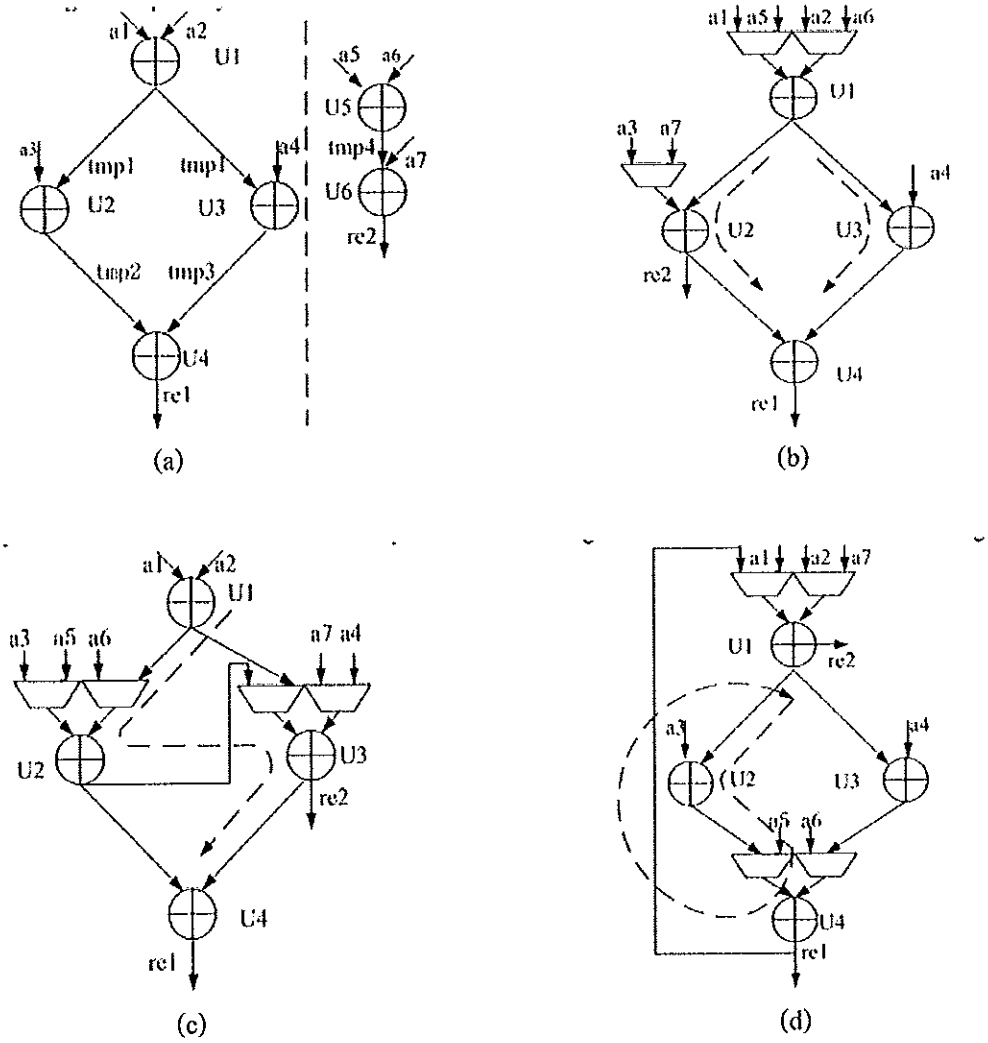


Figure 2.20 Different resources sharing methods implement the same circuit [29].

Figure 2.20 (b) (c) (d) shown the resources sharing in different adder unit. The critical path (b) is equal to the propagation delay of three adders. That is the path through U1-U2-U4 or U1-U3-U4. In (c), but the critical path is the delay of four adders which is shown in broken line, the critical path is only three adders without resource sharing, so it is false path generated by improper resource sharing for the design and should be avoided, and the performance can be improved if the path is removed. The worst case is that a combination loop U1-U2-U4-U1 is caused because of resource sharing like the dashed line in (d). Long runtime of synthesis tool and faulty synthesis result may appear.

2.5 Summary

In this chapter, we firstly introduced the system of hearing aid, and the adaptive noise cancellation architecture. Then we discussed the algorithm for adaptive filter system. We listed several LMS algorithms and analyzed the problem of each algorithm. We will select two LMS algorithm to implement on Matlab and Xilinx ISE. We also intensively gave an overview of the power dissipation source in VLSI circuit design. We listed and discussed some low power design techniques. At the end of this chapter, we introduced the scheduling and resources sharing technique for circuit design in deferent state.

CHAPTER 3

IMPLEMEMT LOW POWER LMS FILTER

3.1 Digital Filter System

A digital processor has performed a digital filter function to calculate the sampled values of the signal. The analog input will first be sampled and transformed into a digital format using an ADC (analog to digital converter). The most operations are multiplication and addition which taking the inputs to multiply with the constant and then add together with the products. If necessary, the results will be converted to analog signals. In a digital filter, a sequence of numbers is used to represent the signal rather than a voltage or current value. Figure 3.1 shows the basic setup of filter a system.

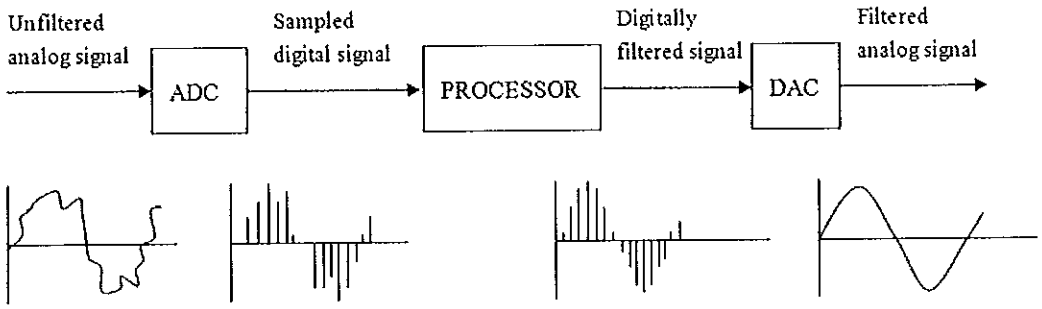


Figure 3.1 Basic setup of a filter system.

3.1.1 Digital Signal Processing (DSP)

The generic architecture of a DSP processor is shown in Figure 3.2. The architecture has two separate memory spaces (coefficient and data) which can be accessed simultaneously. This is similar to the Harvard architecture employed in most of the programmable DSPs. It consists of a hardware multiplier and an adder/subtractor connected to the accumulator so as to be able to execute the multiply-accumulate (MAC) operation efficiently.

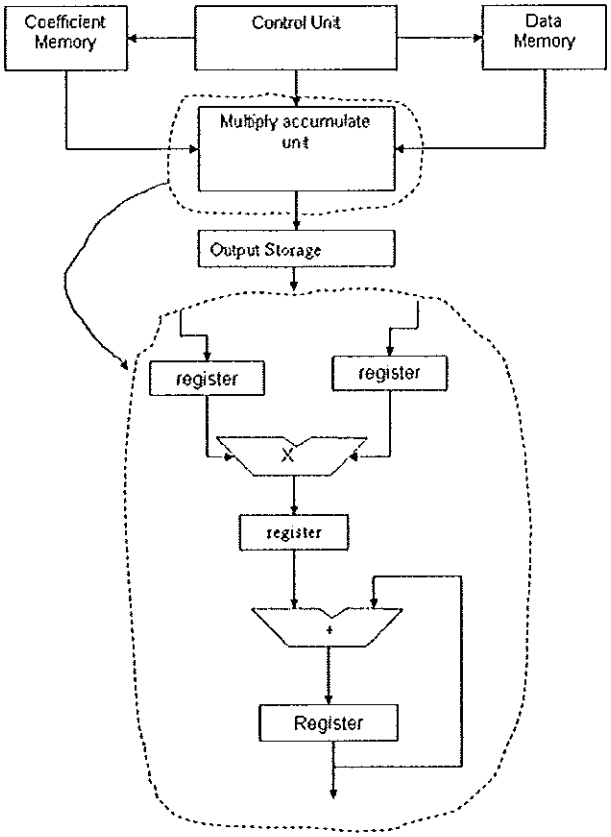


Figure 3.2 A generic DSP processor architecture.

3.1.2 Logic Device Selection

As Programmable Logic Devices (PLD), FPGA is a programmable hardware can easily implement designs with millions of gates on a single chip with wider size limitation which take place in PLD, lead by power consumption and time delay. Designing time and cost can be substantially decreased when compares to the equivalent custom VLSI chips by the easy programmable characterize of the FPGAs, so that reconfigurable system is capable to be developed for executing application at performance. We have chosen standard LMS algorithm and NLMS algorithm for adaptive filter and demonstrated their performance through FPGA implementation.

3.2 Design Statement

The generic architecture of a DSP consists of a multiplier/divider, an adder/subtractor and register. In this dissertation, our selected LMS algorithm will utilize fixed point multiply-accumulate to calculate the FIR block and a divider to update the step size in NLMS algorithm.

3.2.1 Processing of Positive and Negative Number

The data, processed by FPGA, such as the input signals, the coefficients of filter may be positive or negative. Accordingly, it is necessary to employ sign number for expressing the all data inside FPGA. Based on the expression method of the sign number, the highest bit is used as sign bit. In the highest bit, binary digit "0" denotes positive number, binary digit "1" expresses negative number, and all data are denoted by the complement.

3.2.2 Fixed-point Representation

As for digital computation of FPGA, number system influences on the performance of each algorithm. For designing the multiplier and adder inside FPGA, we employ the sign-number multiplier with 8-bits and select fixed-point number computation method. As we know of LMS algorithm equations (2.1 to 2.3) in chapter 2, the correlative fixed-point data is described as Table 3.1 and 3.2:

Table 3.1 Data format of U, d, W and μ

7	6	5	4	3	2	1	0
Sign	Integer		Decimal fraction				

Table 3.2 Data format of y and e

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

For the input signals $u(n)$, W and μ, they have a sign bit and two integer bits, so the decimal is located in other five bits. According to the multiplier operation, the output data will be two times

bits compare with the input data. So y(n) is described as table 3.2. And e(n) is the different between y(n) and d(n), so d(n) should be extension to 16 bits.

3.2.3 Multiplier and Divider Design

The shift-add multiplier scheme is the most basic of unsigned integer multiplication algorithms. The operation of multiplication is rather simple in digital electronics. It has its origin from the classical algorithm for the product of two binary numbers. This algorithm uses addition and shift left operations to calculate the product of two numbers. Two examples are presented below in Figure 3.3.

```
10 x 8 = 80                    -6 x 4 =-24
      1010                          1010
      1000                          0100
-------------------------------------------
    0000                        0000
   0000                          0000
  0000                          111010
 1010                           00000
-------------------------------------------
 1010000                    11101000
```

Figure 3.3 Binary multiplication.

The left example shows the multiplication procedure of two unsigned binary digits while the one on the right is for signed multiplication. The first digit is called Multiplicand and the second is called Multiplier. The only difference between signed and unsigned multiplication is that we have to extend the sign bit in the case of signed one, as depicted in the given right example. There are many inter multipliers in FPGA device. We can design a signed multiplication based on the original multiplier units. The signed multiplication design flow graph is shown in figure 3.4. Here, we assume that the MSB represents the sign of digit. In the multiplier design, we operated the MSB (sign bit) and the remainder number, respectively. It uses the pre-computation structure for low power (shown as Figure 2.17) design.

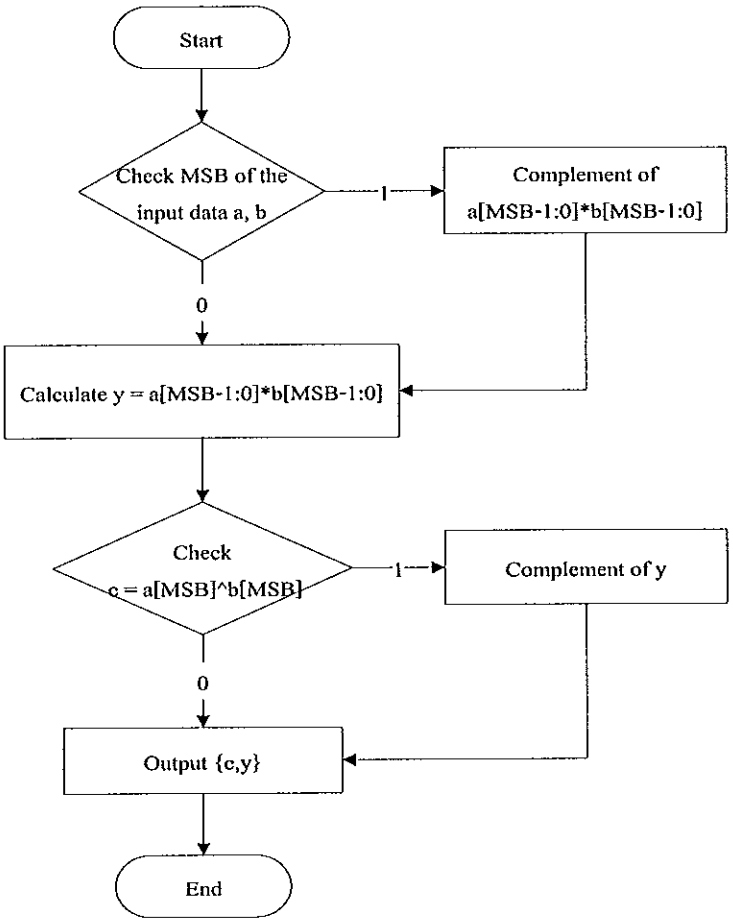


Figure 3.4 Signed multiplication algorithms

In our dissertation, we will implement the NLMS algorithm. The LMS algorithm should utilize a divider to calculate and update the step size μ. We will use the logic core to generate a suitable signed divider.

3.2.4 Bit Extension and Truncation

In this dissertation, we implement an 8-tap adaptive LMS filter. The input/output data and intermediate data are represented like Table 3.1 and 3.2. An 8-bit I/O signal is determined to be a good compromise between the resource costs and filter stability. A bit-extension and truncation method is described in [30] [31]. The single flow graph of LMS filter is described in Figure 3.5.

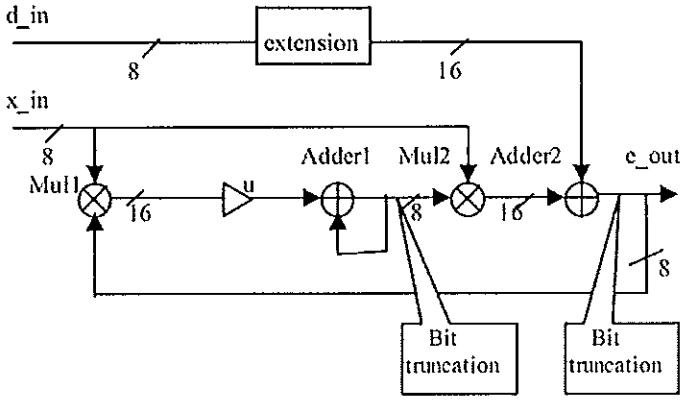


Figure 3.5 Bit-truncation of LMS filter structure [30].

It makes bit-truncation twice in the circuit, each of which is located before the multiplier in saving the resource cost. Apart from that, the location of the first bit-truncation determines the convergence performance of the filter. To meet fast convergence rate and low miss-adjustment error by ANC system, the filter coefficients should be as accurate as possible. Increased word length achieves more accuracy but results in large cost of hardware resource. To make a compromise, it truncates the coefficients to 8-bit right after the adder, before the multiplier. That is to say, in the update process, the coefficients are in 16-bit level, while in the filtering process, they are truncated to 8-bit level. Truncation effect would be serious if the first bit truncation occurs before Adder1, namely the filter coefficients are 8-bit throughout the circuit, and the experimental result of this case is shown in Figure 3.6.

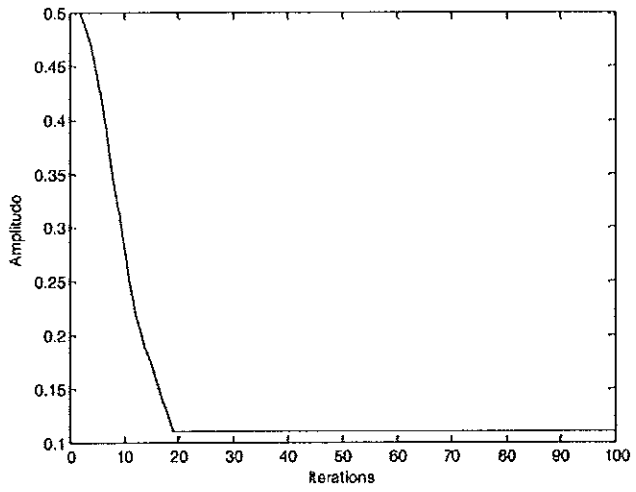


Figure 3.6 The experimental result of bit truncation before adder1 [30].

Figure 3.6 certify that convergence performance can also be affected by bit-truncation effect discussed. The convergence trace of the error signal in the case that the first bit truncation occurs before Adder1 in Figure 3.5 (8-tap filter, $\mu = 0.5$). We can see that the error signal does not converge to zero. So it makes bit-truncation in suitable locations in very important.

3.3 Convergence Rate and Miss-adjustment Error of LMS Algorithm

Tracking ability is also the important property of LMS algorithm. We use a LMS filter (8-tap) with an input signal shown in Figure 3.7, which is a sinusoidal signal corrupted by a random noise signal. Figure 3.7 also shows the tracking ability of the LMS filter, the desired signal and the corrupted signal are plotted, and the red real line represents the output signal with $\mu = 0.05$. We can see that, after about 380 iterations, the result signal is nearly converged to the desired one. The green real line represents the output signal with $\mu = 0.5$. We can know that, after about 50 iterations, the results signal is nearly same the desired signal. However this result line is not smooth. Because of the step size μ is so large, it is influence the stable.

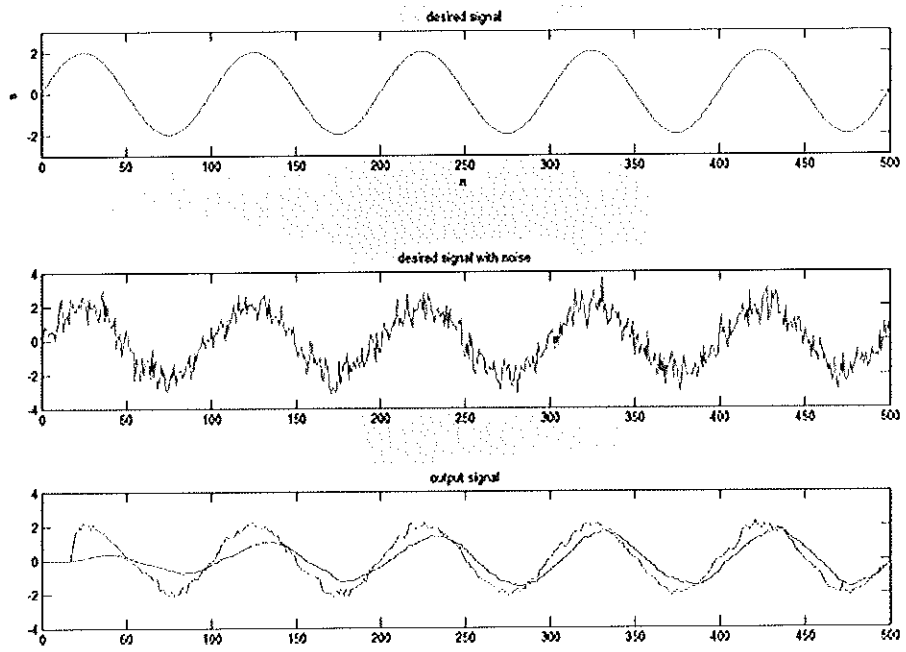


Figure 3.7 Tracking ability of LMS algorithm.

In the chapter 2, we discussed a trade-off between the convergence rate and the state-steady miss-adjustment error of LMS algorithm, which a large μ leads to a high convergence speed with a large state-steady miss-adjustment error; a small μ dominates a low convergence speed with a small miss-adjustment error. We certified this character on MATLAB tool. Figure 3.8 shows the convergence rate with different μ (red line μ = 0.05 and green line μ = 0.5).

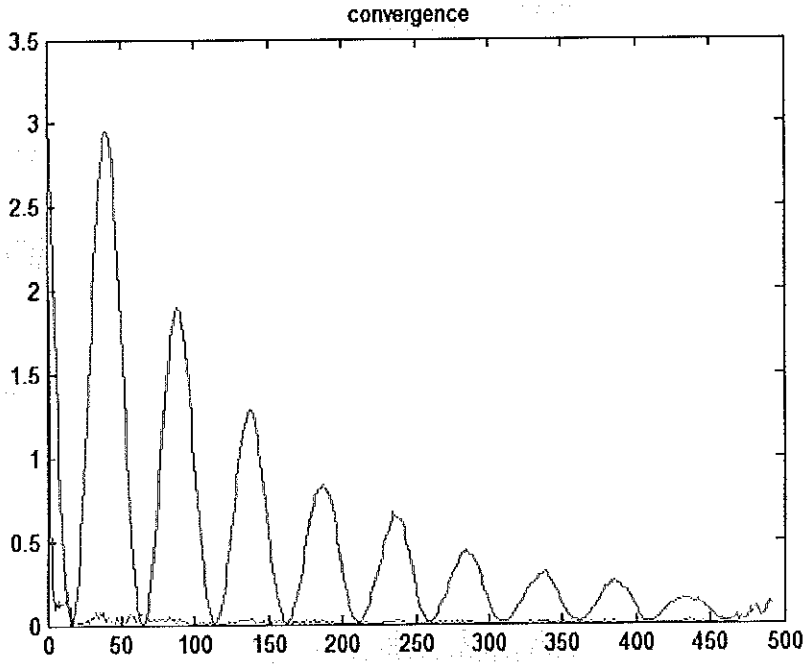


Figure 3.8 The convergence rate of LMS algorithm.

From figure 3.8 we certified relationship between the convergence rate and the step size μ value. A large μ value will lead to a high convergence rate; however a small μ value will lead to a low convergence rate. According to the tracking ability and convergence rate, an appropriate step size μ is important to process a certain signal.

3.4 Fixed Step Size LMS

In previous sections, we explained the concept of LMS algorithm and discussed the influence of the step size μ. According to the discussion, we can choose an appropriate μ for a certain speech signal. As following we will implement a fixed step size μ LMS algorithm using Verilog. And then we changed the data flow graph (DFG) of the traditional LMS circuit to reduce

the logic resources usage and power consumption.

3.4.1 Determine the Step Size μ Based on MATLAB

In this section, we determine an appropriate step size μ for a certain speech signal which is a conversation from a man and a woman. We selected an 8-tap LMS filter and simulated this filter based on Matlab. The simulation structure of LMS algorithm is shown in figure 3.9. According to the adaptive noise cancellation system (Figure 2.3), the primary input signal is the conversation signal with a Gaussian noise signal. The reference input signal is a part of the Gaussian noise signal.

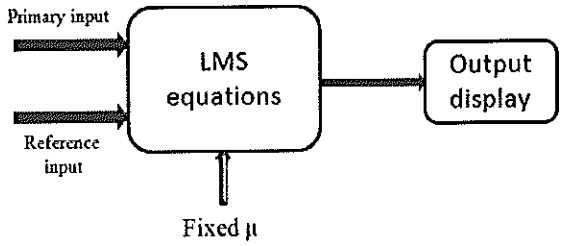
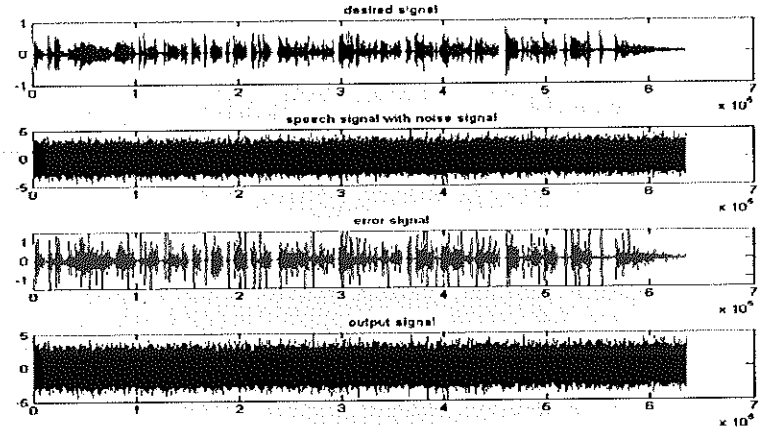
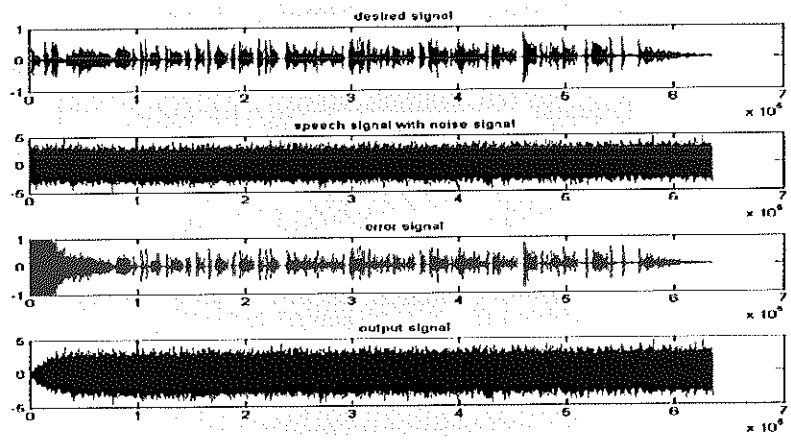


Figure 3.9 The simulation structure of LMS algorithm based on Matlab

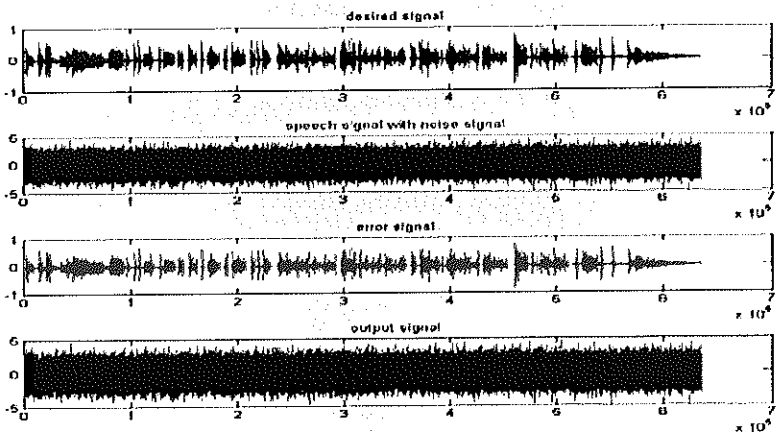
After a lot of simulations, an appropriate step size μ for the certain speech signal is determined. And then we selected three typical results showed in figure 3.10. They are large enough, small enough and an appropriate step size μ.



(a) A larger μ=0.75



(b) A small μ=0.005



(c) An appropriate μ=0.0825

Figure 3.10 Simulation results of different step size μ

Figure 3.10 (a) showed the results with a large enough μ=0.75. Compare with the desired signal and error output signal, we found that the error signal can track the desired signal as soon as possible. But it has a large miss-adjustment error. (b) is the result of a small enough μ value (0.005). From this result we can see that the error output will track the desired signal after $1x10^5$ times. Although the track rate is slowly, the steady able of error output is as well as possible. (c) showed the result with an appropriate μ=0.0825. It has a fast convergence rate with a good output steady able. So we choose μ=0.0825 as the fixed step size.

3.4.2 Original LMS Circuit

According to analyzing the ANC system and LMS algorithm structure, there should design the FIR filter. Two FIR filter structures have been described in [32]. They are direct-form FIR filter (DF) structure and transposed FIR filter (TF) Structure. They are shown in figure 3.11 and figure 3.12.

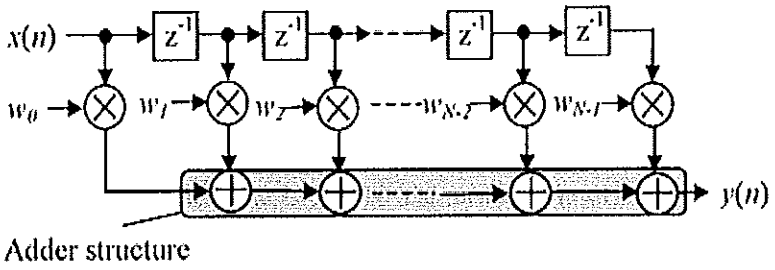


Figure 3.11 Direct-form FIR filter (DF) structure [32]

It reconstructed figure 3.11 like that exchanging the input and output, inverting the direction of signal flow and substituting an adder by a fork, and vice versa. It will get the TF structure (figure 3.12).

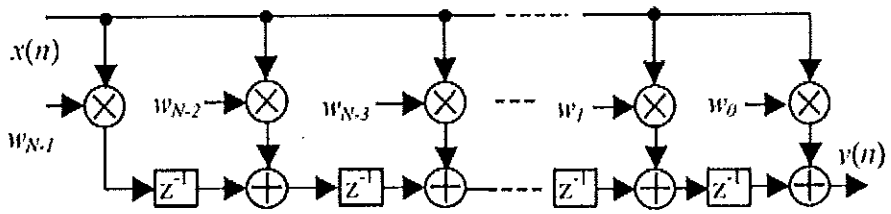


Figure 3.12 Transposed FIR filter (TF) Structure [32]

In general, the preferred choice is the TF FIR filter. Because of this structure is not need an extra shift register for the input data, and is no need for an extra pipeline stage for the adder (tree) of the products to achieve high throughput. The transposed filter enjoys, in the case of a constant coefficient filter. In this design, the LMS filter coefficient should wait and update by the error output. So we still use the DF FIR structure. And it designs the adder structure as binary-tree adder structure (figure 3.13).

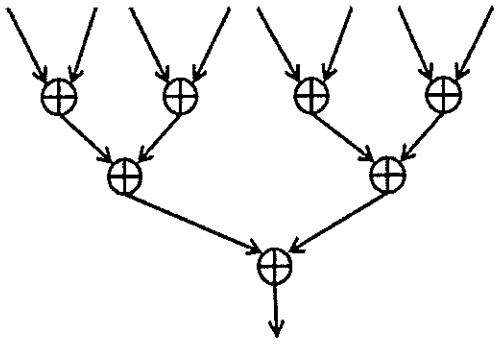


Figure 3.13 Binary-tree adder structure

Translate the fixed step size LMS equation to DFG (data-flow-graph) [36] is shown in figure 3.14. The LMS filter DFG includes FIR filter path and the coefficient update path. The FIR filter block and coefficient use parallel structure design.

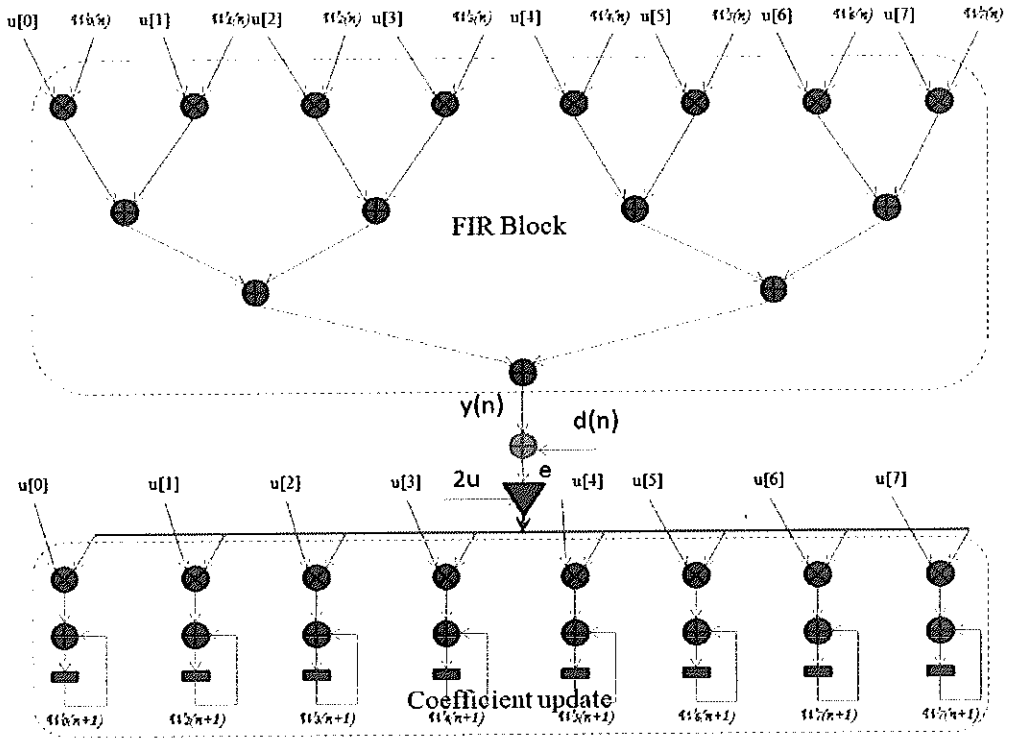


Figure 3.14 The traditional LMS algorithm circuit

It can insert piplining registers in the coefficient update loop or coefficient feed back loop to improve the process speed. It also will increase hardware complexity resulting in an increase in critical path. In addition, it increases output latency which degrades the filter convergence and tracking performance. In our design, it uesd an eight-order LMS filter. It utilized parallel structure. At least, it will consume 16 multipliers and 15 adders. Parallel design can reduce some resources compare with the serial design. In order to reduce more resources and low power consumption we described a new LMS circuit in next part.

3.4.3 Modified LMS Circuit

In order to reduce the logic usage, we proposed a modified circuit of LMS algorithm in this section. It is shown in figure 3.15. Thought analyzed the traditional LMS circuit, the fiter coefficient update compution must wait the error output which is the different of FIR block output y(n) and the input data d(n). The next FIR output y(n+1) cumpution should wait the update coefficient W(n+1). In this feed back loop, it can control the activity of multiplier and adder between the FIR block and the coefficient update bolck.

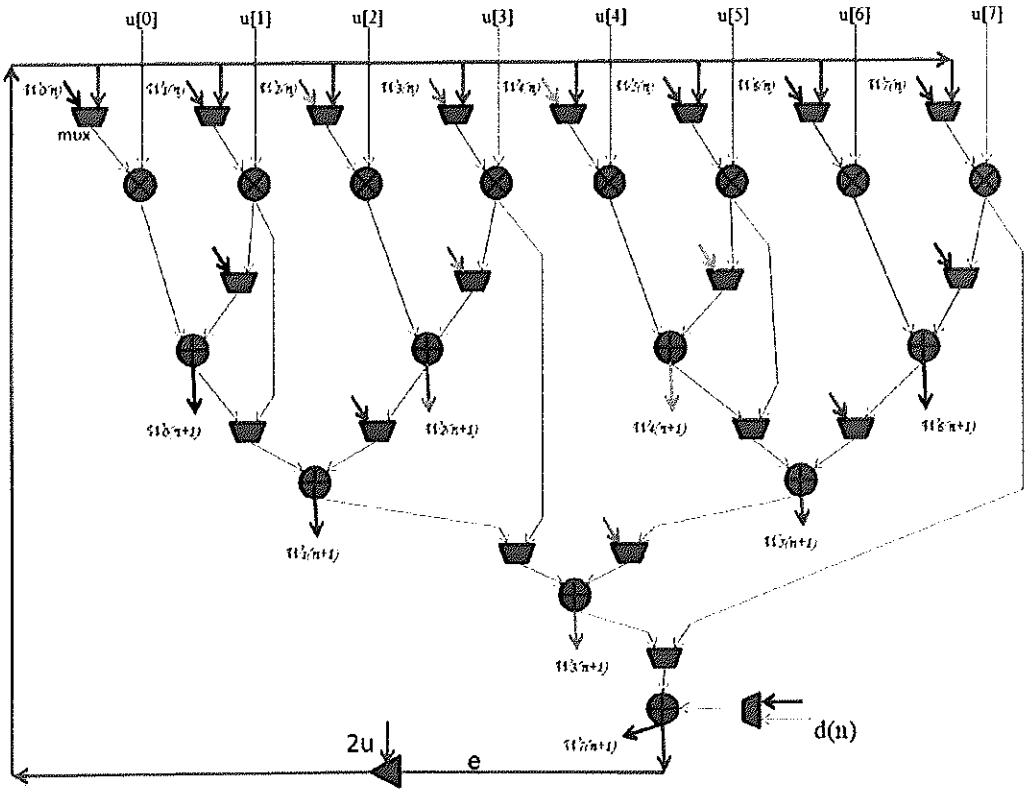


Figure 3.15 The modified LMS circuit

We assume two states 0 and 1, and implement the whole LMS circuit in two clock. The relationship between the clock and states is shown in figure 3.16. In the state (1) it implements the FIR block and computes the error output. In the state (0) it implements the update coefficient block. The input data and the coefficient register will update for next time computation in next state (1). So it implements the FIR block, calculate the output error and update the coefficient in two clock.

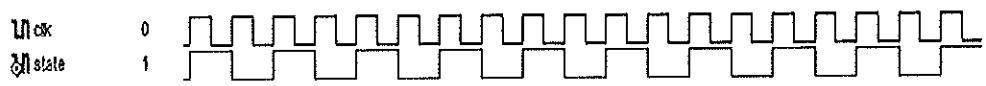


Figure 3.16 Clock and states determine

The detail implementation of figure 3.15 is that how to control the Muxs switch. When the state is 1, the first row Muxs switch to select the coefficient date W, the second row Mux select every output of multipliers as the adders input, the third, fourth and fifth row Muxs

select the output of the adders. The last Mux switch to d(n) as the input of subtraction. In this sate all of the multipliers and adders are used for FIR block and error computation. When the state become 0. All of those Muxs will switch to another side for update the coefficients. Through this work, the FIR bolck and coefficient update block will share the resources in two clocks. In the first clock, all resources use for implement the FIR block and calculate output errors. In the second clock use the same resources update the coefficient for next FIR block calculation. Using this method, it will reduce near half resources compare with the figure 3.14 LMS circuit and it will reduce the power consumption.

3.5 Adjust Step Size LMS Algorithm

The fixed step size LMS filter has a good performance for some certain speech signal. The limited is that it cannot suitable the variable signal as well as possible. In order to get a good performance for variable signal, we introduced an adjust step size LMS algorithm (NLMS). This algorithm can adjust the step size μ by the input data. The adjust formula is shown as following (3.1):

$$u = \frac{0.5}{\sum_{k=0}^{N-1} u_k^2 + 0.25} \qquad (3.1)$$

In the past chapter, we discussed several LMS algorithms. They have benefits and defects respective. The NLMS algorithm can adjust the step size μ for variable input signal, and it is simple to implement.

3.5.1 Implement and Simulate Adjust Step Size LMS Algorithm

As same as the fixed step size μ LMS filter, we simulated the NLMS filter on Matlab platform to certify the feasibility. The simulation structure is shown in figure 3.17.

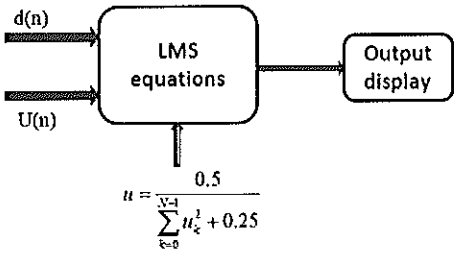
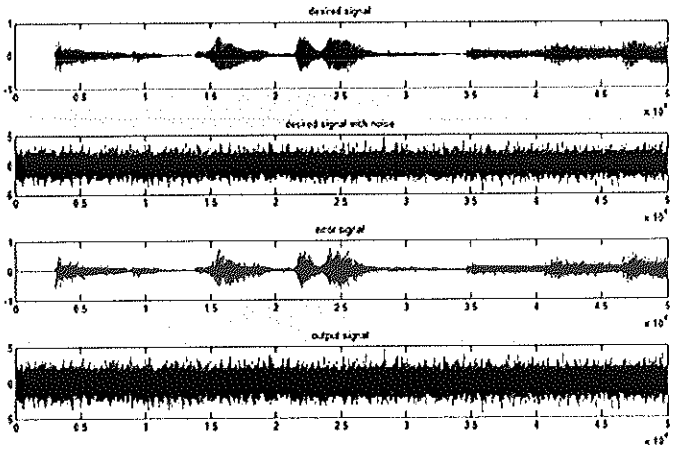
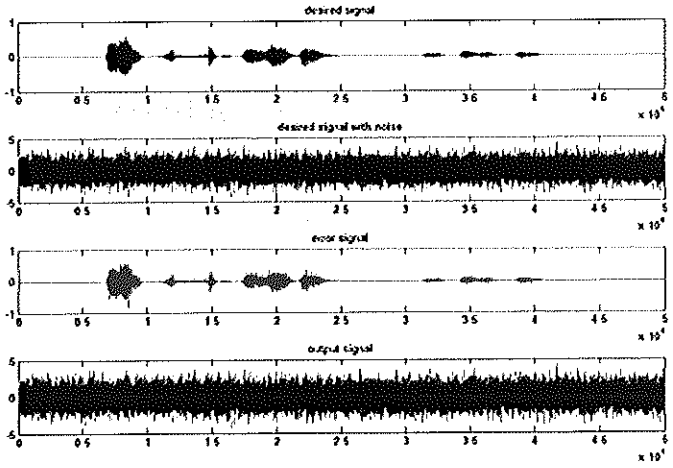


Figure 3.17 The simulation structure of NLMS algorithm based on Matlab

It used equation 3.1 to adjust the step size μ. And the it is certified in two group different signal $U_1(n)$, $d_1(n)$ and $U_2(n)$, $d_2(n)$. The noise signal is Gaussian signal as same as fixed μ LMS filter. The simulate result are shown in figure 3.18.



(a) Input data of $U_1(n)$, $d_1(n)$



(b) Input data of $U_2(n)$, $d_2(n)$

Figure 3.18 Simulate result of different speech signal

Through analyzing error outputs of (a) (b), it can see that the error output have a good tracking ability and the convergence rate is quickly as soon as possible. By synthesizing the results (a) and (b), the NLMS filter has a good performance in deferent signal. As following, we design the adjust μ circuit and simulate on FPGA.

3.5.2 DFG of Adjust Step Size μ Circuit

According to equation 3.1, we can draw the DFG like as figure 3.19. It looks like a FIR filter. It also uses a parallel structure.

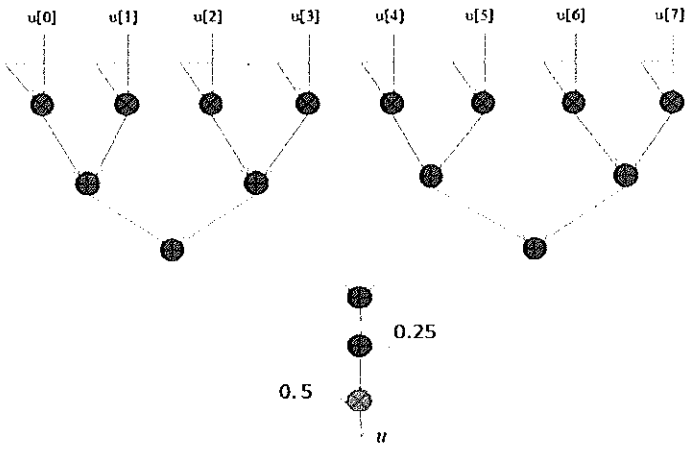


Figure 3.19 The circuit of the step size μ computation

It combines the fixed μ LMS structure (figure 3.14). Using this structure replace the fixed μ. So it will offer a new step size μ for figure 3.14 after every clock. Considering the combination structure, it will consume 24 multipliers, 23 adders and one divider for an 8-tap NLMS filter.

3.5.3 Modified μ Circuit

Considering the logic usage, we analyzed the adjust equation (3.1). There is a regulation in each μ computation. Each μ value is depend on the square sum of the input data u_k. As following it listed several examples.

$$\mu_0 \longrightarrow u_0^2+u_1^2+u_2^2+u_3^2+u_4^2+u_5^2+u_6^2+u_7^2$$
$$\mu_1 \longrightarrow u_1^2+u_2^2+u_3^2+u_4^2+u_5^2+u_6^2+u_7^2+u_8^2$$
$$\mu_2 \longrightarrow u_2^2+u_3^2+u_4^2+u_5^2+u_6^2+u_7^2+u_8^2+u_9^2$$
$$\mu_3 \longrightarrow u_3^2+u_4^2+u_5^2+u_6^2+u_7^2+u_8^2+u_9^2+u_{10}^2$$
$$\mu_4 \longrightarrow u_4^2+u_5^2+u_6^2+u_7^2+u_8^2+u_9^2+u_{10}^2+u_{11}^2$$

From the examples, it is found that each μ computation is just one input data u_k^2 is different between two adjacents μ. The visual regulation is shown in figure 3.20.

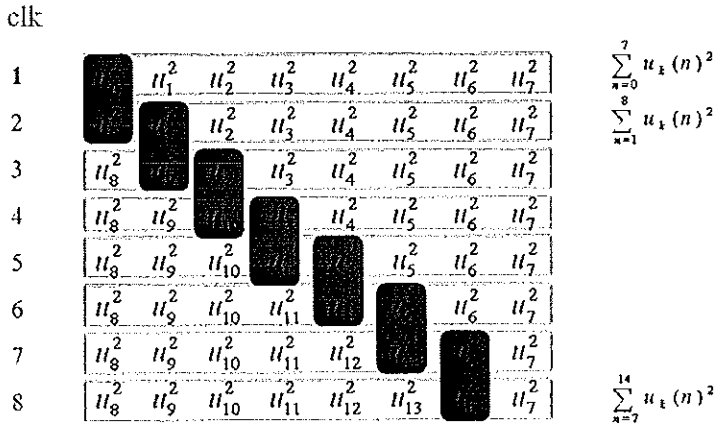


Figure 3.20 The regular of μ computation

According the regular, it will get a new step size μ in every clock. When it computes the first μ, it will do 8 times multiplications. After that, every new μ just do one time multiplication to update the different u_k^2. We redraw the DFG depend on this characteristic. It is shown in figure 3.21.

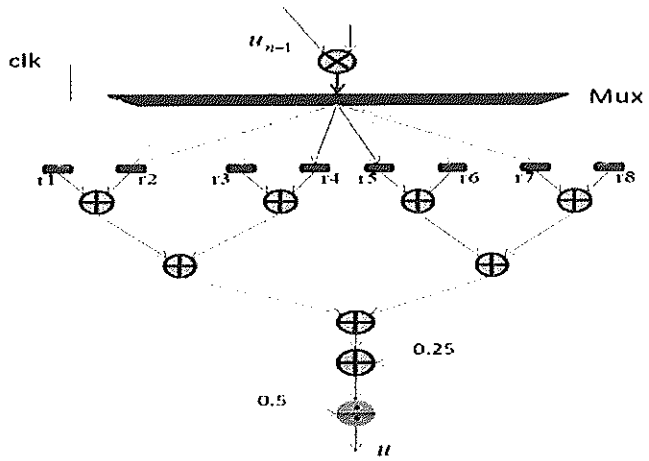


Figure 3.21 Modified circuit of μ compute circuit

It used 8 clocks to computer the first μ value. It used a MUX to control the output of u_k^2. The 1st clock it computes u_0^2 and store the result in register r1. The 2nd clock it computes u_1^2 and store the result in register r2. Repeat the steps, shift the register and update one by one. After 8 clocks it will get the 1st correct value of μ and after this every clock will get one correct μ. From the 8th clock it starts implement the modified LMS circuit as shown in figure.3.15. It completes in every two clocks. For example (figure 3.15), in the 8th clock, the first row MUXs select the coefficient *W*, the second row MUXs select every output of multiplier as the adder input, the third, fourth and fifth row MUXs select the output of the adder. The last MUX choose *d(n)* as the input of adder. After it estimate the output error. In the next clock, all of those MUXs will choose another side for update the coefficients.

Through figure 3.20 works, when computing μ will reduce a lot of multipliers. And structure figure 3.15 shares the resources in every two clocks. Combine the two structures, the total resources usage will be reduced compare with the original circuit. Through the simulation, it also reduced the power consumption.

3.6 Summary

In this chapter, we firstly explained the DSP system and the logic device selection. And then we descried the function design of basic logic, such as adder, multiplier and divider. We also discussed the effect of number representation and the bit extension and

truncation effect of LMS algorithm convergence characteristic. We certified the convergence and tracking ability of LMS algorithm. And then we implemented and simulated the fixed step size LMS algorithms and the adjust step size LMS algorithm on Matlab platform. The simulate results showed that there is a good performance for a certain signal with the appropriate step size. However the fixed step size LMS algorithm can not suit every signal. To solve the problem, we introduced the NLMS algorithm. It adjusts the step size μ by the input data. We also simulate this algorithm on Matlab. It is certified that NLMS can suitable any signal and have a good performance. Finally we implement the two LMS algorithms using Verilog and simulate on Xilinx ISE 10.1. We also introduced the DFG of the two algorithms. To reduce resources usage, we rewrite the DFG depend on (finite state machine) FSM (figure 3.22).

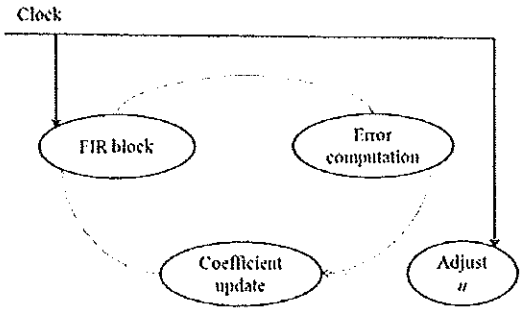


Figure 3.22 FSM of LMS algorithm

The FSM is a feedback loop. Each state should wait the output of the past state. So it can share the resource in each state block. Through analyzed the modified DFG, it reduce a lot logics compared with the original one.

CHAPTER 4

EXPERIMENT RESULTS

We selected two LMS algorithms to implement. First we certified the feasibility of the selected algorithms for the adaptive noise cancellation system based on Matlab platform. And then those designs were coded in Verilog, synthesized and simulated in Xilinx ISE 10.1, and implemented on Virtex chip. In order to show the better performance of the modified LMS filter structure, we also analyzed the power consumption on XPower analyzer. The experimental results demonstrate the proposed architecture achieved more efficient than the original one.

4.1 Convergence Behavior and Tracking Capability

We implemented the LMS algorithms on Matlab. The convergence behavior and tracking capability experimental results of LMS and NLMS algorithm were shown in chapter 3. Figure 3.7 and 3.8 demonstrated convergence behavior and tracking capability effect of the step size μ. It is that a large μ lead to a fast convergence rate with more miss-adjustment error; a small μ lead to slow convergence rate with a small miss-adjustment error. Figure 3.10 and 3.18 showed the feasibility of LMS and NLMS algorithm for speech signal, respectively. A fixed μ LMS filter has a good performance for a certain speech signal. However the NLMS filter is suitable any speech signals.

4.2 Simulation Results of Resources Usage and Power Consumption

In order to compare the main feature between the structure figure 3.14 and 3.15, we used Verilog HDL simulator on Xilinx ISE 10.1. And it selected sc3x400 as the target device. The simulated results is shown in figure 4.1and 4.2, respectively. From figure 4.1 we can see that every clock will get the outputs and update the coefficient. And this structure the maximum frequency is 32.756 MHz. Figure 4.2 showed the modified structure results. It implemented in every two clocks. And the maximum frequency is 43.577 MHz.

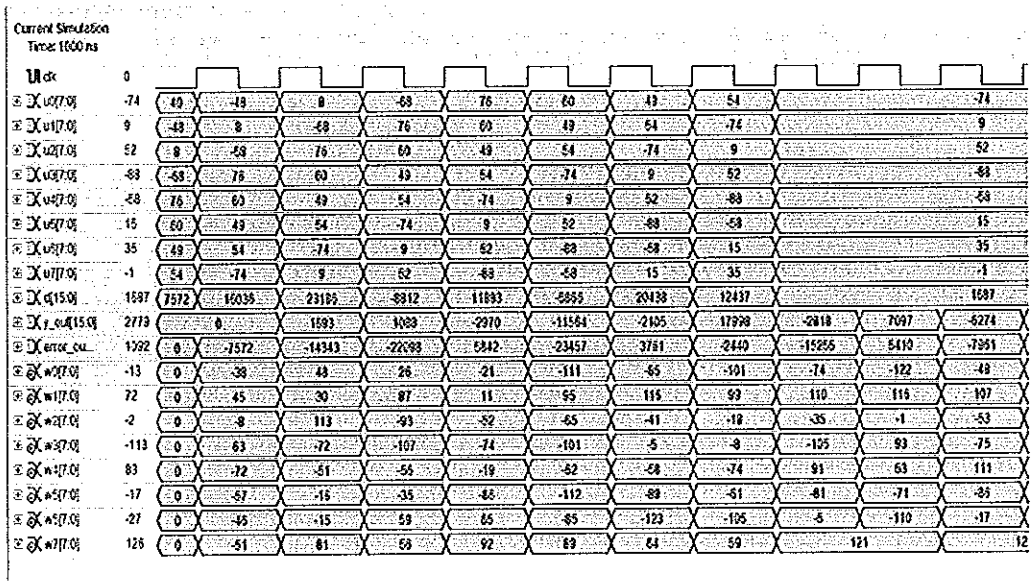


Figure 4.1 Simulate result of structure figure 3.14

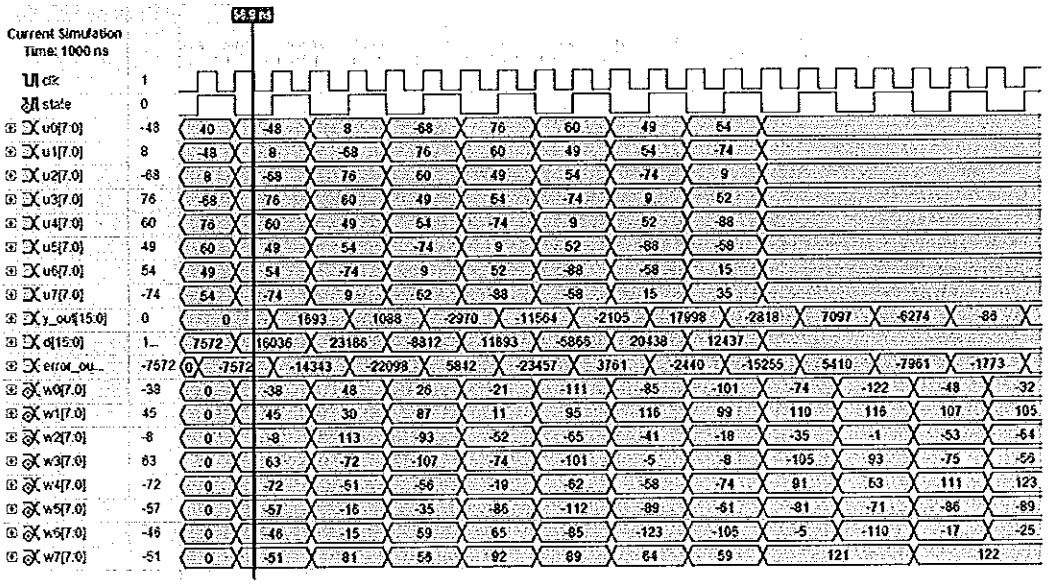


Figure 4.2 Simulate result of structure figure 3.15

Table 4.1 show the device utilization summary of the two circuits. From this table we know the proposed circuit (figure 3.15) use logic resources is less than the original circuit, especially the number of the multipliers. So modified circuit of fixed LMS structure have a better performance than the original circuit.

Table 4.1 Compare the device utilization summary

Target device (sc3x400)		Traditional fixed u LMS circuit (figure 3.14)		Modified fixed u LMS circuit (figure 3.15)	
Logic utilization	Available	Used	Usage	Used	Usage
No. of slices	3884	462	12%	394	10%
No. of flip flops	7168	152	2%	176	2%
No. of input LUTs	7168	838	11%	731	10%
No. of bonded IOBs	173	113	65%	113	65%
No. of Mults (18 x 18)	16	16	100%	8	50%
No. of Gclks	8	1	12%	1	12%

To analyze the power consumption of the two structure, we used XPower analyzer analyze the power consumption. We set the clock frequency is 30 MHz. And the result is shown in table 4.2. The experimental results demonstrated that the dynamic power of the proposed fixed LMS circuit is less than the traditional one. And the total power just 0.060(w) is 57% of the traditional circuit. It economizes 43% power consumption compare with the traditional one.

Table 4.2 Compare with the power consumption

	Traditional fixed u LMS circuit	Modified fixed u LMS circuit
Quiescent Power	0.056(w)	0.056(w)
Dynamic Power	0.049(w)	0.004(w)
Total Power	0.105(w)	0.060(w)

In order to get a good convergence behavior and tracking capability for any input signal, we also introduced a adjust step size LMS (NLMS) algorithm. And we modified the DFG of the adjust μ formula. We coded the original and modified NLMS circuit in Verilog. And then we simulated on Xilinx ISE 10.1 based on Xilinx sc3x1000 device. The resources usage results are shown in Table 4.3. It indicated that the modified NLMS circuit resources usage is less than

the original one.

Table 4.3 The Device Utilization Summary

sc3x1000		Original NLMS circuit (figure 3.14 and 3.19)		Modified NLMS circuit (figure 3.15 and 3.21)	
Logic utilization	Available	Used	Usage	Used	Usage
No. of slices	7680	837	11%	662	9%
No. of flip flops	15360	593	4%	576	4%
No. of input LUTs	15360	1216	8%	841	6%
No. of bonded IOBs	173	173	100%	130	75%
No. of Mults (18 x 18)	24	24	100%	9	38%
No. of Gclks	8	1	12%	1	12%

We analyzed the power consumption on XPower analyzer. We set the clock frequency is 30 MHz. And the result is shown in table 4.4. And the total power just 0.098(w) is 63.6% of the traditional circuit. It economizes 36.4% power consumption compare with the traditional one.

Table 4.4 The Power Consumption

	Original NLMS circuit	Modified NLMS circuit
Quiescent Power	0.104(w)	0.093(w)
Dynamic Power	0.050(w)	0.005(w)
Total Power	0.154(w)	0.098(w)

4.3 Summary

In this chapter, we listed the experimental results. It proved that the modified circuit performance is better than the original one on resources usage and power consumption.

CHAPTER 5

CONCLUSION

To conclude this dissertation, we have presented a low power adaptive and low resources usage LMS filter for noise cancellation system. We analyzed the noise cancellation algorithm and power consumption technique. The whole system is organized as following (figure 5.1):

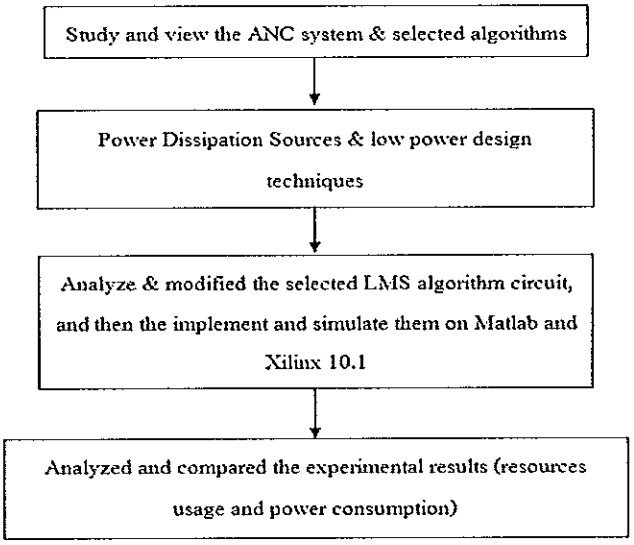


Figure 5.1 Organization of this dissertation

In this work, we selected two types (fixed and adjust μ) of LMS algorithms to implement and certified the feasibility based on Matlab. And then, we implemented and simulated the original/modified circuits on Xilinx ISE 10.1. The experimental results demonstrate that the modified circuits have the better performance compared with the original one. The resources usage of modified LMS circuits is less than the original one, especially multipliers and adders. We also analyzed the power consumption on XPower analyzer. The results show that the modified circuit power consumption is less the original one. The experiment results show that our modified circuit gives the better performance. It can be concluded as

1. Through the simulation of two types of LMS filter on Matlab, it certified that these filters have a good performance for ANC system. Especially the adjust LMS filter is suitable for different speech signal.

2. In order to achieve less resources usage and low power consumption, we use a parallel structure to implement the LMS filters, and use pre-computation structure organized the multiplier design.

3. The modified circuit use logic resources less than the traditional one, especially multipliers and adders. And compared the power consumption at 30 MHz, the modified circuit economized a lot energy.

REFERENCES

[1] Othman O. Khalifa, M. H. Makhtar and M. S. Baharom, "Hearing Aids system for impaired peoples," *International journal of computing & information science, pp.23-26*, APRIL 2004.

[2] Dan Chazan, Y. Medan, and Uzi Shvadron, "Noise Cancellation for Hearing Aids," *IEEE Transactions on acoustics, speech, and signal processing, pp.1697-1705*, NOVEMBER 1988.

[3] Alan V. Oppenheim, Ehud Weinstein, Kambiz C. Zangi, Meir Feder and Dan Gauger, "Single-Sensor Active Noise Cancellation," *IEEE Transactions on speech and audio processing, pp.285,* APRIL 1994.

[4] Upal Mahbub, Celia Shahnaz and Shaikh Anowarul Fattah, "An Adaptive Noise Cancellation Scheme Using Particle Swarm Optimization Algorithm," *ICCCCT 2010, pp.683-686*, OCTOBER 2010.

[5] I.Hasler, O.Buiu, "Design of a Noise Cancellation System for Intelligent Hearing Aids," *Semiconductor Conference, CAS 2004 Proceedings. pp.479,* OCTOBER 2004.

[6] Chia-Feng Juang and Chin-Teng Lin, "Noisy Speech Processing by Recurrently Adaptive Fuzzy Filters," *IEEE Transactions on fuzzy system, pp.139*, FEBRUARY 2001.

[7] B.Widrow et al., "Adaptive noise canceling: Principles and applications," *Proceeding of IEEE, pp. 1692-1716*, DECEMBER 1975.

[8] Simon Haykin, "Adaptive Filter Theory," *Lecture 1-3 Prentice Hall International*, 2002.

[9] Kshab K. Parhi, "VLSI Digital Signal Processing Systems Design and Implementation," university of Minnesota. *pp.7-9*, 1999.

[10] Jafar Ramadhan Mohammed "A New Simple Adaptive Noise Cancellation Scheme Based On ALE and NLMS Filter," *IEEE Transactions on Communication Networks and Services Research, 2007. (CNSR '07), pp.245-252*, MAY 2007.

[11] J. M. Górriz, Javier Ramírez, S. Cruces-Alvarez, Carlos G. Puntonet, "A Novel LMS Algorithm Applied to Adaptive Noise Cancellation" *IEEE signal processing letters, pp.34-37,* JANUARY 2009.

[12] Hirokazu Oba, Minseok Kim and Hiroyuki Arai, "FPGA Implementation of LMS and N-LMS Processor for Adaptive Array Applications," *2006 International Symposium on*

Intelligent Signal Processing and Communication Systems (ISPACS2006), pp.485-488, SEPTEBER 2006.

[13] Shimo-Okubo, Sakura-ku, Saitama, "A Study on Normalized LMS Algorithm Using Rrfined Filtering Technique" *Proceedings of the 5th WSEAS on Signal Processing, pp.264-267,* FEBRUARY 2006.

[14] Hongbing Li, Ningning Tong and Nannan Liu, "A New Variable-Step-Size LMS Adaptive Filtering Algorithm and Its Application in Adaptive Noise Jamming Cancellation System," *8th International Symposium on Antennas, Propagation and EM Theory, 2008. pp.1346,* NOVEMBER 2008.

[15] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. Signal Processing, pp1633-1642,* JULY 1992.

[16] T.Aboulnasr and K.Mayyas, "A robust variable size LMS-type algorithm: analysis and simulation," *IEEE Trans. Signal Processing, pp.631-639,* March 1997.

[17] Joonwan Kim, Alexander D. Poularikas, "Performance Analysis of the Adjusted Step Size NLMS Algorithm," *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory, pp.467-471,* SEPTEMBER 2004.

[18] Joonwan Ken, Chae-Wook Lee and Jae-Kyun Lee, "Comparison of adaptive systems for noise reduction in speech performance," *Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory, 2006. (SSST '06), pp.123-127,* MARCH 2006.

[19] Khosrov DabbaghSadeghipour and Ali Aghagolzadeh, "A New Hardware Efficient, Low Power FIR Digital Filter Implementation Approach," *ICECS-2003, pp.1144,* DECEMBER 2003.

[20] A. T. Erdogan and T.Arslan, "Low Power FIR Filter Implementations Based on Coefficient Ordering Algorithm," *Computer Society Annual Symposium on VLSI Emerging Trends in VLSI Systems Design, pp.226-228,* FEBRUARY 2004.

[21] Michael Keating,David Flynn, Robert Aitken Alan Gibbons and Kaijian Shi, "Low Power Methodology Manual For System-on-Chip Design," chapter 1, 2007.

[22] Julien Lamoureux and Wayne Luk, "An Overview of Low-Power Techniques for Field-Programmable Gate Arrays," *NASA/ESA Conference on Adaptive Hardware and Systems, pp.338-345,* JUNE 2008.

[23] V. Tiwari, S. Malik, and P. Ashar, "Compilation techniques for low energy: an overview," *In Proc. of 1994 IEEE Symp. On Low Power Electronics, pp.38–39,* San Diego, California, USA, OCTOBER 1994,.

[24] Mike Gladden and Indraneel Das, "RTL Low Power Techniques for System-On-Chip Designs".

[25]J. M. Rabaey and M. Pedram, "Low Power Design Methodologies," Kluwer Publishers,1996, ch2-11.

[26] Anantha P. Chandrakasan, Miodrag Potkonjak, Renu Mehra, Jan Rabaey, and Robert W. Brodersen, "Optimizing Power Using Transformations," *IEEE Transactions on computer-aided design of integrated circuit and systems,* VOL. 14, NO. 1, JANUARY 1995.

[27] Anantha P.Chandrakasan, Samuel Sheng and Robert W.Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of solid-state circuits, pp.473-484,* APRIL 1992.

[28] Nattha Jindapeth and Hiroshi Satio,et, "Determination of Worst-case Independent Clock Period for Resource-constrained," *ECTI Conference.* APRIL 2006.

[29] Shushan Qiao, Yong Hei, Bin Wu and Yumei Zhou, "A Technique to Avoid Combination Feedback Loop and Long Critical Path in Resource Sharing," *International Conference on 7th ASIC,2007, pp.1170-1173,* OCTOBER 2007.

[30] Tian Lan and Jinlin Zhang, "FPGA Implementation of an Adaptive Noise Canceller," *2008 International Symposiums on Information Processing, pp.553-558.*SEPTEMBER 2008.

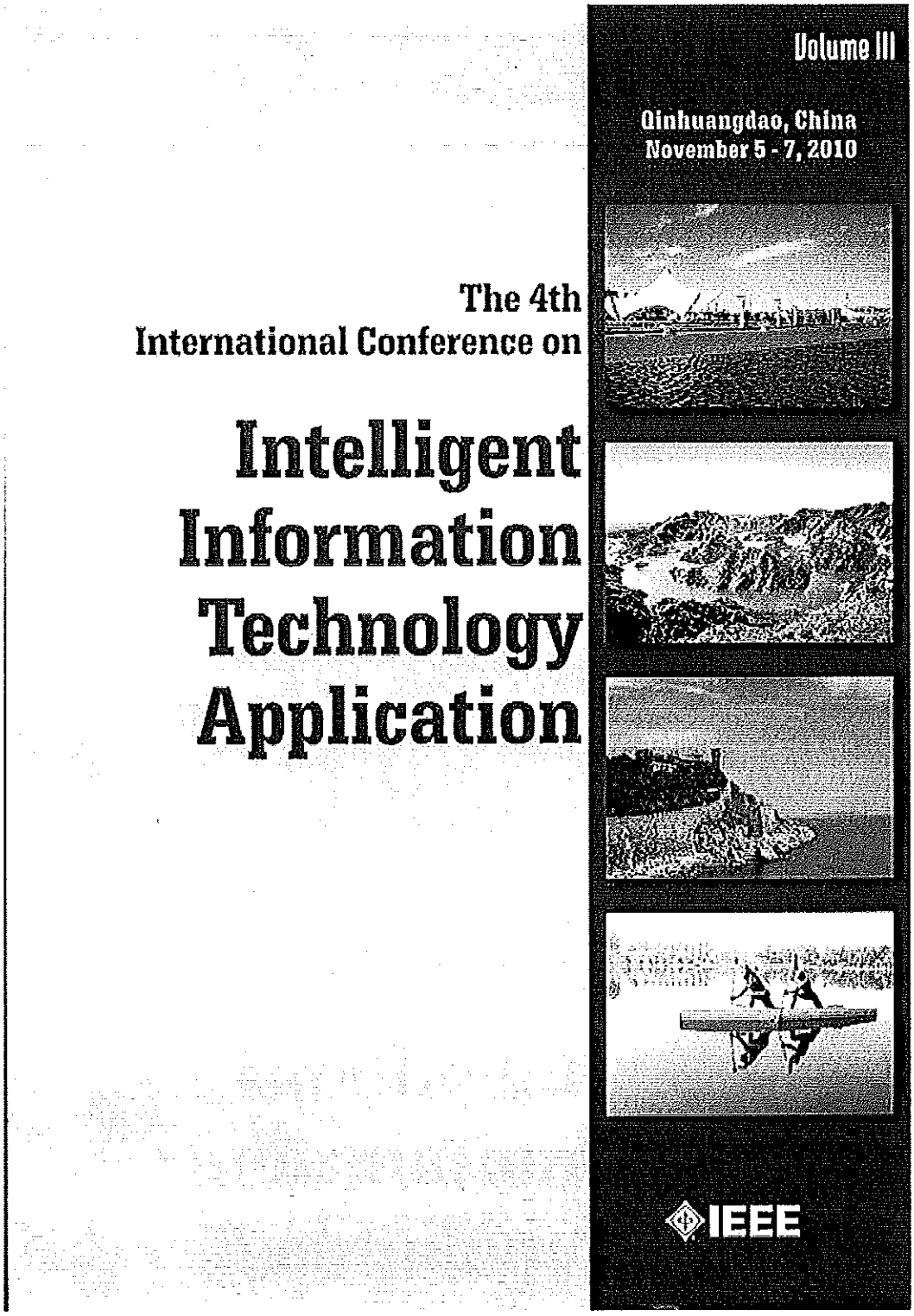
[31] Ahmed Elhossini, Shawki Areibi and Robert Dony, "An FPGA Implementation of the LMS Adaptive Filter for Audio Processing," *Conference on IEEE International Reconfigurable Computing and FPGA's, 2006. pp.1-4,* SEPTEMBER 2006.

[32] Lok-Kee Ting, Roger Woods, "Virtex FPGA Implementation of a Pipelined Adaptive LMS Predictor for Electronic Support Measures Receivers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp.86-95, JANUARY 2005.

APPENDIX

Appendix

Published papers


Volume III
Qinhuangdao, China
November 5 - 7, 2010
The 4th
International Conference on
Intelligent
Information
Technology
Application
IEEE

2010 4th International Conference on Intelligent Information Technology Application

Adaptive LMS Filter for Noise Cancellation Based on FPGA

Wang Shenming Wannarat Suntiamorntut Nattha Jindapetch
1.Computer Engineering, Prince of Songkla University, Songkhla, 90112, Thailand
E-mail: shenming387@163.com, wannarat@coe.psu.ac.th
2. Electrical Engineering, Prince of Songkla University, Songkhla, 90112, Thailand
E-mail: nattha.j@psu.ac.th

***Abstract*—This paper presents a low power design of adaptive LMS (Least-Mean-Square) filter, using an 8bit fixedpoint arithmetic representation, for noise cancellation based on FPGA. The design has been synthesized and simulated in Xilinx ISE. LMS algorithm is widely used in the adaptive filter for noise cancellation system or hearing aid system. We determined the step size parameter u, through simulated the LMS algorithm on MATLAB. After this work, we implemented and simulated the traditional LMS circuit and the proposed LMS circuit in Xilinx ISE. We found that the proposed circuit uses less resource than the traditional circuit. Using the XPower Analyzer tool, the power consumption is reported. The results show that the proposed circuit dissipates less power than the traditional circuit. It nearly economizes 43% power consumption compared with the traditional one.**

Keywords—Adaptive LMS Filter, Low Power Consumption, Noise Cancellation, Hearing Aids, FPGA.

I. INTRODUCTION

In our daily life, we can find many people who are troubled in some types of hearing loss. According to an investigation, approximately 10% of the world's population suffers from the hearing problem [6]. Hearing aid is a small electronic device which helps the hearing impaired person to hear and understand speech clearly. The major processing of hearing aid is noise canceller. It is possible to improve noise cancellation performance by using an adaptive filter to distinguish speech signal from the noise signals. The noise cancellation system of hearing aid system [1] is shown in Fig. 1.

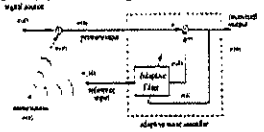


Figure 1. The noise cancellation system

The processing corner is the adaptive filter as shown in Figure1. The adaptive filter is a main component in digital signal processing. It can adjust the coefficient by itself. Normally the adaptive filter uses least-mean-square (LMS) algorithm to perform. The LMS algorithm is very simple and easy to implement on hardware. However, there is a trade-off between the convergence speed and the state-steady-misadjustment error in the standard LMS. So we will determine some correlation parameter of the LMS algorithm based on Matlab in advance.

The hearing aid device should be designed as low power consumption as possible, since the device uses the power source from battery. Due to the cost of ASIC design, the design will be implemented on FPGA fashion. Field programmable gate arrays (FPGAs) have emerged as a DSP implementation platform as they posse highly parallel structures with a high number of arithmetic circuits such as adders and registers which permit high levels of pipelining to be efficiently exploited. The high level synthesis and low power design techniques are applied in this design to achieve a better adaptive LMS filter for the noise cancellation.

The rest of this paper is organized as follows: we briefly introduce the theory of the LMS algorithm and determine the important parameter in Section 2. In Section 3 we intensively analyze and implement the traditional and the proposed circuits of LMS filter. And in Section 4, we describe experimental results which simulated from the ISE platform and XPower Analyzer tool. In this section, we will compare the logic resources usage and power consumption of the two circuit. Finally, we conclude the paper in Section 5.

II. LMS ALGORITHM

The Theory of LMS Algorithm

Least-mean-square (LMS) algorithm is widely used in the adaptive noise cancellation. One reason is its simplicity and robustness to signal statistics. It continuously updates the tap-weight vector by the recently tap-weight values estimation. Fig. 2 shows a 3-tap transversal adaptive filter [2].

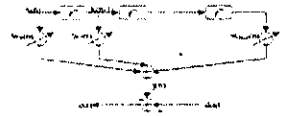


Figure 2. A 3-tap transversal adaptive filter

The fundamental of LMS algorithm architecture was introduced in [2]. It assumes that:

978-1-4244-8666-9/10/$ 26.00 © 2010 IEEE

The input vector matrix

$$X(n) = [x(n), x(n-1), \ldots, x(n-N+1)]$$

The tap-weight vector $W(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]$

The filter output $y(n)$

The output error $e(n)$

The desired signal $d(n)$ includes the desired signal $s(n)$ and other noise signal

The formula of the LMS algorithm is following [2]:

$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i) = W^T(n)X(n)$$

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

$$w_i(n+1) = w_i(n) + 2\mu e(n)x(n-i), \quad i = 0, 1, \ldots, N-1$$

Where $e(n)$ is the error between the desired signal $d(n)$ and the filter output $y(n)$. The filter step size μ is the obtaining optimum performance, the desired signal $d(n)$ should be closed to the noise signal. And then the output error will be close to the desired signal which we want to receive.

LMS algorithm has been implemented and the trade-off between the convergence speed and the state-steady-misadjustment error are indicated to be considered carefully. The step size μ affects directly to both speed and error. A large μ leads to a high convergence speed with a large misadjustment error. A small μ furnishes a low convergence speed with a small misadjustment. The advantages and disadvantages of LMS algorithm are summarized in the Table 1.

Table 1. Some important parameters of LMS algorithm

Convergence speed	A large μ leads to a high convergence speed with a large misadjustment error. A small μ furnishes a low convergence speed with a small misadjustment.
Misadjustment	

Determine The Step Size u

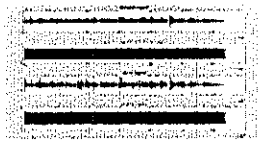
As we understand in the above section, a should determine the step size μ to find the balance between the convergence speed and the misadjustment error. We have simulated the LMS system on Matlab, when μ used different value. The simulation structure and the simulation are shown in Fig. 3 and Table 2.

Figure 3. LMS algorithm simulation structure

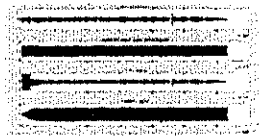
Table 2. Parameter parameters

Desired signal	Noise signal	Number of iterations
A sine wave with a frequency of 1 kHz and amplitude of 1 V	A white Gaussian noise signal with amplitude of 0.5 V	1

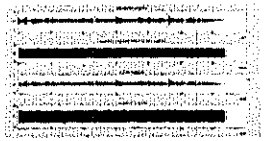
Assuming the hearing aid system shown in Figure3, we assumed we have a speech signal $s(n)$ with a white Gaussian noise signal $v(n)$, and the reference input signal equal to $0.5^*v(n)$. Then through the Matlab simulink and combine with each result. Fig. 4 $d(n)$, $y(n)$, $e(n)$ is shown some comparison results of different μ.



(a) The comparison result of μ equal



(b) The comparison result of μ equal



(c) The comparison result of μ equal

Figure 4. Simulation results with different μ

From the simulation results (a) and (b), we understand that when μ is very large, the convergence speed will very fast but the misadjustment error is also very large. When μ is small enough, the convergence speed is slow with a small state-steady-misadjustment error. From (c) the optimum step size μ has been determined. It may get fast convergence speed with a low misadjustment error. So we choose μ=0.0625 for this design.

III. IMPLEMENT LMS ALGORITHM BASED ON FPGA

A. Fixed-point Representation of Data

As the digital computation of FPGA, number system influences on the performance of such a processor. For

designing the multiplier inside FPGA, we employ the sign-number multiplier with 8bits and fixed-point representation of data. The selected number system is shown in Table 3 and 4.

Table 3 Data format of X and W

8	6	5	4	3	2	1	0
sign	integer			decimal fraction			

Table 4 Data format of d_j and e

8	7	6	5	4	3	2	1	0
sign	integer			decimal fraction				

B. The Original LMS Algorithm Circuit

Translate the LMS equation to DFG (data-flow-graph) [3] is shown in Fig. 5. The LMS filter includes FIR filter path and the coefficient update path.

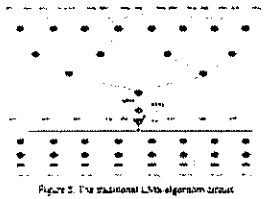


Figure 5. The traditional LMS algorithm circuit

The FIR filter block and coefficient use parallel structure design. For 8 orders LMS filter will consume 16 multipliers and 15 adders. Parallel design can reduce some resources compare with the serial design. In order to reduce more resources and low power consumption we described a new LMS circuit in next section.

C. Proposed LMS Algorithm Circuit

In this section, we proposed a modified circuit of LMS algorithm. It is shown in Fig. 6. We use two clocks to implement the whole LMS algorithm. In the first clock it implements the FIR block and computes the error output. In the second clock it implements the update coefficient block. The clock choice is shown in Fig. 7.

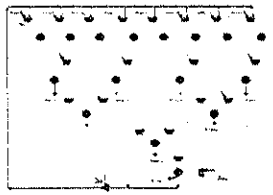


Figure 6. The modified LMS circuit

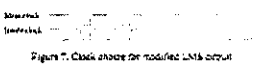


Figure 7. Clock choice for modified LMS circuit

The input data and the coefficient register will update for next time computation in main clock. The main clock is two times of the inside clock. The inside clock controls to implement the FIR block, calculate the output error and update the coefficient.

The detail implementation is when main clock is 0, the first row MUXs select the coefficient W, the second row MUX select every output of multiplier as the adder input, the third, fourth and fifth row MUXs select the output of the second adder. The last MUX choose *data* as the input of adder. After the main clock become 1. All of those MUXs will choose another side for update the coefficients.

Through this work, they share the resources in two inside clocks. In the first inside clock, all resources use for implement the FIR block and calculate output errors. In the second inside use the same resources update the coefficient for next FIR block calculation. Using this method, it will reduce near half resources compare with the past LMS circuit and reduce the power consumption.

D. Low power techniques

Some low power techniques are described in [7]. The total power for a FPGA circuit design consists of dynamic power and static power. Static power consumption is due to the transition leakage. Dynamic power is the power consumed when the device is active. A lot of power reduction methods are proposed in [8], such as clock gating, gate level power optimization, multi-Vdd. In those methods, the clock gating approach [8] [9] [10] is the common way to reduce dynamic power. It turns the clocks off when they do not need to switch. A low power gated-clock finite state machine (FSM) synthesis is proposed in [11]. For LMS algorithm circuit can be modeled as a finite state machine (FSM) which includes several states: FIR block, error computation and coefficient update, as shown in Fig. 8.

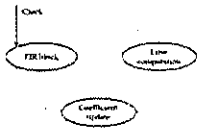


Figure 8. FSM of LMS algorithm circuit

Error computation block should wait FIR block output data. Coefficient update block should wait Error computation block data. The next FIR computation is based on coefficient update. Following the loop, every block must wait the past block output. So every block can share the logic resources each other. The detail is described in part C. We call this method as scheduling & recourses sharing method. We changed the traditional LMS circuit scheduling and share resources. Use this method, it will reduce most logic resources and reduce power consumption. The results are shown in next section.

IV. RESULTS AND DISCUSSIONS

A. Simulation Of The Two Circuit

In order to compare the main feature between these two circuit structures, we used Verilog HDL simulator on Xilinx ISE [4]. And select xc3s400 as the target device. Fig. 9 showed the result of figure 5 structure. From result we can see every clock will get the outputs and update the coefficient. And for this structure the max delay time is 30.592 ns.

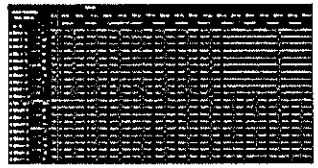


Figure 9. The simulation of traditional LMS circuit

Fig. 10 showed the result of figure 6 structure. From result we can see one clock will get the outputs and next clock update the coefficient. And for this structure the max delay time is 22.048 ns.

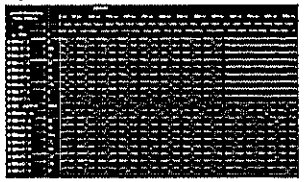


Figure 10. The simulation of proposed LMS circuit

B. Device Utilization Summary

Table 5 show the device utilization summary of the two circuits. From this table we know the proposed circuit use logic resources is less than the traditional circuit, especially the number of the multipliers.

Table 5 Compare the device utilization summary

		Traditional LMS circuit		Proposed LMS circuit	
Logic Utilization	Available	Used	Utilized	Used	Utilized
No. of Slices	3584	462	12%	394	10%
No. of Slice Flips	7168	722	5%	726	5%
No. of 4 input LUTs	7168	839	11%	83	1%
No. of MULT18X18s	16	16	100%	8	50%
IOBs					
No. of IOBs	97	16	16%	8	8%
No. of GCLKs	8	1	12%	1	1%

C. Power Consumption

We use XPower analyzer [5] analyze the power consumption of the two circuit. The result shows in table 6. The result shows that the dynamic power of the proposed LMS circuit is less than the traditional one. And the total power just 0.660(w) is 57% of the traditional circuit. It economizes 43% power consumption compare with the traditional one.

Table 6 Compare with the power consumption

	Traditional LMS circuit	Proposed LMS circuit
Quiescent Power	0.505(w)	0.505(w)
Dynamic Power	0.095(w)	0.055(w)
Total Power	0.600(w)	0.660(w)

V. CONCLUSIONS

In this paper, we presented an adaptive LMS filter for noise cancellation system. We determine the step size u of LMS algorithm on Matlab tool. We described the traditional LMS algorithm circuit and the modified LMS circuit. Compare with the two circuits, the multipliers and adders usage of the modified circuit are less than the traditional.

At last, we showed the results of the two circuits. From the results we know the modified circuit is more performance than the traditional one. Not only logic usage is less than traditional, but also economizes 43% power consumption compare with the traditional one.

REFERENCES

[1] LHASLER,O.SUEL, " Design Of A Noise Cancellation System For Intelligent Hearing Aids " Proc.IEEE 0-7803-8486-7/04, 2004.

[2] Simon Haykin, "Adaptive Filter Theory" Prentice Hall International, 2002.

[3] Lok-Kee Ting, Roger Woods and Colin F.N.Cowan, "Virtex FPGA Implementation of a Pipelined Adaptive LMS Predictor for Electronic Support Measures Receivers" 1063-6339/2005IEEE.

[4] Xilinx ISE simulator, www.xilinx.com.

[5] Xilinx power tools, www.xilinx.com.

[6] Othman O. Khalifa, M. H. Makhtar and M. S. Baharom "Hearing Aids system for impaired peoples" international journal of computing & information science Vol.2,No.1,April 2004.

[7] Julien Lamoureux, Wayne Luk, "An Overview of Low-Power Techniques for Field-Programmable Gate Arrays" 0-7695-3105-3/06 2006 IEEE.

[8] Michael keating, David Flynn, "Low Power Methodology Manual For System-on-Chip Design" chapter 2.

[9] Takahiro Lum, Steve Yang and Wan-Chung Yang, "Logic Synthesis for Low Power using Clock Gating and Rewiring" ISLSVLSI'10,May16-18,2010,Providence,Rhodeisland, USA.

[10] Mike Gladden, "RTL Low Power Techniques for System-On-Chip Designs"

[11] Luca Benini, Giovanni De Micheli, "Automatic synthesis of Low-Power Gated -Clock Finite-State Machines" 0278-0070/00 IEEE.

ECTI-CON 2011

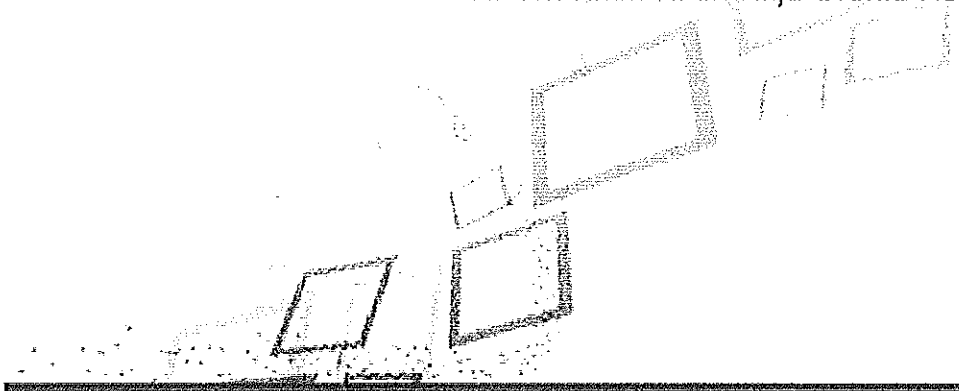
KHON KAEN UNIVERSITY

8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association, Thailand - Conference 2011

Khon Kaen, Thailand

May 17-19, 2011

Pullman Khon Kaen Raja Orchid Hotel



Scheduling & Resources Sharing Technique for Adaptive LMS Filter

Wang Shenming[1], Wannarat Suntiamorntut[2], Nattha Jindapetch[3], Ji Qiufan[4]
Computer Engineering, Prince of Songkla University, Hatyai Songkla Thailand 90112[1,2,3]
Beijing Research Institute of Automation for Machinery Industry, Beijing china[4]
Email: shenming3473@163.com[1], wannarat@coe.psu.ac.th[2], nattha.c@psu.ac.th[3], "76647011@qq.com"[4]

***Abstract*—This paper presents a scheduling & resources sharing technique for adaptive LMS (Least-Mean-Square) filter with low power consumption and low resources utilization. It used an 8-bit fixed-point arithmetic representation for the adaptive filter system based on FPGA. The design has been synthesized and simulated in Xilinx ISE. LMS algorithm is widely used in the adaptive filter for noise cancellation system or hearing aid system. We verified the parameter step size u, through simulated the LMS algorithm on MATLAB. And then, we implemented and simulated the original LMS circuit and the proposed LMS circuit which use the scheduling & resources sharing technique design in Xilinx ISE. We found that the proposed circuit uses resources less than the original circuit. Based on the XPower Analyzer tool, the power consumption is estimated. The results showed that the proposed circuit dissipates less power than the original circuit. It nearly economizes 30.4% power consumption compared with the original one.**

Keywords—Adaptive LMS Filter, Scheduling & Resources Sharing, Low Power Consumption, Noise Cancellation, Hearing Aids, FPGA.

I. INTRODUCTION

In our daily life, we can find many people who are troubled in some types of hearing loss. According to an investigation, approximately 10^{8} of the world's population suffers from the hearing problem [4]. Hearing aid is a small electronic device which helps the hearing impaired person to hear and understand speech clearly. The major processing of hearing aid is noise canceller. It is possible to improve noise cancellation performance by using an adaptive filter to distinguish speech signal from the noise signals. The noise cancellation system of hearing aid system [1] is shown in Fig. 1.

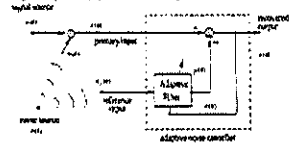


Fig. 1. The noise cancellation system

The adaptive filter is a main component in digital signal processing. It can adjust the coefficient by itself. Normally the adaptive filter uses least-mean-square (LMS) algorithm to perform. The LMS algorithm is very simple and easy to implement on hardware. However, there is a trade-off between the convergence speed and the state-steady-misadjustment error. And it is controlled by the step size. We introduced an appropriate method to balance the trade-off.

The hearing aid device should be designed as low power consumption as possible, since the device uses the power source from battery. Due to the cost of ASIC design, the design will be implemented on FPGA fashion. Field programmable gate arrays (FPGAs) have emerged as a DSP implementation platform as they posse highly parallel structures with a high number of arithmetic circuits such as adders and registers which permit high levels of pipelining to be efficiently exploited. The scheduling & resources sharing technique is applied in this design to achieve a better performance.

The rest of this paper is organized as follows: we briefly introduced the LMS algorithm and verified the method of adjust the parameter step size u in Section 2. In Section 3 we intensively analyze and implement the original and the proposed circuits of LMS filter. And in Section 4, we describe experimental results which simulated from the Xilinx ISE platform and XPower Analyzer tool. In this section, we compared with logic resources usage and power consumption of the two circuits. Finally, we conclude the paper in Section 5.

II. LMS ALGORITHM

A. The Theory of LMS Algorithm

Least-mean-square (LMS) algorithm is widely used in the adaptive noise cancellation. One reason is its simplicity and robustness to signal statistics. It continuously updates the tap-weight vector by the recently tap-weight values estimation. Fig. 2 showed a N-tap transversal adaptive filter [2].

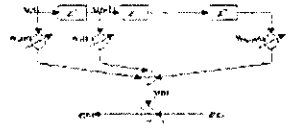


Fig. 2. A basic transversal adaptive filter

The fundamental of LMS algorithm architecture was introduced in [2]. It assumes that:
The input vector matrix: $X(n)=[x(n),x(n-1),...x(n-N-1)]$
The tap weight vector: $W^{T}(n)=[w_0, w_1, ..., w_{N-1}]$
The filter output: $y(n)$
The output error: $e(n)$
The input signal $X(n)$ includes the speech signal and other noise signal.
$d(n)$ is a reference noise signal.
The formula of the LMS algorithm is following:

$$y(n)=\sum_{i=0}^{N} w_i(n)x(n-i)=W^{T}(n)X(n)$$

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

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

Where $e(n)$ is the error between the signal $d(n)$ and the filter output $y(n)$. The filter step size is u. For obtaining optimum performance, the trade-off between the convergence speed and the state-steady-misadjustment errors of LMS algorithm is required to be considered carefully. The step size u affects directly to both speed and error. A large u leads to a high convergence speed with a large misadjustment error. A small u dominates a low convergence speed with a small misadjustment. We discuss the parameter u depend on the input data in next part.

B. Determine The Step Size u

As we mentioned in the past part, it should determine the step size u to find the balance between the convergence speed and the state-steady-misadjustment error. We use the following equation to adjust the parameter u and verified the LMS system on Matlab.

$$u=\frac{0.5}{\sum_{i=1}^{N} X_i^2}=0.25$$

The simulation structure is shown in Fig. 3.

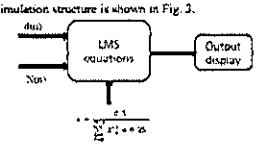


Fig. 3. LMS algorithm simulation structure

In order to verifying the method is suitable every signal, we assumed we have two different speech signal src_1 and src_2. They mixed with a white Gaussian noise signal rnd and the $d(n)$ input signal equate to $0.5*rnd$. Then implemented and simulated the two signals on the Matlab and got the results. As shown in Fig. 4 (a) (b).

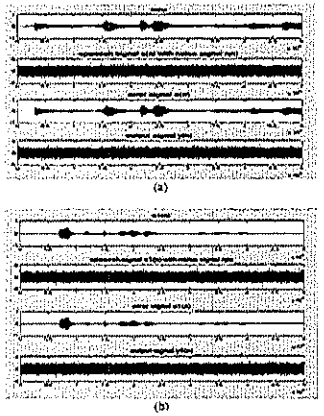


Fig. 4. Simulate result of adjust u LMS algorithm

From the simulation results (a) and (b), we validated that the adjust u LMS algorithm can find a better balance between the convergence speed and the state-steady-misadjustment error. And the system can be suitable any speech signal.

III. IMPLEMENT LMS ALGORITHM BASED ON FPGA

A. Fixed-point Representation of Data

As for digital computation of FPGA, number system influences on the performance of each algorithm. For designing the multiplier inside FPGA, we employ the sign-number multiplier with N-bits and fixed-point representation of data. The selected number system is shown in Table 1 and 2.

Table 1. Data format of X and W

-	6	5	4	3	2	1	0
sign	integer		decimal fraction				

Table 2. Data format of d, y and e

15	14	13	12	9		1	0
sign	integer		decimal fraction				

B. The Original LMS Algorithm Circuit

Translate the LMS equation to DFG (data-flow-graph) [3] is shown in Fig.5. The adjust u equation DFG is shown in Fig.6. The LMS filter includes FIR filter path, the coefficient update path and adjust u path.

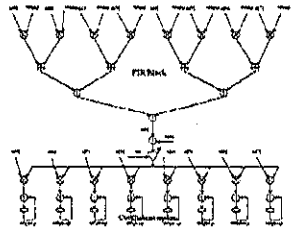


Fig. 5. The original LMS algorithm circuit

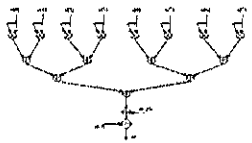


Fig. 6. The adjust u DFG

The FIR filter block and coefficient use parallel structure design. For 8 orders LMS filter will consume 24 multipliers, 23 adders, 1 subtractor and 1 divider. Parallel design can reduce some resources compare with the serial design. In order to reduce more resources and low power consumption we described a modified LMS circuit in next part.

C *Proposed LMS Algorithm Circuit*

In this part, we proposed a modified circuit of LMS algorithm. It is shown in Fig.7 and Fig.9. We use two clocks to implement the FIR and coefficient update block. In the first clock it computes the FIR block output and error output. In the second clock it implements the update coefficient block.

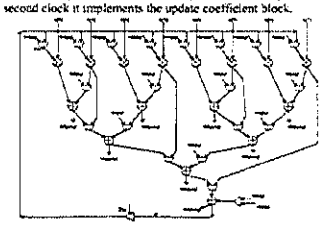


Figure 7. The modified LMS circuit

From the adjust u equation, we got the regulation of each u computation. It is shown in Fig.8. The value of u is depend on the input data X_k. And when compute each u, there just one X_k is different between two adjacent u. So it just use one multiplier can implement the accumulate of x^2. And the modified adjust u DFG is shown in Fig.9.

Fig. 8. The regulation of u computation

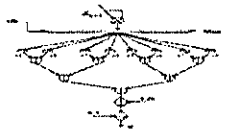


Fig. 9. The modified DFG of adjust u equation

The input data (X) and the coefficient (W) register will update for next time computation every two clock. The clock controls to implement the FIR block, calculate the error output and update the coefficient and the adjust u block.

It used 8 clocks to computer the first u value. It used a Mux to control the output of x^2. The 1st clock it computes x_0^2 and store the result in register r1. The 2nd clock it computes x_1^2 and store the result in register r2. Repeat the steps, shift the register and update one by one. From the 8th clock it will get the 1st correct value of u and after this every clock will get one correct u. From the 8th clock it starts implement the modified LMS circuit as shown in Fig.7. It completes in every two clocks. For example, in the 8th clock, the first row Muxs select the coefficient W, the second row Muxs select every output of multiplier as the adder input, the third, fourth and fifth row Muxs select the output of the adder. The last Mux choose $d(n)$ as the input of adder. After it estimate the output error. In the next clock, all of those Muxs will choose another side for update the coefficients.

Through Fig.9 work, when computing u will reduce a lot of multipliers. Structure Fig.7 shares the resources in every two clocks. All resources use for implement the FIR block and calculate output errors in one clock. In the other clock use the same resources update the coefficient for next FIR block calculation. Using this method, it will reduce a lot of logic resources compare with the original LMS circuit and reduce the power consumption.

D. *Scheduling & resources sharing technique*

Some low power techniques are described in [5]. The total power for a FPGA circuit design consists of dynamic power and static power. Static power consumption is due to the transition leakage. Dynamic power is the power consumed when the device is active. A lot of power reduction methods are proposed in [6], such as clock gating, gate level power optimization, multi-Vdd. In those methods, the clock gating approach [6] [7] [8] is the common way to reduce dynamic power. It turns the clocks off when they do not need to switch. A low power gated-clock finite state machine (FSM) synthesis is proposed in [9]. And a resources sharing technique is described in [10] [11]. For LMS algorithm circuit can be modeled as a finite state machine (FSM) which includes several states, FIR block, error computation and coefficient update, adjust u, as shown in Fig. 10.

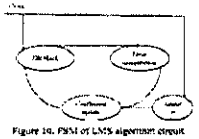


Figure 10. FSM of LMS algorithm circuit

Error computation block should wait FIR block output data. Coefficient update block should wait Error computation block data and u. The next FIR computation is based on coefficient update. Following the loop, every block must wait the past block output. So they can share the logic resources each other in deferent clock. The detail is described in part C. We call this method as scheduling & recourses sharing method. Use this method, it will reduce most logic resources and reduce power consumption. The results are shown in next section.

IV. RESULTS AND DISCUSSIONS

A. *Device Utilization Summary*

In order to compare the main feature between these two circuit structures, we used Verilog HDL simulator on Xilinx ISE. And select xc3x1000 as the target device. The max delay time of the original LMS structure is 32.592 ns. The modified structure the max delay time is 22.048 ns. Table 3 show the device utilization summary of the two circuits.

Table 3.Compare the device utilization summary

		Original LMS circuit		Proposed LMS circuit	
Logic utilization	Available	Used	Usage	Used	Usage
No. of slices	5081	437	12%	665	9%
No. of slip flops	14560	565	4%	476	4%
No. of input LUTs	29120	1236	4%	941	6%
No. of bonded IOBs	270	193	100%	130	50%
No. of Multis	24	24	100%	9	38%
No. of Gclks	8	1	12%	1	12%

From this table we know the proposed circuit use logic resources is less than the original circuit, especially the number of the multipliers.

B. *Power Consumption*

We use XPower analyzer simulate and analyze the power consumption of the two circuits. We set the clock frequency is 30 Mhz. The result is shown in table 4. The result shows that the dynamic power of the proposed LMS circuit is less than the original one. And the total power just 0.098(w) is 63.6% of the original circuit. It economizes 36.4% power consumption compare with the original one.

Table 4.Compare with the power consumption

	Original LMS circuit	Proposed LMS circuit
Quiescent Power	0.108(w)	0.095(w)
Dynamic Power	0.050(w)	0.003(w)
Total Power	0.154(w)	0.098(w)

V. CONCLUSIONS

In this paper, we presented an adaptive LMS filter for noise cancellation system. We verified and determined the step size u of LMS algorithm on Matlab. We described and implemented the original LMS algorithm circuit and the modified LMS circuit. Compare with the two circuits, the multipliers and adders usage of the modified circuit are less than the original one.

At last, we showed the simulation results of the two circuits. From the results we know the modified circuit is more performance than the original one. Not only logic usage is less than original circuit, but also economizes 36.4% power consumption compare with the original one.

REFERENCES

[1] LHASLER,O.BUIL, " Design Of A Noise Cancellation System For Intelligent Hearing Aids " Proc.IEEE 0-7803-8490-7/04, 2004.

[2] Simon Haykin, "Adaptive Filter Theory" Prentice Hall International, 2002.

[3] Lok-Kee Ting, Roger Woods and Colin.F.N.Cowan, "Virtex FPGA Implementation of a Pipelined Adaptive LMS Predictor for Electronic Support Measures Receivers" 1063-8210/ 2005IEEE.

[4] Othman O Khaleifa, M. H. Makhzan and M. S. Rahaman "Hearing Aids system for impaired hearing" *International journal of computing & Information Science*, Vol.2,No.1,April 2004.

[5] Julien Lamoureux, Wayne Luk, "An Overview of Low-Power Techniques for Field-Programmable Gate Arrays" 978-0-7695-3166-2/ 08 2008 IEEE.

[6] Michael Keating, David Flynn, "Low Power Methodology Manual For System-on-Chip Design" chapter 2.

[7] Tak-Kei Lam, Steve Yang and Wai-Chung Tang, "Logic synthesis for Low Power using Clock Gating and Rewiring" GLSVLSI'10,May16-18,2010,Providence,Rhodeland, USA.

[8] Mike Gladden, "RTL Low Power Techniques for System-On-Chip Designs".

[9] Luca Benini, Giovanni De Micheli, "Automatic synthesis of Low-Power Gated-Clock Finite-State Machines" 0278-0070/ 96 IEEE.

[10] Nastha Jindareth and Hiroshi Saito,et "DETERMINATION OF WORST-CASE INDEPENDENT CLOCK PERIOD FOR RESOURCE-CONSTRAITINED SYSTEM" *the 2006 ECTI Conference*.

[11] ShuhanQiao,YongHei, BinWu,YumeiZhou"A Technique to Avoid Combination Feedback Loop and Long Critical Path in Resource Sharing"1-07/ 2007 IEEE.

VITAE

Name Mr. Wang Shenming

Student ID 5110120130

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering	JiangXi University of Science and Technology	2008

List of Publication and Proceedings

[1] Wang Shenming, Wannarat Suntiamorntut, and Nattha Jindapetch, "Adaptive LMS Filter for Noise Cancellation Based on FPGA," *Proceedings of the IITA International Conference (IITA 2010)*, pp. 1-4, Qinhuangdao, China, Nov. 2010.

[2] Wang Shenming, Wannarat Suntiamorntut, and Nattha Jindapetch, "Scheduling & Resources Sharing Technique for LMS Filter," *Proceedings of ECTI-CON 2011*, Khon Kaen, Thailand, May, 2011.