Report Number: WUCSE-2003-35

2003-05-01

Power Consumption of Digital Hearing Aid Computations Using Customized Numerical Representations

Jing Lu

We investigate the impact of numerical representation on the power consumption of digital hearing aids. A fundamental building block, a non-linear amplifier, is implemented using traditional 16-bit linear or customized 9-bit logarithmic and 10-bit floating point numerical representations. An individual channel of a multi-channel hearing aid is constructed, targeting both FPGA and ASIC deployment options. Using signal transition counts in the post-synthesis simulation to model power consumption, we compare the relative power consumption of the non-linear amplifiers, a full hearing aid channel, and the complete hearing aid signal processing for these three numerical representations. Our results show that for... **Read complete abstract on page 2.**

Power Consumption of Digital Hearing Aid Computations Using Customized Numerical Representations

Jing Lu

Complete Abstract:

We investigate the impact of numerical representation on the power consumption of digital hearing aids. A fundamental building block, a non-linear amplifier, is implemented using traditional 16-bit linear or customized 9-bit logarithmic and 10-bit floating point numerical representations. An individual channel of a multi-channel hearing aid is constructed, targeting both FPGA and ASIC deployment options. Using signal transition counts in the post-synthesis simulation to model power consumption, we compare the relative power consumption of the non-linear amplifiers, a full hearing aid channel, and the complete hearing aid signal processing for these three numerical representations. Our results show that for the non-linear amplifier, the logarithmic and floating-point representations provide significant savings over a traditional linear representation. However, since the total power consumption is dominated by the FIR filters, the total power saving is on the order of the filters.

Department of Computer Science & Engineering

Washington University in St. Louis
School of Engineering & Applied Science

2003-35

Power Consumption of Digital Hearing Aid Computations Using Customized Numerical Representations - Master's Thesis, May 2003

Authors: Lu, Jing

April 29, 2003

Abstract: We investigate the impact of numerical representation on the power consumption of digital hearing aids. A fundamental building block, a non-linear amplifier, is implemented using traditional 16-bit linear or customized 9-bit logarithmic and 10-bit floating point numerical representations. An individual channel of a multichannel hearing aid is constructed, targeting both FPGA and ASIC deployment options. Using signal transition counts in the post-synthesis simulation to model power consumption, we compare the relative power consumption of the non-linear amplifiers, a full hearing aid channel, and the complete hearing aid signal processing for these three numerical representations. Our results show that for the non-linear amplifier, the logarithmic and floating-point representations provide significant savings over a traditional linear representation. However, since the total power consumption is dominated by the FIR filters, the total power saving is on the order of the filters

Power Consumption of Digital Hearing Aid Computations Using Customized Numerical Representations

Jing Lu

Jing Lu, "Power Consumption of Digital Hearing Aid Computations Using Customized Numerical Representations," Master's Thesis, Technical Report WUCSE-2003-35, Department of Computer Science and Engineering, Washington University, Saint Louis, MO, 2003.

Computer and Communications Research Center
Washington University
Campus Box 1115
One Brookings Dr.
St. Louis, MO 63130-4899

WASHINGTON UNIVERSITY

SEVER INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

POWER CONSUMPTION OF DIGITAL HEARING AID COMPUTATIONS USING CUSTOMIZED NUMERICAL REPRESENTATIONS

by

Jing Lu

Prepared under the direction of Professor Roger Chamberlain

A thesis presented to the Sever Institute of
Washington University in partial fulfillment
of the requirements for the degree of

Master of Science

May, 2003

Saint Louis, Missouri

WASHINGTON UNIVERSITY
SEVER INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ABSTRACT

POWER CONSUMPTION OF DIGITAL HEARING AID COMPUTATIONS USING CUSTOMIZED NUMERICAL REPRESENTATIONS

by Jing Lu

ADVISOR: Professor Roger Chamberlain

May, 2003
Saint Louis, Missouri

We investigate the impact of numerical representation on the power consumption of digital hearing aids. A fundamental building block, a non-linear amplifier, is implemented using traditional 16-bit linear or customized 9-bit logarithmic and 10-bit floating point numerical representations. An individual channel of a multi-channel hearing aid is constructed, targeting both FPGA and ASIC deployment options. Using signal transition counts in the post-synthesis simulation to model power consumption, we compare the relative power consumption of the non-linear amplifiers, a full hearing aid channel, and the complete hearing aid signal processing for these three numerical representations. Our results show that for the non-linear amplifier, the logarithmic and floating-point representations provide significant savings

over a traditional linear representation. However, since the total power consumption is dominated by the FIR filters, the total power saving is on the order of the filters.

to my husband

Contents

List of Tables

List of Figures

Acknowledgments

My foremost thanks go to my thesis advisor Dr. Roger Chamberlain. Without him, this thesis would not have been possible. I thank him for his patience and encouragement. His insightful suggestions and knowledgeable discussions are always my great help.

I give my gratitude to my former advisor Dr. Robert E. Morley. His visionary thought and rich experiences in circuit design greatly impressed me. This study of logarithmic signal processing also benefited from his experience and advice.

I also thank Dr. John W. Lockwood for his valuable feedback.

I'd like to thank Eric Hemmeter, who shared his knowledge and experience with me and spent time helping me set up some of the experiments.

This research is supported in part by NSF under grant DGE-0138624 through Washington University and by NIH under grant 1R4-3DC04028-02 through BECS Technology, Inc., and Hearing Emulations, LLC. I thank Dr. Julius Goldstein, the Principal Investigator of the NIH grant and President of Hearing Emulations, LLC.

Finally, I thank my husband for always being there with his support and love when I needed him most. I also want to thank my parents and sister in China. Their love is always my strength.

Jing Lu

Washington University in Saint Louis
May 2003

Chapter 1

Introduction

Hearing aids are one of many modern, portable, digital systems requiring power efficient design in order to prolong battery life. Hearing aids perform signal processing functions on audio signals. With the advent of many new signal processing techniques, their requirement for higher computational ability has put additional pressure on power consumption. In this thesis, we are specifically interested in the impact of numerical representation on the power consumption of digital hearing aids. We investigate the use of a traditional linear numerical representation and customized logarithmic and floating-point numerical representations for processing audio signals. Through comparison, we show how the power consumption can be lowered for audio signal processing using customized numerical representations while maintaining the overall signal quality.

1.1 Multichannel Hearing Aid Signal Processing

The need for improved hearing aids is widely attested to by the nationally supported research efforts worldwide. In [30], Sigfrid Soli said:

> Over 28 million Americans have hearing impairments severe enough to cause a communications handicap. While hearing aids are the best means of treatment for the vast majority of these people, only about 5 million of them own hearing aids, and fewer than 2 million aids are sold annually. Market surveys of hearing aid owners have found that only slightly more than half (58%) of these people are satisfied with their aids.

The discussion below is based upon the introduction from [13].

The basic audiological problem existing in current hearing aid designs is the loudness recruitment, or loss of dynamic range [33, 23]. Modern hearing aids automatically compress the range of sound levels into a much smaller range, as needed. Many people agree that the most general and potentially successful design is a multichannel compressive hearing aid that addresses the compression needs of each band of audible frequencies, but sharp disagreement exists whether the dynamic range compression should be instantaneous or slowly adapting [27].

Compressive amplification with automatic gain control of linear amplifiers currently dominates advanced hearing aid design. Extensive research has been conducted in this area [9]. The normal cochlea, a snail-sharped cavity filled with fluid, uses essentially non-linear, rapidly compressive amplification under efferent control [21, 28], whose salient characteristics have been modeled [12] and are currently being explored for use in multichannel hearing aids [14]. Essentially, imitating the important aspects of a healthy ear can provide guidance to the design of future hearing aids. A block diagram of the required signal processing is shown in Figure 1.1 [13].

The input signal comes in from the left, is sampled at a rate of 32 kS/s, and is simultaneously presented to N bandpass filters for separation into distinct channels. The figure shows a 6-channel system, with each channel comprising an octave band, and an overall frequency range from 125 Hz to 8 kHz. Signal amplification in each channel is linear at low sound pressure levels and compressive (power-law) above a threshold. This is shown via a non-linear amplifier in each channel. The output

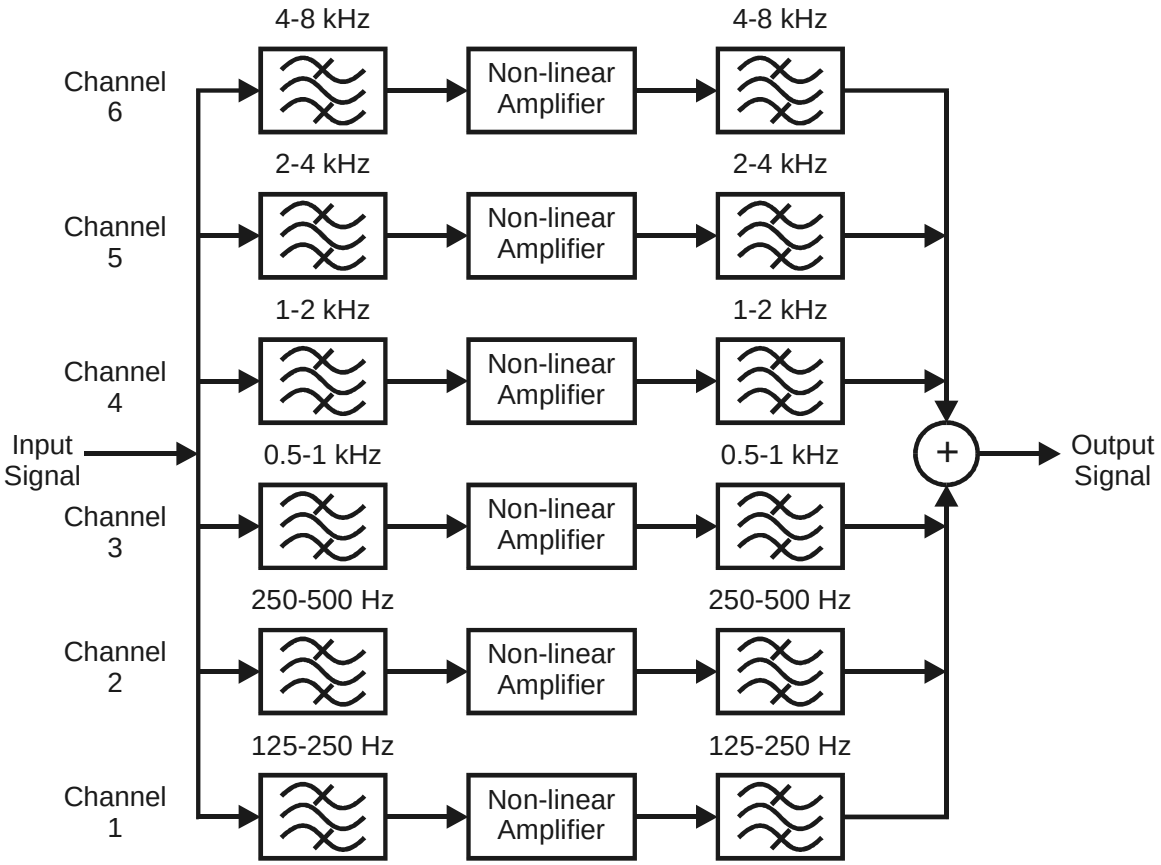

Figure 1.1: Block diagram of multichannel hearing aid signal processing.

of the non-linear amplifier is again bandpass filtered to remove undesired higher-order harmonics introduced by the non-linear amplification. The outputs from the N channels are then summed and presented to the user.

Pilot psychoacoustic experiments with a design simulation that implemented the system of Figure 1.1 demonstrated that both normal and impaired hearing subjects comprehend speech in noise at least as well as with advanced hearing aids [17, 16, 25].

1.2 Contributions

In a digital hearing aid, the resource limitations can be extreme, given that the entire device (including the battery) needs to fit within the ear canal. As a result, power consumption must be held to an absolute minimum. In this thesis, we investigate the power savings associated with constructing the hearing aid using a numerical representation customized to the needs of the application. Specifically, we compare the relative power consumption of three designs, one using a traditional 16-bit linear representation, one using a 9-bit logarithmic representation and the other using a 10-bit floating-point representation. Each design is targeted in two directions, an FPGA implementation and an ASIC implementation. Signal transition counts in the post-synthesis simulation are used to evaluate relative power consumption. For the non-linear amplifier itself in the hearing aid channel, the logarithmic and floating-point representations are shown to provide significant savings over a traditional linear representation (32× and 14× for the FPGA target, and 38× and 8× for the ASIC target). Since in a channel the total power consumption is dominated by the FIR filters, the total power saving is on the order of the filters, which is over 2.5× for the

logarithmic representation vs. the linear representation and 2× for the floating-point representation vs the linear representation.

The author's work in this thesis is listed as follows.

- Implementations of the non-linear amplifier using linear, logarithmic, and floating-point numerical representations in VHDL, including implementations of several Baugh-Wooley multipliers with different bit widths.
- An implementation of a full hearing aid channel in VHDL, with a master controller synchronizing the FIR filters and the non-linear amplifier.
- pre- and post-synthesis simulations along with shell scripts to automate the process of running all the various combinations of the non-linear amplifier, FIR filter, and input set on FPGA and ASIC targets.
- A C++ program for verification of the exponentiation function in the non-linear amplifier using the linear numerical representation. The program performs the same function as the hardware. Its outputs are compared with the hardware outputs. Noise due to errors introduced by polynomial approximation in the hardware implementation is examined.
- C++ programs to generate parameterized Look-Up-Tables (LUTs) in the non-linear amplifier using the floating-point representation.
- A C++ program to parse and sum signal transition counts from ModelSim outputs.

1.3 Related Work

Using reduced bit width or customized numerical representations has been shown to be explored in low power application design. In [26], Okuma et al. presented a technique that reduces the redundant access energy of on-chip data memory by exploring the active data bitwidth of data which is accessed. Their experimental results showed a significant energy reduction compared to the monolithic memory for JPEG and MPEG-2 applications. Junghuwan et al. studied the power minimization problem for data dominated applications based on a novel concept called partially guarded computation [8]. Through dynamically disabling most significant bit computation to remove unnecessary transitions, they reported a 10% to 44% power reduction with reasonable area and delay overhead in functional units.

In [29], Sacha and Irwin compared several techniques (fixed- and floating-point configurations, CORDIC arithmetic, and logarithmic representations) for performing QRDRLS adaptive filtering. The architecture-level power modeling showed that logarithmic arithmetic switched less capacitance and therefore consumed less energy for a given residual error level than the other methods. In [10], Engel et al. recognized the potential usefulness of sign/logarithm encoding in integrated circuit implementation with substantial savings in both area and power. They showed that by choosing an appropriate logarithm base and a sufficiently large number of quantization states the sign/logarithm encoding scheme offers performance very close to that provided by commercial 16-bit codecs. In the paper, they also described the designs of logarithmic digital-to-analog and analog-to digital convertors for use in a digital hearing aid. Gaffar et al. [11] described a method for customizing the representation of floating-point numbers that exploits the flexibility of an FPGA. They used an iterative method to determine the appropriate size of the mantissa and exponent for each operation in a

design which satisfies a given error specification for the output relative to a reference representation.

Sullivan [32] developed a method to estimate the power consumption for VLSI DSP designs. He examined variable parts including multiplier, memory, bus, controller, and interconnects, and showed that a Baugh-Wooley multiplier is more power efficient than comparable shift and add multipliers. He also compared systems using a linear representation with those using a logarithmic representation, showing that the logarithmic representation uses only 30% of the power necessary for the linear representation.

In [5, 6], Chamberlain et al. compared the power consumption of a 16-bit linear representation with several different floating-point representations (4- to 6-bit exponent and 4- to 6-bit mantissa) and a 9-bit logarithmic notation. For each representation, they designed a hardware MAC unit in the VHDL language and performed a standard-cell synthesis, layout, and place-and-route targeting the AMI Semiconductor 0.5 micron VLSI integrated circuit process. The resulting design was simulated using the Mentor Graphics MACH-PA power analysis tool, with input vectors modeling a 21-tap finite impulse response band-pass filter. Their results showed a significant power savings (greater than 5x) using both the floating-point representations and the logarithmic representation. In their later work [7, 18], Hemmeter et al. compared the results of their earlier investigation with a logic-level simulation that models the system in a discrete-event fashion, showing that the signal transition counts in the simulation have a linear relationship with the power consumption. They observed that the increased execution speed of the logic-level simulation makes it possible to investigate a much wider design space, with acceptable inherent inaccuracies due to the discrete model, early in the design cycle.

We extend the above work to include the implementation of a non-linear amplifier using different numerical representations. Using the same discrete model, we investigate the relative power consumption of the non-linear amplifier, a full hearing aid channel, and a complete hearing aid signal processing.

1.4 Outline

This thesis is organized as follows. In chapter 2, we introduce the three numerical representations (16-bit linear, 9-bit logarithmic, and 10-bit floating-point representations) and examine their qualifications for audio signal processing. Then we describe the functionality of the non-linear amplifier and present the detailed implementation of the non-linear amplifier using the three numerical representations. In chapter 3, we give a review of the implementation of the FIR filter, another fundamental component in the hearing aid, developed by E. Hemmeter [18]. Then the construction of a full hearing aid channel is described. Chapter 4 presents the discrete event simulation model for power estimation. We compare the relative power consumption of the non-linear amplifiers, a full channel, and the complete hearing aid signal processing for the three numerical representations. Finally, we conclude in Chapter 5 with a summary and a discussion of future work.

Chapter 2

Implementation of a Non-linear Amplifier

2.1 Multichannel Hearing Aid Architecture

To ease the computational burden, the real-time implementation of the hearing aid utilizes a multirate design for the signal processing, which is illustrated in Figure 2.1 [13]. The input signal comes in on the upper left side of the figure, is sampled at a rate of 32 kS/s, and is delivered to an allpass filter in channel 6 (for equalization of the group delay across the channels) and to a lowpass filter and downsampler (so that signals in channels 5 through 1 have sampling rates that are successively halved). The bandpass filters and non-linear amplifiers in the center of the diagram are the same as before, only the frequency of execution is diminished for lower frequency channels. The output of the second bandpass filter in each channel is first added to the output from any lower frequency channels, upsampled, and lowpass filtered.

In the prototype implementation, the bandpass filters are 21-tap FIR filters designed by windowing and sampling IIR Butterworth bandpass impulse responses

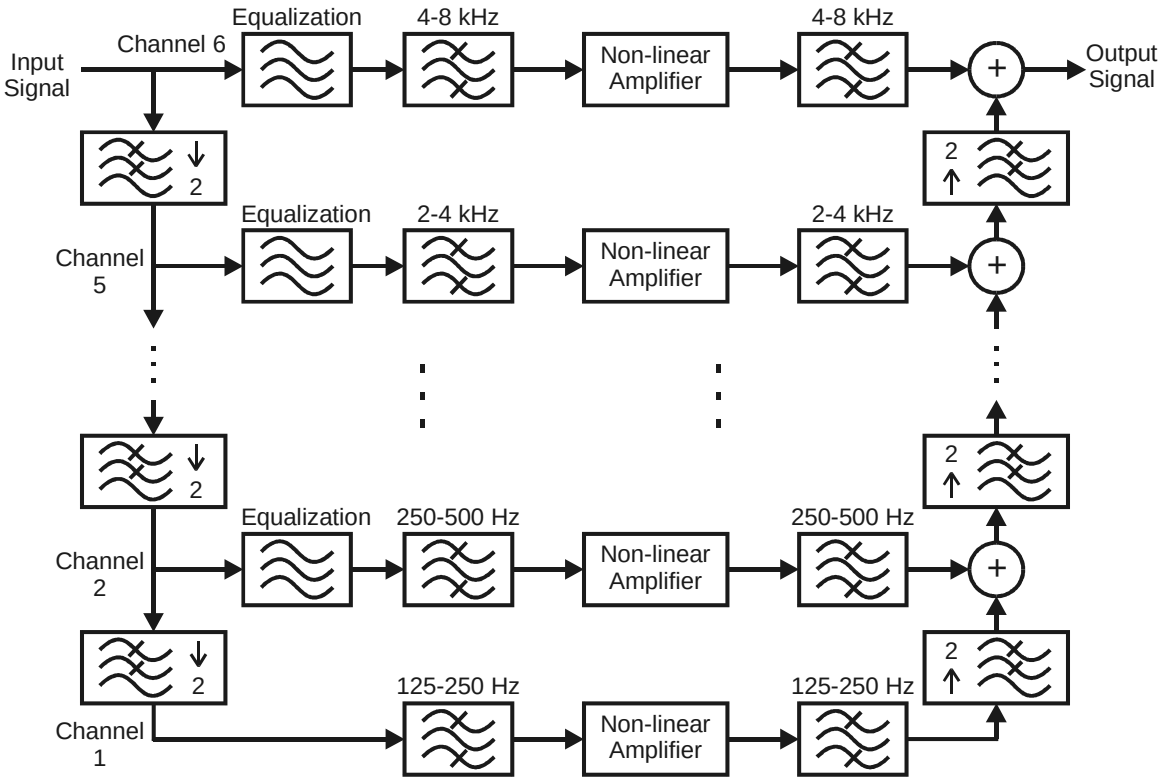

Figure 2.1: Multirate signal processing flow diagram for the digital hearing aid.

and are identical for all channels. The allpass equalization filters are simply circular delay buffers. Since this function can be combined into the filter that follows it, its power is not explicitly modeled. The lowpass filters are 21-tap FIR filters with a normalized cutoff frequency of 0.3π, also identical for all channels. Additional details are available in [13].

2.2 Numerical Representations

Focusing our attention on audio signals that communicate human speech, a dynamic range of approximately 100 dB and a signal-to-quantization-noise ratio (SQNR) of approximately 30 dB have been shown to be adequate [31]. In this investigation, we compare the power consumption of a 16-bit linear representation (in two's complement Q0.15 format), a 9-bit sign-magnitude logarithmic representation (using base 0.941 logarithms [22]) and a 10-bit floating point representation (with one sign bit, 4 exponent bits, and 5 mantissa bits). The dynamic range can be expressed as follows:

$$Dynamic\ Range\ (dB) = 20\log_{10}\left(\frac{x_{max}}{x_{min}}\right)$$

where x_{max} corresponds to the largest representable value and x_{min} corresponds to the smallest non-zero representable value.

SQNR is quantified as follows:

$$SQNR = 20\log_{10}\left(\frac{\frac{x_i}{\sqrt{2}}}{\frac{|x_i - x_{i+1}|}{\sqrt{12}}}\right)$$

where x_i and x_{i+1} represent the ith and $(i+1)$th value present in the number representation. The above expression makes the assumption that the input signal is a sinusoid

with peak value x_i and the quantization error is uniformly distributed between x_i and x_{i+1}.

2.2.1 16-bit Linear Representation

A number (x) in a 16-bit linear representation (in two's complement Q0.15 format) has a value in the range of -1 ($x = 1000000000000000_2$) to $+(1 - 2^{-15})$ ($x = 0111111111111111_2$), with a dynamic range of 90.3 dB. The value of x can be computed using the following formula.

$$x = \sum_{i=1}^{15} a_i 2^{-i} - a_0$$

where a_0 is the MSB and a_{15} is the LSB.

Figure 2.2 shows the range of representable values and the SQNR for those values in the 16-bit linear representation. The linear representation has an SQNR that ranges from near 0 dB, well below the 30 dB we desire, up to almost 100 dB, which is much more than necessary.

2.2.2 9-bit Logarithmic Representation

A number (x_l) in a 9-bit sign-magnitude logarithmic representation (using base 0.941 logarithms [22]) has a value in magnitude ranging from 1.84×10^{-7} ($x_l = 011111111_2$ or 111111111_2) to 1 ($x_l = 000000000_2$ or 100000000_2). Its dynamic range is 134.7 dB. The value of x_l can be computed as follows.

$$x_l = (-1)^{a_0} \times \sum_{i=1}^{8} a_i 2^{(8-i)}$$

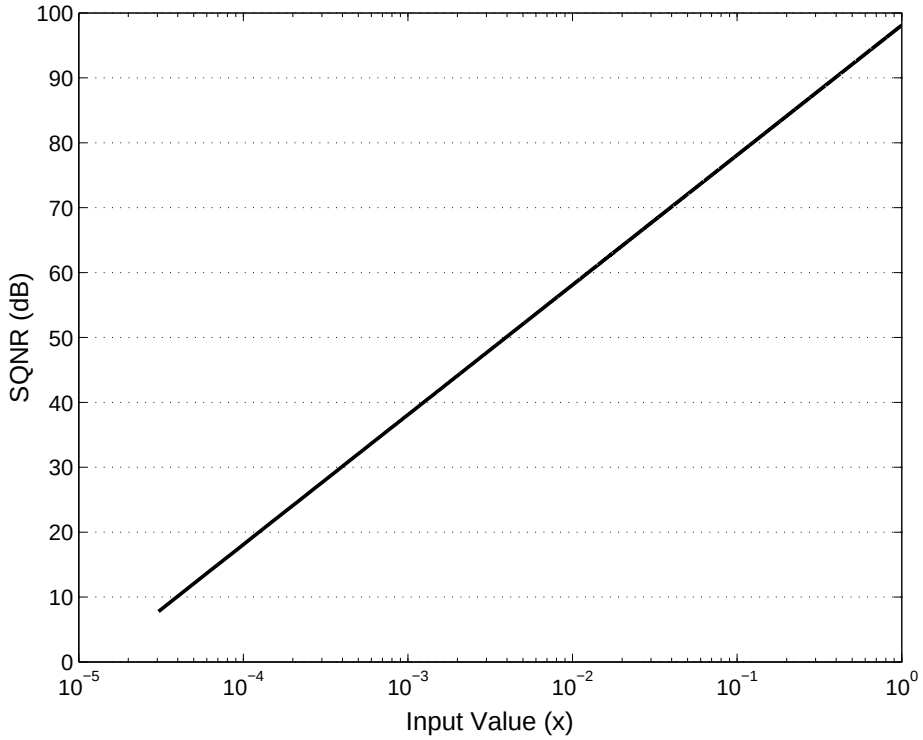

Figure 2.2: SQNR for 16-bit linear representation.

where a_0 is the MSB (also is the sign bit) and a_8 is the LSB. x_l denotes the logarithmic value of x, which is

$$x = sng(x_l) \times (0.941)^{(|x_l|)}$$

Figure 2.3 shows the range of representable values and the SQNR for those values in the 9-bit logarithmic representation. The logarithmic representation has a flat SQNR just over 30 dB for most of the representable values. In the dead zone near zero where the quantization noise is greater than the representable value, the SQNR drops sharply. In reality, due to the limitations of the A/D converters the dead zone can be much bigger than what is shown in this figure. More information can be found in [22]. The following formula quantifies the SQNR for the 9-bit logarithmic representation.

$$SQNR = 20 \log_{10} \left(\frac{\frac{x_i}{\sqrt{2}}}{max(\frac{|x_i - x_{i+1}|}{\sqrt{12}}, x_{min})} \right)$$

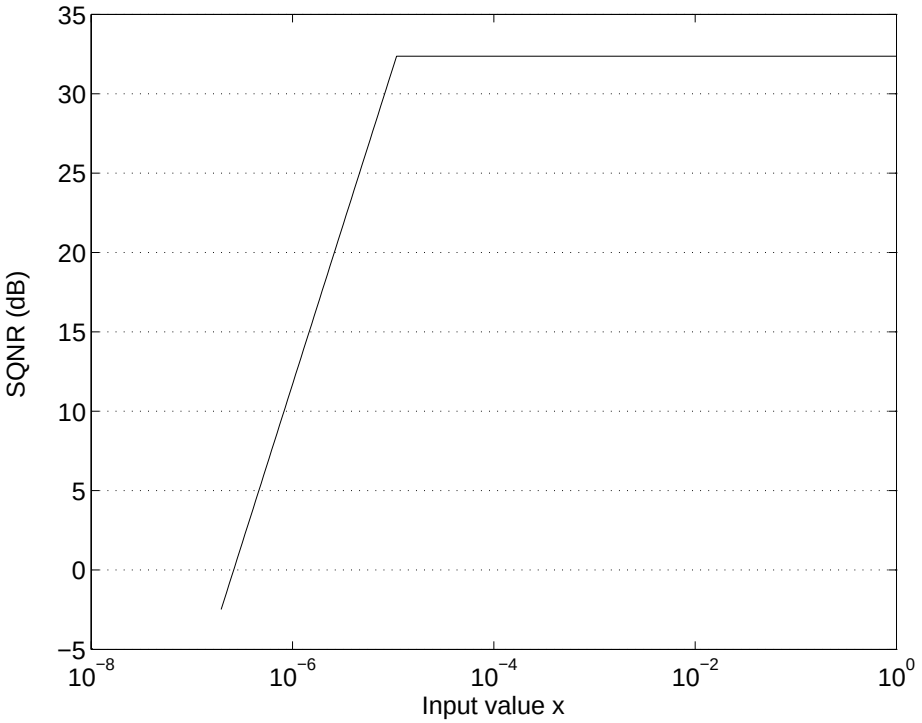

Figure 2.3: SQNR for 9-bit log representation.

2.2.3 10-bit Floating-point Representation

A number (x_f) in a 10-bit floating-point representation (1 sign bit, 4 exponent bits, and 5 mantissa bits) can be expressed in bit format as follows:

$$\underbrace{a_0}_{sign}\ \underbrace{a_1a_2a_3a_4}_{exponent}\ \underbrace{a_5a_6a_7a_8a_9}_{mantissa}$$

where a_0 is the MSB and a_9 is the LSB.

The value of x_f can be computed using the following formula. Let

$$\begin{aligned} sign &= a_0 \\ expo &= a_1 2^3 + a_2 2^2 + a_3 2^1 + a_4 2^0 \\ mant &= a_5 2^4 + a_6 2^3 + a_7 2^2 + a_8 2^1 + a_9 2^0 \end{aligned}$$

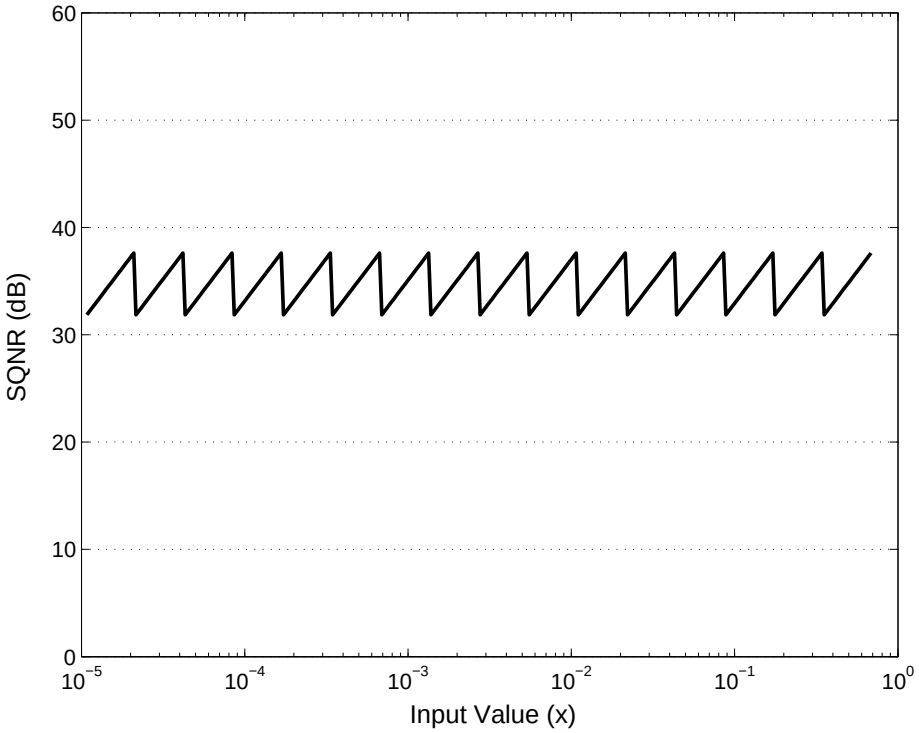

Figure 2.4: SQNR for 10-bit float representation.

Then,

$$x_f = (-1)^{sign} \times \frac{mant}{2^5} \times 2^{(expo)-(2^4-1)}$$

where mantissa is normalized with a leading '1', and the exponent is in excess notation with a bias of 15. In magnitude x_f ranges from 2^{-16} ($x_f = 0000010000_2$ or 1000010000_2) to 0.96875 ($x_f = 0111111111_2$ or 1111111111_2). Specifically, $x_f = 0000000000$ is used to represent zero value. Its dynamic range is 96.3 dB.

Figure 2.4 shows the range of representable values and the SQNR for those values in the 10-bit floating point representation. The SQNR of the floating-point representation varies in a sawtooth fashion, ranging from slightly over 30 dB to just below 40 dB.

2.2.4 Summary of Numerical Representation Properties

Table 2.1 summaries the properties of the three numerical representations in the above sections. Although the logarithmic representation uses fewer bits than the other two representations, it has the greatest dynamic range. The logarithmic representation also has flat SQNR over a big portion of its dynamic range, which is slightly over the required 30 dB. The linear representation gives the highest maximum SQNR, however, at low sign input level, the SQNR is lower than 30 dB. Table 2.1 also shows that the floating-point representation has properties in between the logarithmic and linear representations.

Table 2.1: Summary of the properties of the linear, logarithmic and floating-point representations.

numerical representation	dynamic range (dB)	min SQNR (dB)	max SQNR (dB)
16-bit linear	90.3	7.8	98.1
9-bit log	134.7	< 0	32.4
10-bit floating-point	96.3	31.9	37.6

2.3 Non-linear Amplifier

The following equation describes the functionality of the non-linear amplifier [13].

$$y = \begin{cases} A \times x & \text{if } |x| \leq t \\ B \times x^p & \text{if } |x| > t \end{cases}$$

where x is the input value (ranging from -1 to 1), y is the output, t is the compression threshold, A is the gain in the linear region, and p is the compression ratio (with values between $\frac{1}{4}$ and $\frac{1}{2}$). The value of B is determined by A, p, and t, ensuring the

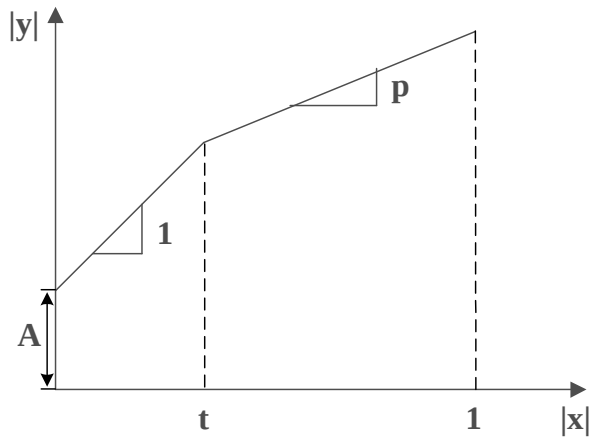

Figure 2.5: The non-linear amplifier function on a log-log scale.

two curves meet at the threshold. Since A and B are positive, y has the same sign as x. To simplify the computation, we restate the equation as follows.

$$y = sgn\,(x) \times \begin{cases} A \times |x| & \text{if } |x| \leq t \\ B \times |x|^p & \text{if } |x| > t \end{cases}$$

Figure 2.5 illustrates the non-linear amplifier function on a log-log scale. $|x|$ and $|y|$ denote the magnitudes of the input and output signals, respectively. At low signal levels (below the compression threshold) the non-linear amplifier has a linear response with a high gain. At high signal levels (above the compression threshold) signals experience $1 : p$ compression (p is the compression ratio). The two-piece amplification function has a unit slope for the first piece and p for the second piece.

2.3.1 Linear Representation

Implementation

Clearly, the implementation issues associated with the above expression center around the need to perform exponentiation. Here, we implement the computation of x^p by transforming it into $2^{p \times \log_2(x)}$. A polynomial approximation is then used for the

$\log_2(x)$ and 2^w ($w = p \times \log_2(x)$) implementations. Base 2 is chosen to enable the use of shifting for portions of the required transformations. Starting with the $\log_2(x)$ implementation, the Taylor series expansion for $\ln(1+u)$ ($0 \leq u \leq 1$) is well behaved and used extensively [19]. To convert into a range that can be directly used, we normalize x to

$$x = (1+u) \times 2^m$$

where u is from 0 to 1 and m is from -15 to -1. Then $\log_2(x)$ can be transformed as follows:

$$\begin{aligned}
\log_2(x) &= \left(\frac{1}{\ln(2)}\right) \times \ln(x) \\
&= \left(\frac{1}{\ln(2)}\right) \times (\ln(1+u) + m \times \ln(u)) \\
&= m + \left(\frac{1}{\ln(2)}\right) \times \ln(1+u) \\
&= m + \left(\frac{1}{\ln(2)}\right) \\
&\quad \times \left(C_1 \times u + C_2 \times u^2 + C_3 \times u^3 + C_4 \times u^4 + C_5 \times u^5 + C_6 \times u^6\right)
\end{aligned}$$

This results in a computation dominated by multiply-accumulate operations.

To compute 2^w ($w = p \times \log_2(x)$), we use a similar strategy to the one above. As mentioned previously, the range of p is limited to between $\frac{1}{4}$ and $\frac{1}{2}$. As a result, w is limited in range from -3.5 to 0 for p of $\frac{1}{4}$ and from -7 to 0 for p of $\frac{1}{2}$. Given that the Taylor expansion for e^x is accurate even with just a few terms when x is between 0 and 1, we split w into integer and fractional parts.

$$w = \lfloor w \rfloor + (w - \lfloor w \rfloor) = i + f,$$

where i denotes $\lfloor p \times \log_2(x) \rfloor$, which is a non-positive integer; and f denotes $(p \times \log_2(x) - \lfloor p \times \log_2(x) \rfloor)$, which is a positive fraction. Now,

$$2^{p \times \log_2(x)} = 2^{i+f} = 2^i \times 2^f$$

Since f ranges from 0 to 1 and $\ln(2)$ is less than 1, A Taylor expansion can be used to compute 2^f as follows.

$$\begin{aligned} 2^f &= e^{(f \times \ln(2))} \\ &= 1 + \ln(2) \times f \\ &\quad + \left(\left(\frac{1}{2!}\right) \times (\ln(2))^2\right) \times f^2 \\ &\quad + \left(\left(\frac{1}{3!}\right) \times (\ln(2))^3\right) \times f^3 \\ &\quad + \left(\left(\frac{1}{4!}\right) \times (\ln(2))^4\right) \times f^4 \\ &\quad + \left(\left(\frac{1}{5!}\right) \times (\ln(2))^5\right) \times f^5 \\ &\quad + \left(\left(\frac{1}{6!}\right) \times (\ln(2))^6\right) \times f^6 \end{aligned}$$

Thus, 2^f can be easily implemented using a multiply-accumulate unit. Since i is an integer, a simple shift operation is all that is necessary to implement 2^i.

The computation structure for x^p is shown in Figure 2.6. As analyzed above, the functions $\log_2(u)$ and 2^f can be implemented using a multiplier-accumulator. A Baugh-Wooley multiplier [2] is designed to minimize power consumption for the multiplication operation.

Figure 2.7 shows the computation structure of the complete non-linear amplifier using a linear numerical representation. The comparator compares the input data x and the threshold t. Based on the result of this comparison, x is sent to either

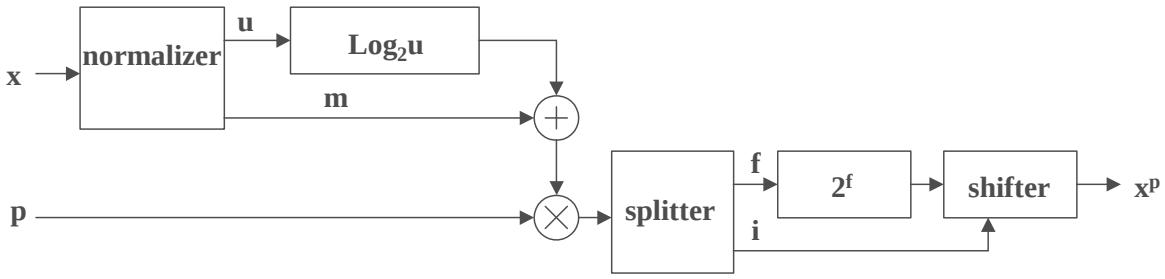

Figure 2.6: Computation structure for x^p using a linear representation.

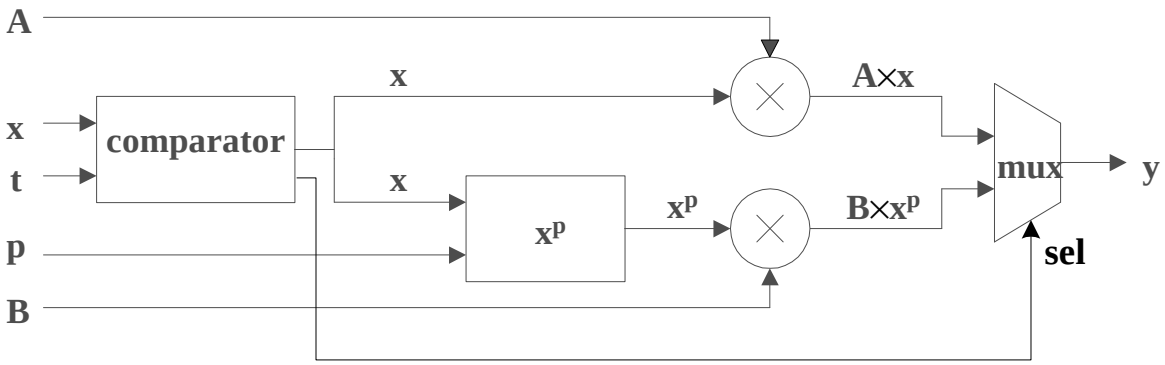

Figure 2.7: Computation structure for the non-linear amplifier using a linear representation.

the multiplier or exponentiation function at the next level. At the same time, only the upper path or the lower path is enabled to process the input data. When the processing is done, a sel signal will notify the component mux to send data out.

Error Analysis

Given that a polynomial approximation is used to compute x^p, we next examine how much error is introduced by this approximation. For comparison purposes, we implemented x^p in C. For the same input set, we compare the output of the C version of the exponentiation function (y) with the simulation output of the VHDL version (y'). The noise due to numerical error associated with the signal y is $|y - y'|$. The

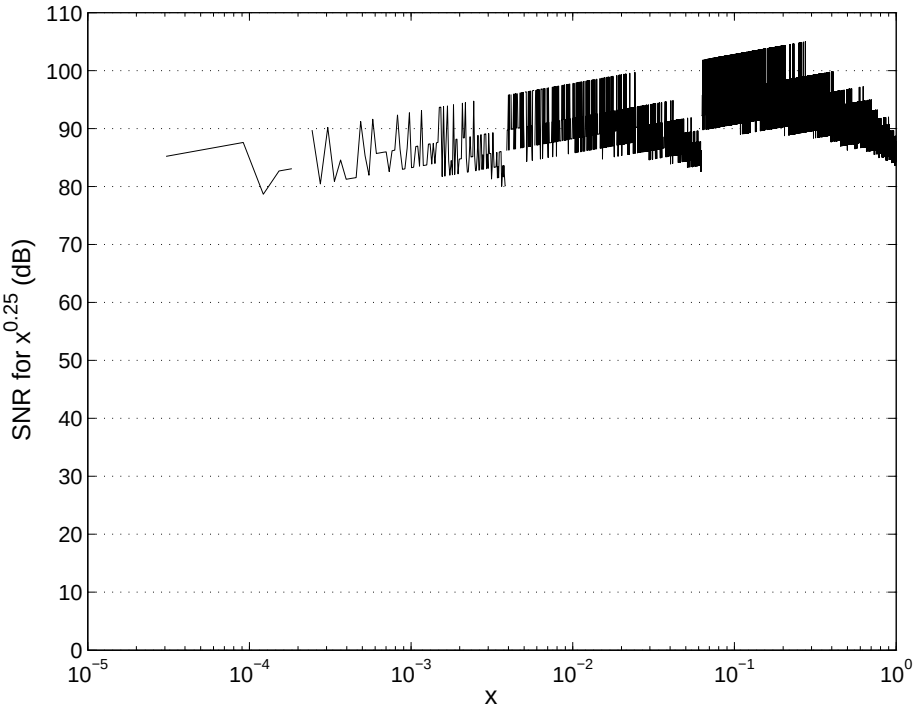

Figure 2.8: Signal to noise ratio for p = 0.25.

signal-to-noise ratio (SNR) is quantified as:

$$SNR = 20 \log \left(\frac{\frac{y}{\sqrt{2}}}{\frac{|y-y'|}{\sqrt{12}}} \right)$$

The SNR for a normalized signal is commonly used to evaluate the validity of an implementation [4]. In the above expression, the input signal is assumed to be a sinusoid with amplitude y and the numerical error is assumed to be uniformly distributed in the range 0 to $|y - y'|$.

Figure 2.8 through Figure 2.10 show three comparison results for different values of p. Some points are missing in the figures because when $|y - y'|$ is equal to 0 the corresponding SNR is infinity. Compared to the SQNR for the 16-bit linear representation, SNR for the x^p computation is well above 30 dB, especially for small values of x, implying sufficient precision in the polynomial approximation.

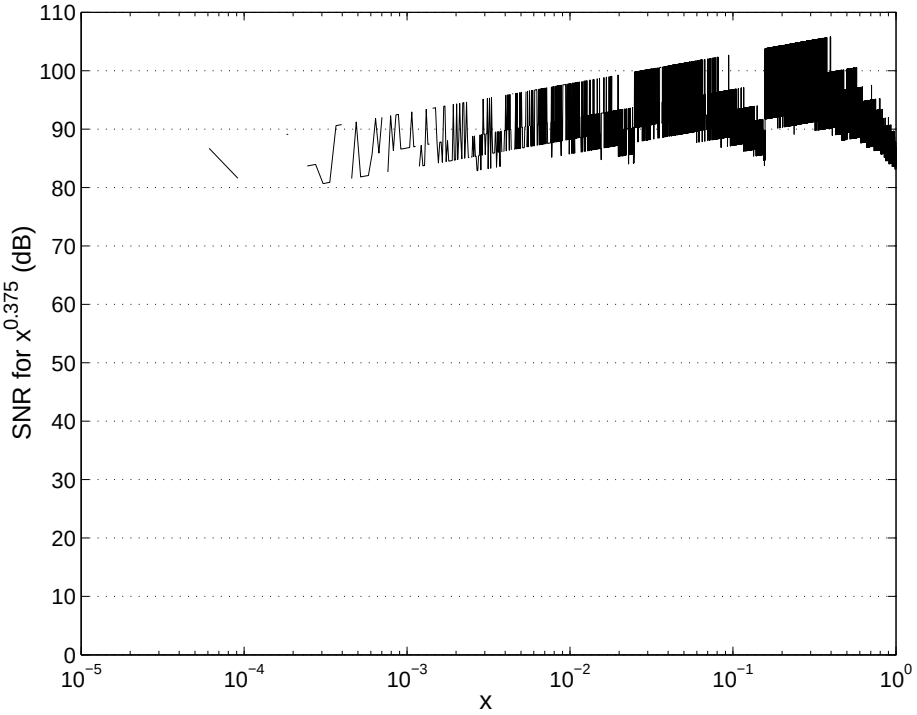

Figure 2.9: Signal to noise ratio for p = 0.375.

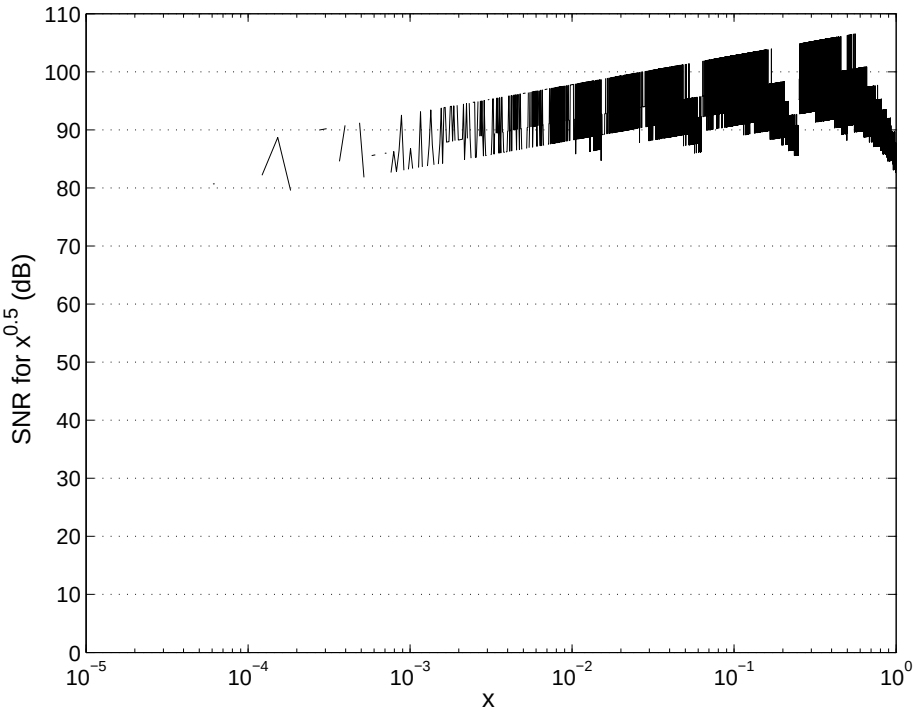

Figure 2.10: Signal to noise ratio for p = 0.5.

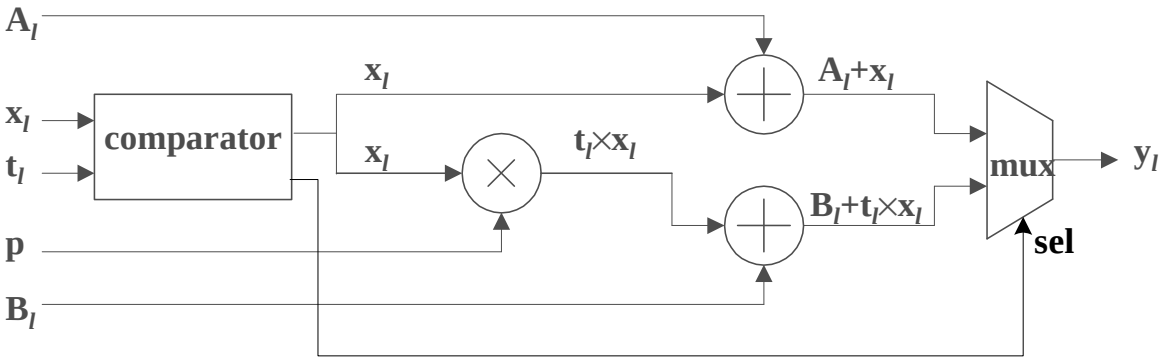

Figure 2.11: Computation structure for the non-linear amplifier using logarithmic representation.

2.3.2 Logarithmic Representation

In the above implementation of the non-linear amplifier, most of the power is consumed computing x^p, which is comprised of a number of multiply-accumulate operations. By using a logarithmic representation, the exponentiation operation is implemented as a multiplication and multiplication is implemented as an addition. We expect to see significant power savings using the logarithmic representation.

Implementation

The following equation shows the functionality of the non-linear amplifier using a logarithmic representation.

$$y_l = sgn\,(x_l) \times \begin{cases} A_l + |x_l| & \text{if } |x_l| \geq t_l \\ B_l + p \times |x_l| & \text{if } |x_l| < t_l \end{cases}$$

where y_l denotes $\log_{0.941}(y)$ and similar notation applies to A_l, B_l, x_l and t_l.

Figure 2.11 shows the computation structure of the non-linear amplifier using a logarithmic representation. It has similar structure as that of the linear representation amplifier in Figure 2.7 except that internal processing modules are much simpler.

2.3.3 Floating-point Representation

Implementation

The following equation shows the functionality of the non-linear amplifier using a floating-point representation.

$$y_f = sgn\left(x_f\right) \times \begin{cases} A_f \times |x_f| & \text{if } |x_f| \leq t_f \\ B_f \times |x_f|^p & \text{if } |x_f| > t_f \end{cases}$$

Like the linear representation non-linear amplifier, the implementation of the exponentiation is the key part in the floating-point representation non-linear amplifier. The same technique used in the implementation of the linear representation non-linear amplifier could also be used here. However, we will exploit the limited number of bits in each field of the floating-point representation. Since parameter p (between $\frac{1}{4}$ and $\frac{1}{2}$) is fixed for the individual patient who uses the hearing aid, and exponent and mantissa have only 4 and 5 (with a leading '1') bits each, two 16 entry LUTs were deployed for the table lookup of the exponentiation of the exponent and mantissa. In the normalizer, the mantissa of the lookup results are normalized, with the exponent of each lookup result adjusted accordingly. Figure 2.12 shows the computation structure of the non-linear amplifier using a floating-point representation. It has similar structure as that of the linear representation amplifier in Figure 2.7 except that the exponent implementation uses table lookup.

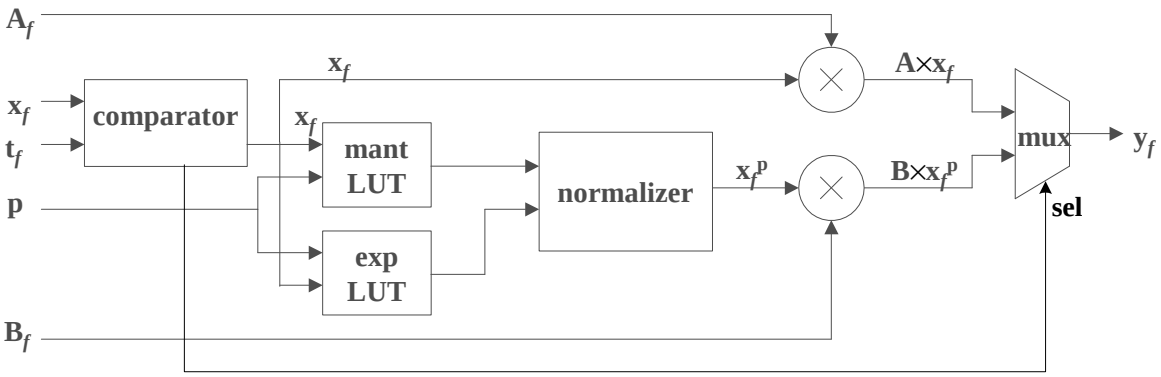

Figure 2.12: Computation structure for the non-linear amplifier using floating-point representation.

Chapter 3

Full Hearing Aid Channel Design

In this chapter, we first introduce the implementations of the FIR filter, another fundamental component in the hearing aid design. This is the work of E. Hemmeter [18]. Using the FIR filter and the non-linear amplifier we construct a full hearing aid channel (Specifically, channel 6 of Figure 2.1).

3.1 Implementation of FIR Filter

The Finite Impulse Response (FIR) filters perform the calculation shown below.

$$y(j) = \sum_{i=1}^{n} c(i)x(j-i)$$

where n is the number of taps in the filter, $c(i)$ is the coefficient of tap i, and $x(j-i)$ is the $(j-i)$th input value. The filters in the design of [14] have 21 taps and therefore 21 coefficients. Figure 3.1 and Figure 3.2 show the frequency response and impulse response of the FIR filter [18].

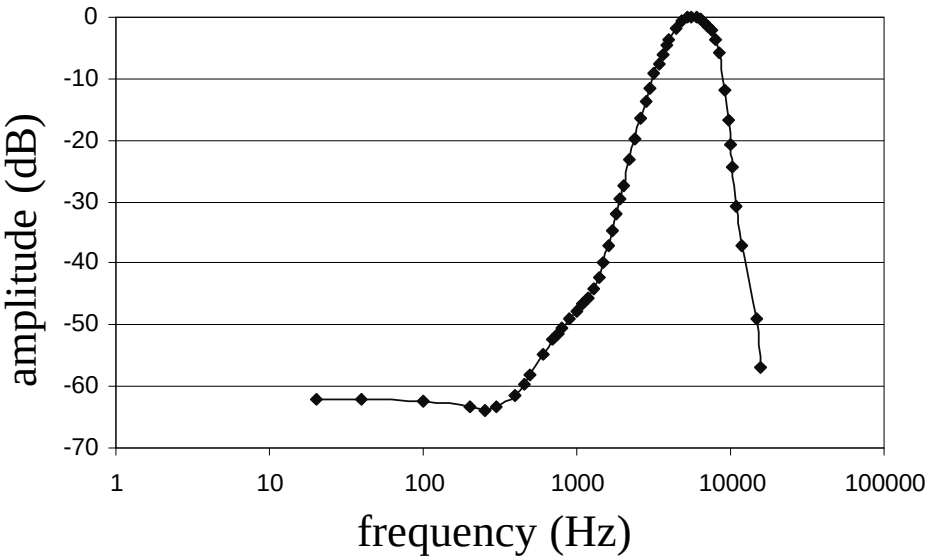

Figure 3.1: Frequency response of the FIR filter.

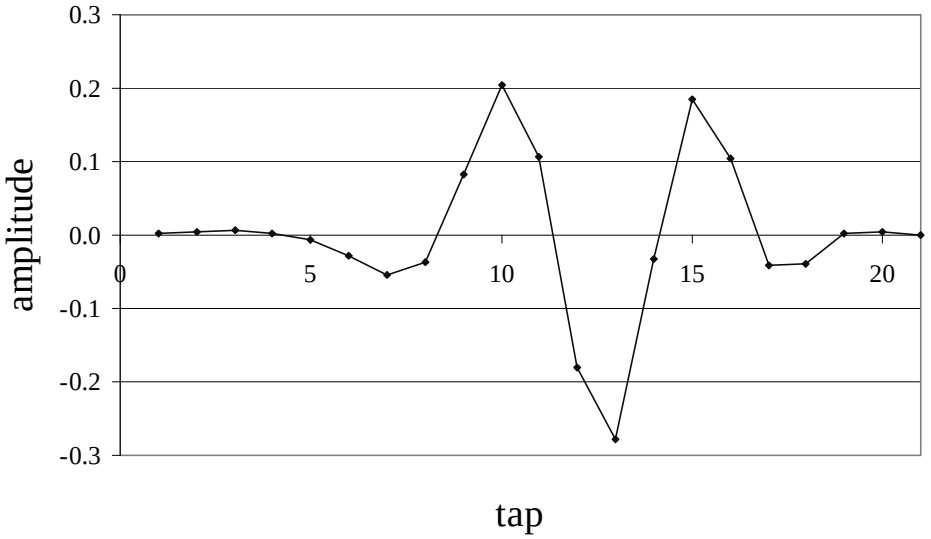

Figure 3.2: Impulse response of the FIR filter.

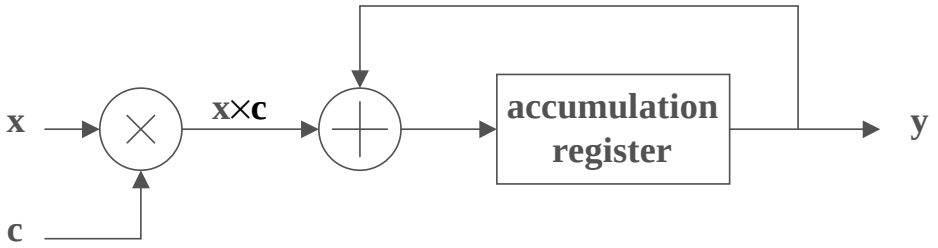

Figure 3.3: Computation structure for the FIR using linear representation.

3.1.1 Linear Representation

As shown in Figure 3.3, the filter implementation is straightforward with a linear representation. To minimize power consumption a Baugh-Wooley [2] multiplier is used instead of a simple shift-add multiplier. The accumulation is a basic summation function with some clipping logic. After 21 cycles the accumulated result is available to be output.

3.1.2 Log Representation

The logarithmic implementation (Figure 3.4) of an FIR filter is not as straightforward as the linear implementation. The filter still performs multiply-accumulation operations, but the multiplication is implemented using an adder and the accumulation is implemented using a LUT. The adder consumes much less power than a multiplier, and the power consumption of the look-up table is minimized by only using it when required.

Clearly, an adder is used to implement multiplication, since

$$\log(x \times y) = \log(x) + \log(y)$$

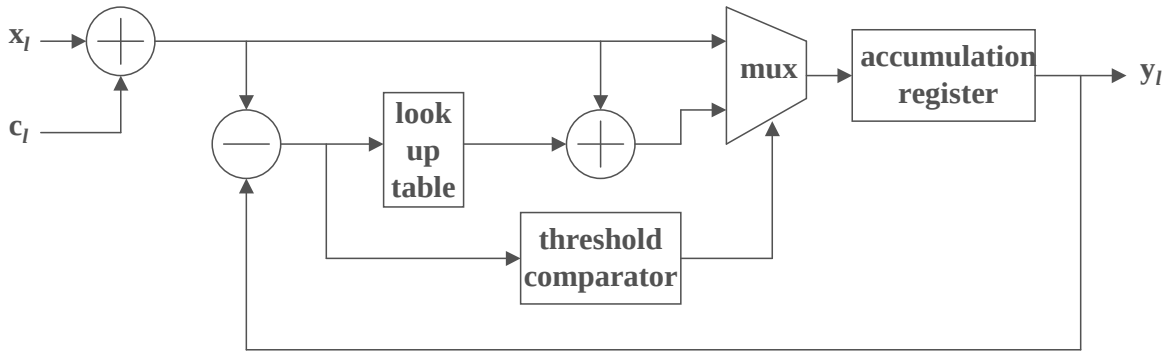

Figure 3.4: Computation structure for the FIR using logarithmic representation.

so to compute $z = x \times y$, $z_l = x_l + y_l$ is implemented. To implement accumulation the following relationship is exploited

$$\log(x+y) = \log(\frac{x+y}{x}) + \log(x) = \log(1+\frac{y}{x}) + \log(x)$$

This calculation is instantiated as $LUT(y_l - x_l) + x_l$, where $LUT(a_l)$ computes $\log(1+a_l)$ and $a_l = y_l - x_l$. If necessary, the inputs x and y can be exchanged so that x is larger than y (i.e., y_l is larger than x_l since the base of the logarithm is 0.941).

To conserve power the look-up table is only accessed if the difference, $y_l - x_l$, is small enough that y is significant with respect to x (i.e., y is outside the quantization noise of x). If x is much larger than y, the value of x_l is passed to the output without exercising the look up table.

3.1.3 Floating-point Representation

The structure of the floating-point FIR unit is shown in Figure 3.5. It operates as follows: the data input and coefficient are separated into their three parts: sign bit, exponent bits, and mantissa bits; the mantissa bits are multiplied together using partial products, while the exponents are added; the sign bits are combined using an exclusive-or function to give the sign of the multiply result. Next, the mantissa of

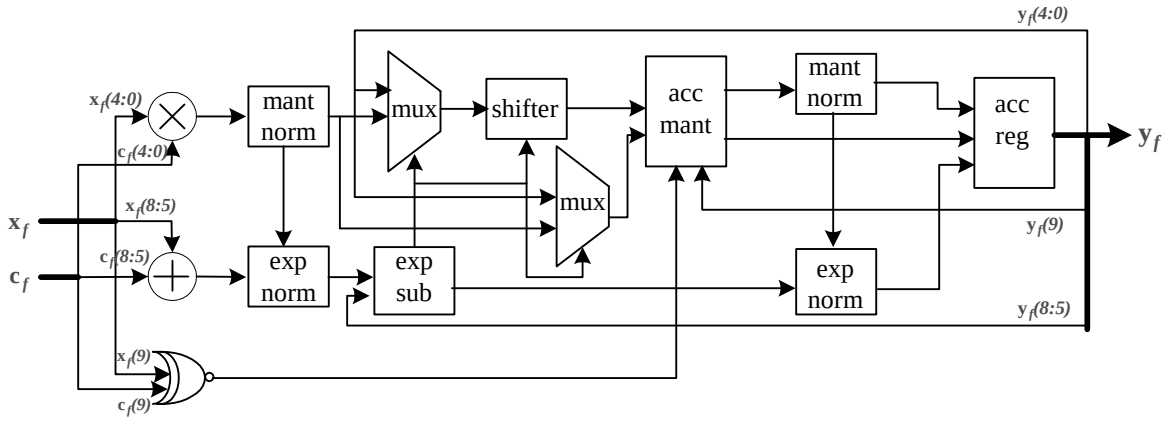

Figure 3.5: Computation structure for the FIR using floating-point representation.

the resulting multiply is normalized and the exponent is adjusted appropriately. The result of the multiply is compared to the current output of the accumulator register. First, the mantissa of the number with the smaller exponent is shifted, in preparation for the add, such that both numbers have the same exponent. The shifted mantissas are then added together and the result is normalized, again adjusting the exponent appropriately. The final result is then latched in the register when the controller asserts the latch enable signal.

3.2 Implementation of A Full Channel

A full hearing aid channel is constructed using two FIR filters and one non-linear amplifier. It is executed at a rate of 32 kS/s, which corresponds to channel 6 in Figure 2.1. All the parameters of the non-linear amplifier take on realistic values, so that the simulation results indicate how the circuit operates in typical usage.

Figure 3.6 shows the implementation block diagram of a hearing aid channel. For a single data input, a 21-tap FIR filter needs more processing time than the non-linear amplifier. Therefore, a master control is designed to synchronize the FIR filters and non-linear amplifier. The master controller monitors the output enable

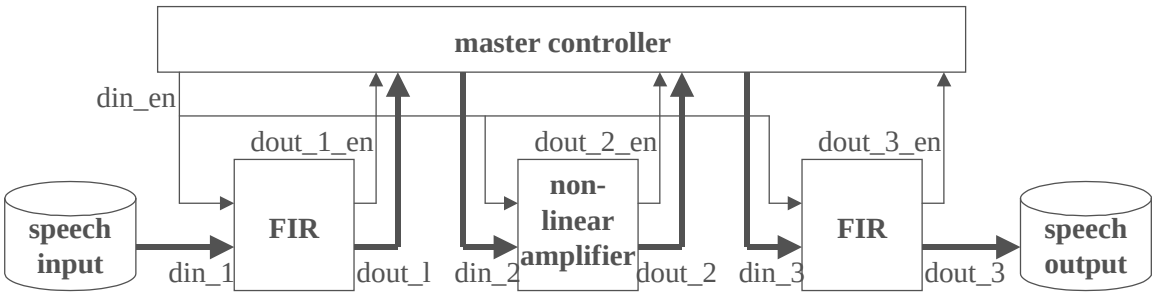

Figure 3.6: Implementation block diagram of a hearing aid channel.

signals. When any of them becomes valid (high for one clock cycle), the corresponding output data are latched. When seeing all three valid output enable signals, the master controller releases an input enable signal. In the meanwhile, the output data latched in the master controller are released to the downstream component. For the first FIR filter, input data comes from a file, where sampled speech data are stored. The output data from the 2nd FIR filter are also written to a file.

Chapter 4

Power Consumption

4.1 Modeling Power Consumption

The post-synthesis logic-level simulation models digital systems in a discrete-event fashion. [7] validated the use of discrete-event simulation models for power estimation in this type of system. Reporting transition counts explicitly is strictly an energy measure; however, by keeping the number of input samples constant across tests, energy is linearly related to power.

The input set was three seconds of audio sampled at 32 kS/s from the Speech In Noise (SPIN) audiological test for human speech intelligibility [3, 20] (Figure 4.1). All of the designs are specified using the VHDL language and during the synthesis step targeted to either an FPGA or an ASIC. The FPGA target is a Xilinx Virtex 2000E, and the ASIC target is the ADK standard-cell library from the Mentor Graphics Higher Education Program [1]. Table 4.1 shows a list of standard cells in the ADK standard-cell library.

Table 4.1: A list of standard cells in the ADK standard-cell library.

cell name					
and02	and03	and04	or02	or03	or04
nand02	nand03	nand04	nor02	nor03	nor04
inv01	inv02	inv04	inv08	inv12	inv16
buf02	buf04	buf08	buf12	buf16	mux21
latch	latchr	latchs	latchsr	fake_vcc	fake_gnd
dff	dffr	dffs	dffsr	fadd1	hadd1
sff	sffr	sffs	sffsr	xnor2	xor2
ao21	ao22	ao221	ao32	aoi21	aoi22
aoi221	aoi222	aoi32	aoi321	aoi322	aoi33
aoi332	aoi333	aoi422	aoi43	aoi44	oai21
oai22	oai221	oai222	oai32	oai321	oai322
oai33	oai332	oai333	oai422	oai43	oai44
tri01	trib04	trib08			

4.1.1 Simulation Results for the Non-linear Amplifier

In this section, we show the simulation results for the non-linear amplifier exercised using unfiltered speech input vectors. Results of the filtered speech data is presented in the next subsection. Table 4.2 shows the signal transition counts in the post-synthesis simulation of the non-linear amplifiers. For the FPGA target, the signal transition counts for the linear numeric representation are 69.7 times and 18.4 times greater than those for the logarithmic representation and floating-point representation, respectively. For the ASIC target, the logarithmic representation and floating-point representation achieve 54.4 times and 10.2 times power savings relative to the linear representation, respectively. For both FPGA and ASIC targets, the non-linear amplifier using the linear numerical representation is the most power consuming implementation. This is because the component "x^p" (Figure 2.6) with twelve 16-bit multipliers and one 20-bit multiplier consumes more than 75% of the total power. While the relative power consumption across targets is inappropriate to judge using

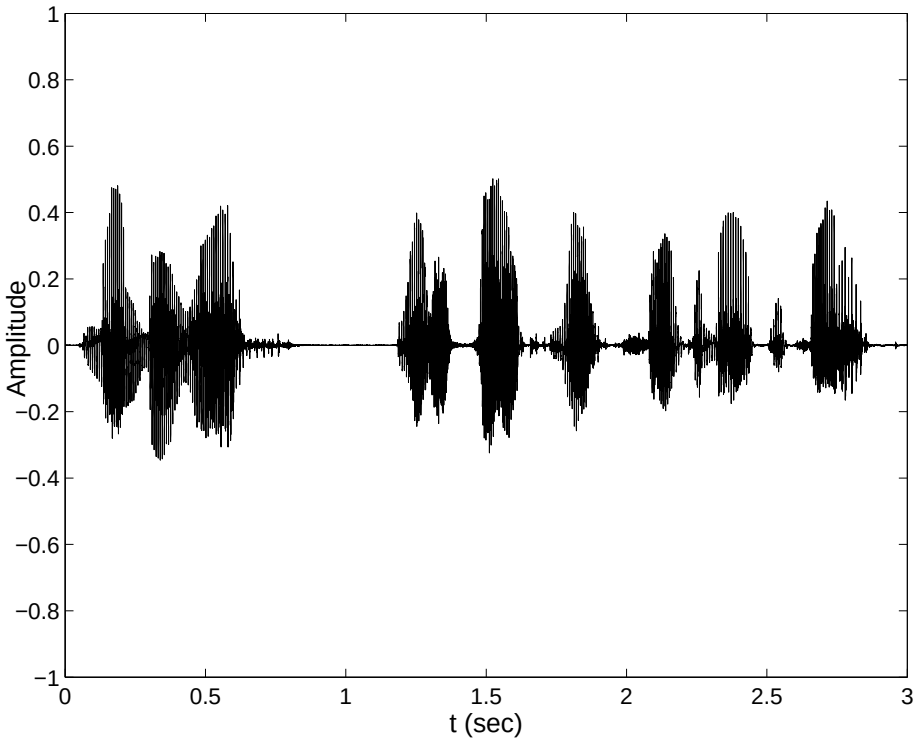

Figure 4.1: Waveform of the 3 second speech input.

signal transition counts, clearly a significant savings is present independent of the implementation technology for both logarithmic and floating-point representations verses linear representation.

Table 4.2: Signal transition counts for the non-linear amplifier exercised using speech input vectors.

numerical representation	FPGA target	ASIC target
linear	1.484×10^9	3.213×10^9
log	2.129×10^7	5.910×10^7
floating-point	8.080×10^7	3.145×10^8

4.1.2 Simulation Results for a Hearing Aid Channel

FPGA-targeted Hearing Aid Channel

Table 4.3 shows the signal transition counts for the two FIR filters and the non-linear amplifier (NLA) for an FPGA-targeted hearing aid channel using speech input vectors. The last row shows the total transition counts of the hearing aid channel. Figure 4.2 and Figure 4.3 present the signal transition counts and relative power savings for the FPGA-targeted hearing aid channel. Note that the two identical FIR filters have different signal transition counts for each of the three numerical representations. This is due to the difference in the input vectors to the two FIR filters. Although the power savings within the non-linear amplifier is dramatic for the logarithmic representation verses the linear representation, the overall power savings is limited to 2.5×. This is because the total power consumption is dominated by the two FIR filters, where the logarithmic representation has slightly less than a 2.5× power improvement over the linear representation. Within the non-linear amplifier, the power savings for the floating-point representation compared to the linear representation is 14×. However, the total power saving is less than unity because for both FIR filters the signal transition counts using floating-point are 17.6% and 13.4% greater than those using the linear representation, respectively. In effect, the power penalty in the FIR filters is greater than the power savings realized in the non-linear amplifier.

ASIC-targeted Hearing Aid Channel

Table 4.4 shows the signal transition counts for the ASIC-targeted hearing aid channel. The signal transition counts and power savings in graphical form are shown in Figure 4.4 and Figure 4.5. Again, speech input is used to exercise the channel. Not surprisingly, for logarithmic and linear representation, the results tell a similar

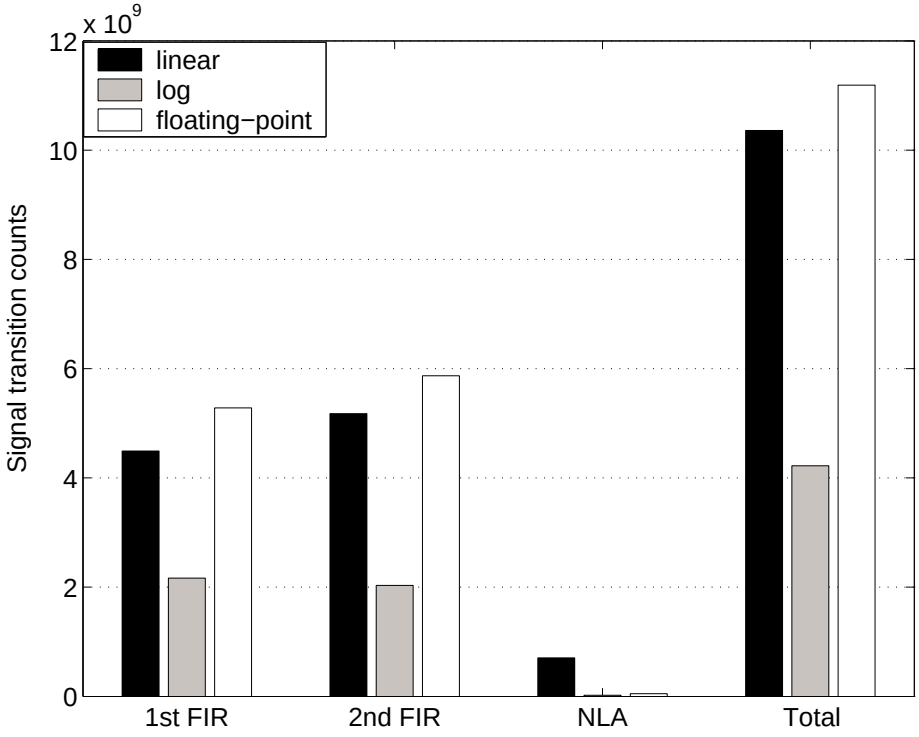

Figure 4.2: Signal transition counts for FPGA-targeted hearing aid channel.

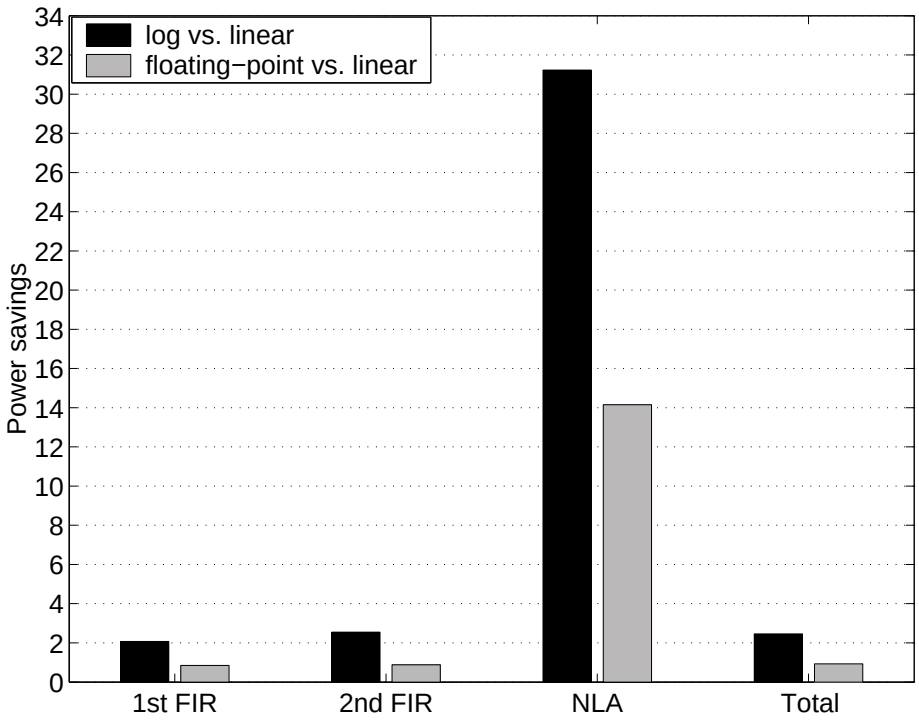

Figure 4.3: Power savings for FPGA-targeted hearing aid channel.

Table 4.3: Signal transition counts for an FPGA-targeted hearing aid channel exercised using speech input vectors.

FGPA target			
numerical representation	linear	log	floating-point
1st FIR	4.492×10^9	2.167×10^9	5.281×10^9
2nd FIR	5.176×10^9	2.033×10^9	5.869×10^9
nonlinear amplifier	7.045×10^8	2.256×10^7	4.977×10^7
total transition counts	1.037×10^{10}	4.223×10^9	1.119×10^{10}

story as that in the previous section, although the platforms are different. The total power savings using logarithmic representation verses linear representation is 2.9×, slightly higher than 2.5×. One interesting phenomenon is that for the FPGA-targeted FIRs in Figure 4.2, the average signal transition counts using the floating-point representation is 15.5% greater than those using the linear representation; while for the ASIC-targeted FIR in Figure 4.4, the average signal transition counts using the floating-point representation is 53.9% lower than those using the linear representation. Due to this reason, the overall power savings using the floating-point representation is improved by 2.1× over the linear representation.

Table 4.4: Signal transition counts for an ASIC-targeted hearing aid channel exercised using speech input vectors.

ASIC target			
numerical representation	linear	log	floating-point
1st FIR	7.763×10^9	3.022×10^9	4.292×10^9
2nd FIR	7.301×10^9	2.953×10^9	3.820×10^9
nonlinear amplifier	2.350×10^9	6.134×10^7	2.924×10^8
total transition counts	1.741×10^{10}	6.036×10^9	8.406×10^9

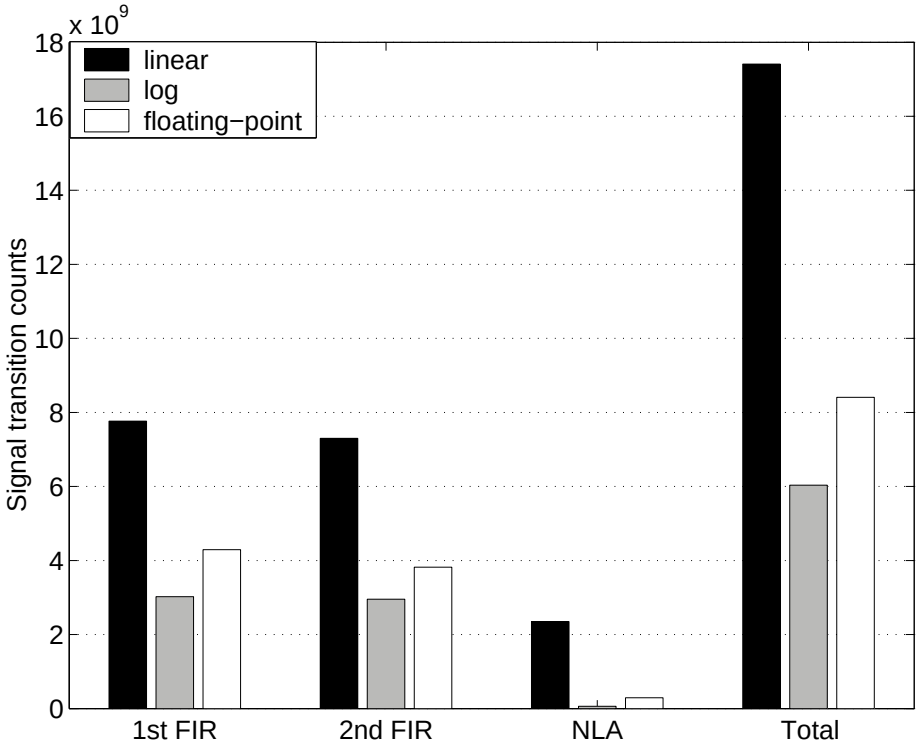

Figure 4.4: Signal transition counts for ASIC-targeted hearing aid channel.

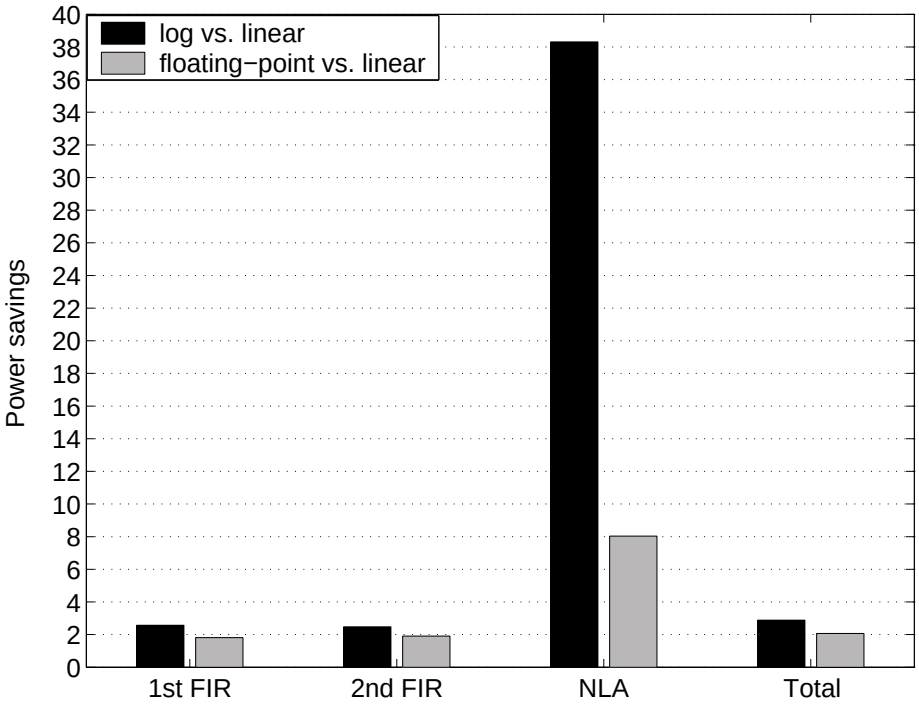

Figure 4.5: Power savings for ASIC-targeted hearing aid channel.

4.2 Results for the Full Hearing Aid

Using the results from section 4.1.2, we can readily calculate the relative power savings for two 6-channel hearing aids (Figure 1.1 and Figure 2.1). There exist many other hearing aid architectures, while we limit our investigation to these two architectures. Although memory requirement and chip area of a hearing aid design are very important evaluation factors, our primary interest is in the power consumption of these two architectures. For simplicity, we assume that different filters consume the same amount of energy, modeled by the mean of the two measured filters. This may not be the case since the input vectors to each filter are different, which can result in different signal transition counts. Although the bandpass filter coefficients are identical, a different set of coefficient are used to implement the lowpass filter. Therefore if we apply the same speech input vectors used in the above sections to a multirate hearing aid Figure 2.1, the following formula gives us the total transition counts.

$$
\begin{aligned}
C_{total} &= C_6 + C_5 + C_4 + C_3 + C_2 + C_1 \\
&= 3C_{FIR} + C_{NLA} \\
&\quad + \frac{1}{2}(4C_{FIR} + C_{NLA}) \\
&\quad + \frac{1}{4}(4C_{FIR} + C_{NLA}) \\
&\quad + \frac{1}{8}(4C_{FIR} + C_{NLA}) \\
&\quad + \frac{1}{16}(4C_{FIR} + C_{NLA}) \\
&\quad + \frac{1}{32}(3C_{FIR} + C_{NLA}) \\
&= 6.844C_{FIR} + 1.967C_{NLA}
\end{aligned}
$$

where C_{total} is the total transition counts, C_{FIR} and C_{NLA} denote the transition counts of the FIR filter on average and the non-linear amplifier, respectively. The signal transition counts for channel 6 to channel 1 are denoted by C_6 to C_1. The signal transition counts of each channel are also listed separately. The last term gives the total transition counts. Note that for the purposes of this calculation, the lowpass filters on the left side of the channels are considered with the higher-frequency channel, since they operate at the higher sample rate. On the contrary, the lowpass filters on the right side of the channels are considered with the lower-frequency channel because of their operation on the lower sample rate.

Similarly, for the design using a uniform sampling rate (unirate) shown in Figure 1.1, the total signal transition counts are given by:

$$\begin{aligned} C_{total} &= C_6 + C_5 + C_4 + C_3 + C_2 + C_1 \\ &= 2C_{FIR} + C_{NLA} \\ &\quad +4C_{FIR} + C_{NLA} \\ &\quad +8C_{FIR} + C_{NLA} \\ &\quad +16C_{FIR} + C_{NLA} \\ &\quad +32C_{FIR} + C_{NLA} \\ &\quad +64C_{FIR} + C_{NLA} \\ &= 126C_{FIR} + 6C_{NLA} \end{aligned}$$

Note that the length of the FIR filters in Figure 1.1 increases inversely with frequency range of each channel.

The total signal transition counts for the twelve designs shown in Tables 4.5 and 4.6 provide estimates of their relative efficiency.

Table 4.5: Total signal transition counts for multirate and unirate hearing-aid designs on FPGA target.

	FPGA target		
	linear	log	floating-point
multirate	3.447×10^{10}	1.438×10^{10}	3.825×10^{10}
unirate	6.133×10^{11}	2.647×10^{11}	7.028×10^{11}

Table 4.6: Total signal transition counts for multirate and unirate hearing-aid design on ASIC target.

	ASIC target		
	linear	log	floating-point
multirate	5.617×10^{10}	2.056×10^{10}	2.833×10^{10}
unirate	9.631×10^{11}	3.768×10^{11}	5.128×10^{11}

In general, it is seen that FPGA-targeted and ASIC-targeted hearing aid implementations provide approximately 43:1 and 46:1 power savings with multirate-log processing over unirate-linear processing, respectively. No matter on which platform (FPGA or ASIC) the design is targeted, the FIR filters are the dominating component in power consumption. As can be observed in the above tables, about 18:1 is the benefit from multirate processing over unirate processing.

Although our primary interest is in the power consumption of these two hearing aid architectures, memory requirements and chip area are also important design criteria. Our calculation shows that an estimated 1008 buffers are needed for the equalizations and FIR filters in the multirate architecture; for the unirate architecture, the total number of buffers needed by the FIR filters are nearly 2000. This significant memory requirement implies that both designs are too memory intensive and with current technology an in-the-ear aid is unlikely to be able to afford that much memory due to area constraints.

Chapter 5

Conclusion

5.1 Remarks

In this thesis, the power savings associated with constructing a hearing aid using a numerical representation customized to the needs of the application is investigated. Specifically, we compare the relative power consumption of three designs, one using a traditional 16-bit linear representation, one using a 9-bit logarithmic representation, and the other using a 10-bit floating-point representation. Each design is targeted in two directions, an FPGA implementation and an ASIC implementation. Signal transition counts in the post-synthesis simulation are used to evaluate relative power consumption. A hearing aid component, non-linear amplifier, is implemented and evaluated. A complete hearing aid channel is also constructed. The non-linear amplifier implementation using a logarithmic numerical representation is shown to provide significant power savings (more than 30×) over that of a traditional linear numeric representation. The power improvement using floating-point representation is about 14 and 8 times over using linear representation for FPGA and ASIC targets, respectively. Since the total power consumption is dominated by the FIR filters, the total

power saving is on the order of the filters. Using the results for an individual channel, we also compute the relative power savings for the entire hearing aid for multirate processing and unirate processing. Although the specific results presented here are limited to 9-bit logarithmic, 10-bit floating-point, and 16-bit linear representations, the general message is much broader. With the ready availability of FPGA and ASIC fabrication, rigid, fixed-function computational hardware is no longer a necessity of modern digital system design, and significant power savings can result if the requirements of the application are used to specify the properties of the numerical representation.

5.2 Future Work

In [7], Chamberlain et al. evaluated the validity of using signal transition counts to model actual power consumption for audio signal processing applications targeted on an ASIC platform. We have assumed that the same relationship exists for the hearing aid applications targeted on an FPGA platform. Our future work is to build a hearing aid channel on the Field Programmable Port Extender (FPX) platform [24], measure the real power consumption, and compare the power numbers with the signal transition counts obtained from the post-synthesis logic simulation.

In this thesis, the floating-point representation investigated used a traditional sign-magnitude format. Hemmeter [18] has shown a benefit in terms of power consumption for multiply accumulate operations if the mantissa of the floating-point number is represented in two's complement form, enabling the use of a Baugh-Wooley multiplier. The use of this floating-point representation in the non-linear amplifier is also of interest.

References

[1] Mentor Graphics Higher Education Program: ASIC Design Kit. http://www.mentorg.com/partners/hep/AsicDesignKit/ASICindex.html, 1997-2001.

[2] C. Baugh and B. Wooley. A two's complement parallel array multiplication algorithm. *IEEE Transactions on Computers*, 22, 1973.

[3] R.C. Bilger, J.M. Nuetzel, W.M. Rabinowitz, and C. Rzechowski. Standardization of a test of speech perception in noise. *Speech Hear. Res.*, 1984.

[4] D. Cachera and T. Risset. Advances in bit width selection methodology. In *Proc. of the IEEE Int'l Conf. on Application-Specific Systems, Architectures, and Processors*, 2002.

[5] R. Chamberlain, Y.H. Chew, V. DeAlwis, E. Hemmeter, J. Lockwood, R. Morley, E. Richter, J. White, and H. Zhang. Novel numerical representations for low-power audio signal processing. In *Proc. of Int'l Hearing Aid Research Conf.*, 2002.

[6] R. Chamberlain, Y.H. Chew, V. DeAlwis, E. Hemmeter, J. Lockwood, R. Morley, E. Richter, J. White, and H. Zhang. Power consumption of customized numerical representations for audio signal processing. In *Proc. of High Performance Embedded Computing Workshop*, 2002.

[7] R. Chamberlain, E. Hemmeter, R. Morley, and J. White. Modeling the power consumption of audio signal processing computations using customized numerical representations. In *Proc. of 36th Annual Simulation Symposium*, pages 249–255, 2003.

[8] Junghwan Choi, Jinhuan Jeon, and Kiyoung Choi. Power minimization of functional units by partially guarded computation. In *Proc. of Int'l Symp. on Low Power Electronics and Design*, 2000.

[9] H. Dillon. Compression? yes, but for low or high frequencies, for low or high intensities, and for what response times? *Ear and Hearing*, 17:287–307, 1996.

[10] G. L. Engel, R. E. Morley, S. W. Kwa Jr., and R. J. Fretz. Integrated circuit logarithmic digital quantizers with applications to low-power data interfaces for speech processing. *VLSI Signal Processing IV*, 1990.

[11] Altaf Abdul Faffar, Wayne Luk, Peter Y.K. Cheung, and Nabeel Shirazi. Customising floating-point designs. In *Proc. of the 10th IEEE Symp. on Field-Programmable Custom Computing Machines*, 2002.

[12] J.L. Goldstein. Modeling rapid compression on the basilar membrane as multiple-bandpass nonlinearity filtering. *Hear. Res.*, 49:39–60, 1990.

[13] J.L. Goldstein. Hearing aids based on models of cochlear compression using adaptive compression thresholds. U.S. Patent Appl. ♯20020057808, May 2002.

[14] J.L. Goldstein. PI, Hearing aids based on models of cochlar compression, NIH SBIR grant DC04028 Phases I and II. awarded to BECS Technology, Inc., Jan. 1999 - July 2003.

[15] J.L. Goldstein, M. Oz, P. Gilchrist, and M. Valente. Alternative compressive hearing aid algorithms derived from loudness psychophysics and cochlear models. In *Proc. of Int'l Hearing Aid Research Conf.*, 2002.

[16] J.L. Goldstein, M. Valente, and R.D. Chamberlain. Acoustic and psychoacoustic benefits of adaptive compression thresholds in hearing aid amplifier that mimic cochlear function. *Hear. Res.*, 109:2355(A), 2001.

[17] J.L. Goldstein, M. Valente, R.D. Chamberlain, P. Gilchrist, and D. Ivanovich. Pilot experiments with a simulated hearing aid based on models of cochlear compression. In *Proc. of Int'l Hearing Aid Research Conf.*, 2000.

[18] Eric Hemmeter. Reducing power consumption using customized numerical representations in digital hearing aids. Master's thesis, Washington University, May 2003.

[19] Texas Instruments. TMS320C54x DSP library programmer's reference, 2001.

[20] D.N. Kalikow, K.N. Stevens, and L.L. Elliott. Development of a test of speech intelligibility in noise using sentence materials with controlled word predictability. *Acoust. Soc. Am.*, 1977.

[21] N.Y.S. Kiang, M.C. Liberman, W.F. Sewell, and J.J. Guinan. Single unit clues to cochlear mechanisms. *Hear. Res.*, 22:171–182, 1986.

[22] S.W. Kwa. *Design and analysis of a hybrid sign/logarithm delta-sigma analog-to-digital converter for audio applications.* PhD thesis, Washington University, 1996.

[23] R.P. Lippmann, L.D. Braida, and N.I. Durlach. Study of multichannel amplitude compression and linear amplification for persons with sensorineural hearing loss. *Acoust. Soc. Am.*, 69:524–534, 1981.

[24] John W. Lockwood, Naji Naufel, Jon S. Turner, and David E. Taylor. Reprogrammable network packet processing on the Field programmable Port Extender (FPX). In *ACM International Symposium on Field Programmable Gate Arrays (FPGA'2001)*, pages 87–93, Monterey, CA, USA, February 2001.

[25] M.Valente, D.A. Fabry, L. Potts, and R.E. Sandlin. Comparing the performance of the widex senso digital hearing aid with analog hearing aids. *Am. Acad. Audiol.*, 9:342–360, 1998.

[26] Takanori Okuma, Yun Cao, Masanori Muroyama, and Hiroto Yasuura. Reducing access energy of on-chip data memory considering active data bitwidth. In *Proc. of Int'l Symp. on Low Power Electronics and Design*, 2002.

[27] R. Plomp. Noise, amplification and compression: Consideration of three main issues in hearing aid design. *Ear and Hearing*, 15:2–21, 1994.

[28] L. Robles, M.A. Ruggero, and N. C. Rich. Basilar membrane mechanics at the base of the chinchilla cochlea. I. input-output function, tuning curves, and response phases. *Acoust. Soc. Am.*, 80:1364–1374, 1986.

[29] John R. Sacha and Mary Jane Irwin. The logarithmic number system for strength reduction in adaptive filtering. In *Proc. of Int'l Symp. on Low Power Electronics and Design*, 1998.

[30] S. D. Soli. Hearing aids: Today and tomorrow. *Acoust. Soc. Am.*, 1994.

[31] S.D. Soli. Offline simulations of logarithmic FIR filters: Measurements of processing distortion and perrceptual tests of its audibility. Technical report, Bioscience Laboratory, 3M Center, 1988.

[32] T.J. Sullivan. *Estimating the power consumption of custom CMOS digital signal processing integrated circuits for uniform and logarithmic number systems.* PhD thesis, Washington University, 1993.

[33] E. Villchur. Signal processing to improve speech intelligibility in perceptive deafness. *Acoust. Soc. Am.*, 53:1646–1657, 1973.

Vita

Jing Lu

Date of Birth	Febuary 4, 1974
Place of Birth	Xinxiang, China
Degrees	B.S. Electrical Engineering, June 1997 M.S. Computer Engineering, May 2003

May 2003