

Washington University in St. Louis

## Washington University Open Scholarship

---

All Computer Science and Engineering  
Research

Computer Science and Engineering

---

Report Number: WUCSE-2003-30

2003-05-01

### Reducing Power Consumption Using Customized Numerical Representations in Digital Hearing Aids

Eric E. Hemmeter

This thesis examines the effects of changing the numerical representation of audio signals in digital hearing aids to minimize power consumption. Within the hearing aid design a majority of the power used is consumed in the many finite impulse response filters. The main processing involved in these filters is a multiply-accumulate function. We examine the power consumption of 12 different multiply-accumulate units that use the following numerical representations: a 16-bit linear representation, a 9-bit logarithmic representation, and 10 different floating-point representations ranging from 9 to 13 bits. A selection of the multiply-accumulators are simulated using a continuous-circuit simulator. The... [Read complete abstract on page 2.](#)

Follow this and additional works at: [https://openscholarship.wustl.edu/cse\\_research](https://openscholarship.wustl.edu/cse_research)

---

#### Recommended Citation

Hemmeter, Eric E., "Reducing Power Consumption Using Customized Numerical Representations in Digital Hearing Aids" Report Number: WUCSE-2003-30 (2003). *All Computer Science and Engineering Research*. [https://openscholarship.wustl.edu/cse\\_research/1076](https://openscholarship.wustl.edu/cse_research/1076)

Department of Computer Science & Engineering - Washington University in St. Louis  
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

## Reducing Power Consumption Using Customized Numerical Representations in Digital Hearing Aids

Eric E. Hemmeter

### Complete Abstract:

This thesis examines the effects of changing the numerical representation of audio signals in digital hearing aids to minimize power consumption. Within the hearing aid design a majority of the power used is consumed in the many finite impulse response filters. The main processing involved in these filters is a multiply-accumulate function. We examine the power consumption of 12 different multiply-accumulate units that use the following numerical representations: a 16-bit linear representation, a 9-bit logarithmic representation, and 10 different floating-point representations ranging from 9 to 13 bits. A selection of the multiply-accumulators are simulated using a continuous-circuit simulator. The power estimates from this are compared with signal transition counts from a discrete event simulator to quantify the relationship between transition counts and power consumption. This relationship is then used to examine other numerical representations.

# **Reducing Power Consumption Using Customized Numerical Representations in Digital Hearing Aids**

**Eric E. Hemmeter**

Eric E. Hemmeter, "Reducing Power Consumption Using Customized Numerical Representations in Digital Hearing Aids," Master's Thesis, Technical Report WUCSE-2003-30, 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

---

REDUCING POWER CONSUMPTION USING CUSTOMIZED NUMERICAL  
REPRESENTATIONS IN DIGITAL HEARING AIDS

by

Eric E. Hemmeter

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

---

REDUCING POWER CONSUMPTION USING CUSTOMIZED NUMERICAL  
REPRESENTATIONS IN DIGITAL HEARING AIDS

by Eric E. Hemmeter

---

ADVISOR: Professor Roger Chamberlain

---

May, 2003  
Saint Louis, Missouri

---

This thesis examines the effects of changing the numerical representation of audio signals in digital hearing aids to minimize power consumption. Within the hearing aid design a majority of the power used is consumed in the many finite impulse response filters. The main processing involved in these filters is a multiply-accumulate function. We examine the power consumption of 12 different multiply-accumulate units that use the following numerical representations: a 16-bit linear representation, a 9-bit logarithmic representation, and 10 different floating-point representations ranging from 9 to 13 bits. A selection of the multiply-accumulators are simulated using a continuous-circuit simulator. The power estimates from this are compared with signal transition counts from a discrete event simulator to quantify

the relationship between transition counts and power consumption. This relationship is then used to examine other numerical representations.

# Contents

List of Tables . . . . .	v
List of Figures . . . . .	vi
Acknowledgments . . . . .	viii
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 How the Ear Works . . . . .	2
1.2 Hearing Aid Requirements . . . . .	7
1.3 Hearing Aid Architecture . . . . .	8
1.4 Related Work . . . . .	11
1.5 Contributions . . . . .	14
1.6 Organization . . . . .	15
1.7 Acronyms . . . . .	16
<b>2 Numerical Representations . . . . .</b>	<b>18</b>
2.1 Linear Representation . . . . .	19
2.2 Logarithmic Representation . . . . .	21
2.3 Floating Point Representations . . . . .	25
2.4 Revised Floating Point Representation . . . . .	29
2.5 Representation Characteristics . . . . .	31
2.5.1 Signal to Quantization Noise . . . . .	31
2.5.2 Dynamic Range . . . . .	33
<b>3 Finite Impulse Response Filters . . . . .</b>	<b>37</b>
3.1 Linear MAC . . . . .	38
3.1.1 Baugh-Wooley Multiplier . . . . .	38
3.2 Logarithmic MAC . . . . .	39

3.3	Floating Point MAC . . . . .	41
3.4	Revised Floating Point MAC . . . . .	42
3.5	Simulation Modeling . . . . .	43
3.5.1	Input Sets . . . . .	43
3.5.2	Verification Steps . . . . .	45
<b>4</b>	<b>Power Measurements . . . . .</b>	<b>48</b>
4.1	Technologies . . . . .	48
4.2	Signal Transition Counts . . . . .	49
4.3	MachPA Power Simulation . . . . .	51
4.4	Power vs. Signal Transition Counts . . . . .	51
4.5	Power Relationship Across Representations . . . . .	54
4.6	Power Consumption for Entire Aid . . . . .	58
4.7	Conclusions . . . . .	61
<b>5</b>	<b>Summary and Conclusions . . . . .</b>	<b>62</b>
5.1	Future Work . . . . .	63
	<b>References . . . . .</b>	<b>66</b>
	<b>Vita . . . . .</b>	<b>69</b>



# List of Tables

2.1	Linear representation examples . . . . .	21
2.2	Logarithmic representation examples . . . . .	23
2.3	1-4-5 floating point representation examples . . . . .	28
2.4	4-6 revised floating point representation examples . . . . .	31
3.1	Final Baugh-Wooley Array of Partial Products . . . . .	39
4.1	Power consumption with respect to the linear representation. . . . .	55
4.2	Power consumption of a full hearing aid. . . . .	61

# List of Figures

1.1	The outer ear. . . . .	2
1.2	The inner ear. . . . .	3
1.3	Some common sounds and their sound pressure level. . . . .	4
1.4	Normal hearing range compared to an impaired hearing range. . . . .	5
1.5	Audiogram of a normal hearing person. . . . .	6
1.6	Audiogram of an impaired hearing person. . . . .	7
1.7	Normal hearing range compared to an amplified hearing range. . . . .	8
1.8	Unirate hearing aid architecture. . . . .	9
1.9	Multirate hearing aid architecture. . . . .	10
1.10	Impulse response of the 4-8 kHz band-pass FIR filter. . . . .	12
1.11	Frequency response of the 4-8 kHz band-pass FIR filter. . . . .	13
2.1	Block diagram of the hearing aid from microphone to speaker. . . . .	19
2.2	A comparison of linear inputs and their logarithmic equivalents. . . . .	23
2.3	A comparison of linear inputs plotted on a log scale to their logarithmic equivalents. . . . .	24
2.4	Close up of $x = 1$ on linear vs logarithmic comparison. . . . .	25
2.5	Close up of $x$ close to 0 on linear vs logarithmic comparison. . . . .	26
2.6	SQNR of Linear Representation. . . . .	32
2.7	SQNR of Logarithmic Representation. . . . .	33
2.8	SQNR of 1-4-4 Representation. . . . .	34
2.9	SQNR of 1-4-5 Representation. . . . .	35
2.10	SQNR of 1-4-6 Representation. . . . .	36
2.11	Dynamic range of various representations. . . . .	36
3.1	Block diagram of linear MAC. . . . .	38
3.2	Block diagram of logarithmic MAC. . . . .	40

3.3	Block diagram of 1-5-5 MAC. . . . .	41
3.4	Revised floating point block diagram for two's complement representation. . . . .	43
3.5	The speech input set. . . . .	44
3.6	The random input set. . . . .	45
3.7	The verification flow for ASIC testing. . . . .	46
3.8	Actual 1-5-5 hardware output overlaid with golden model output for a music sample. . . . .	47
4.1	One unit of an FPGA. . . . .	49
4.2	1-5-5 IC station layout. . . . .	52
4.3	The relationship between power and signal transition count. . . . .	53
4.4	Estimated power for ASIC target with speech input for all representations. . . . .	54
4.5	Estimated power for ASIC target with random input for all representations. . . . .	55
4.6	Breakdown of power consumption within logarithmic MAC with random input. . . . .	56
4.7	Breakdown of power consumption within logarithmic MAC with speech input. . . . .	57
4.8	Signal transition counts for FPGA target with speech input. . . . .	58
4.9	Signal transition counts for FPGA target with random input. . . . .	59

# Acknowledgments

I would like to thank the many people that assisted me. Professor Roger Chamberlain for the idea, the guidance, and all of his support including the repetition to make up for my lacking background. Jason White for his work in the initial phases of the project and the many tools he wrote that made my life that much easier. Ed Richter for the original FIR controller and the knowledge he shared so readily. Varuna DeAlwis, Yen Hsiang Chew, and Huakai Zhang for their work on various components and designs of MACs. The Washington University Computer Engineering department and the National Science Foundation and it's GK-12 program for financial support.

I would also like to thank Laura, my sister, Pete, my brother-in-law, and my nephews, Thomas, and Matthew, for the many home cooked meals and time out to play. My parents for their support throughout this process and throughout my life leading up to it. Lastly to my future wife, Ginny, thank you for all of the late night talks and reassurances, the many trips between St. Louis, Indianapolis, and Chicago, and for all your love and support.

Eric E. Hemmeter

*Washington University in Saint Louis*  
*May 2003*

# Chapter 1

## Introduction

Over 28 million Americans have some level of sensorineural hearing loss. However, only about 5 million of these people actually own a hearing aid and of those only a little over half are happy with the performance of their aid [24]. The largest complaint has to do with understandability of speech in noise. Activities like having a conversation in a crowded restaurant or while walking down a busy street are not handled well by current hearing aid designs. The aids cannot distinguish between speech and other noise or speech that the user is interested in and speech that the user wants to ignore. Current plans for overcoming these issues involve more signal processing which in turn requires more power. Currently, batteries in full featured aids only last a few days. With the joint pressures of more power required and trying to fit the whole aid into the ear canal (which severely restricts battery volume) any opportunity to save power is quite important.

This thesis examines using different numerical representations for the audio signals in an attempt to exploit the properties of these representations to minimize power consumption. We examine 12 different numerical representations: a 16-bit linear representation, a 9-bit logarithmic representation, and 10 different floating point

representations ranging from 9 to 13 bits. For each representation we evaluate the power requirements for multiply-accumulate operations, the dominant computation required for audio signal processing applications.

## 1.1 How the Ear Works

The outer ear, seen in Figure 1.1 [5], is made up of the pinna, the ear canal, the ear drum, and the three smallest bones in the human body, the malleus, the incus, and the stapes. The pinna, with all of its folds, funnels compression waves down into the ear canal. At the end of the ear canal is the ear drum, a very thin membrane that is

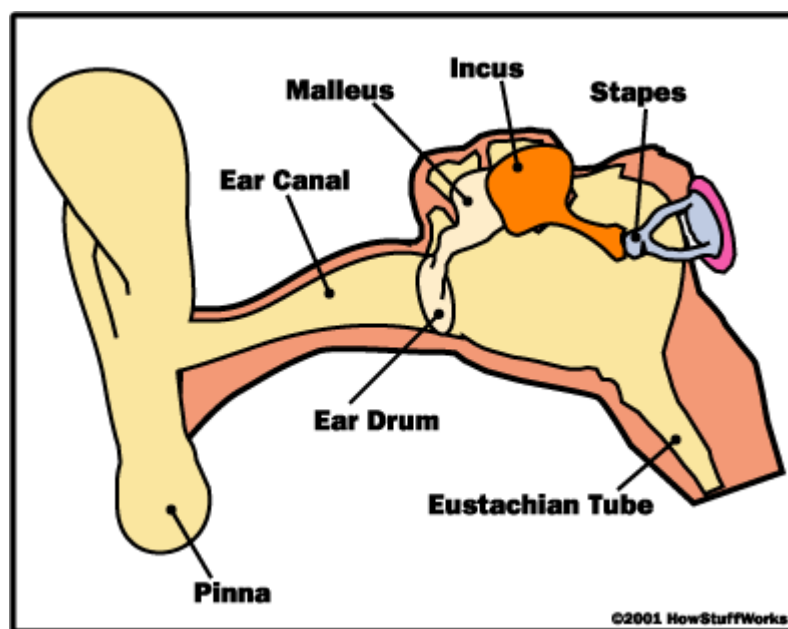


Figure 1.1: The outer ear.

vibrated by the compression waves traveling through air that make up sound. These vibrations are passed to the small bones, the malleus, or hammer, the incus, or anvil, and the stapes, or stirrup. These three bones amplify the incoming vibrations and pass them along to the cochlea, seen in Figure 1.2 [5]. Inside the cochlea is a fluid

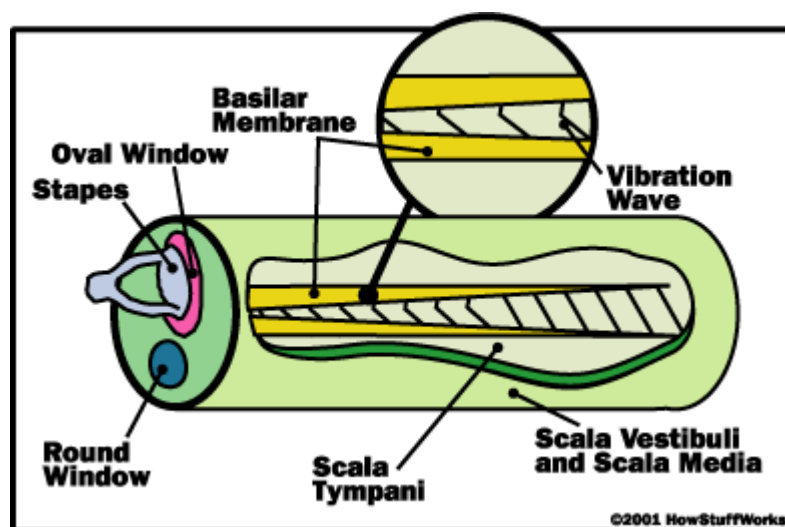


Figure 1.2: The inner ear.

that helps with balance and allows the vibrations from the stapes to propagate along the basilar membrane. All along the surface of the membrane are 20 to 30 thousand little hairs. Each of these hairs is a different length and thus resonates at a different frequency. The hairs respond non-linearly to changes in loudness [11]. These hairs are what allow humans to normally hear between the frequencies of 20 Hz to 20 kHz. The hairs are connected to the auditory nerve in such a way that when a sound at a certain frequency resonates with a hair an electrical impulse is sent to the brain. The brain uses what it knows of the shape of the pinna to help interpret the signals into recognizable sounds and to determine the direction that the sound came from.

Normal hearing humans have a lower threshold of hearing at 0 dB sound pressure level (SPL). This is point at which sounds just become noticeable. The upper range is limited by the pain threshold at approximately 100 dB SPL. This is the level at which sounds can be painful and physically damaging. Figure 1.3 shows some common noises and their corresponding loudness in dB SPL. Sounds above 100 dB

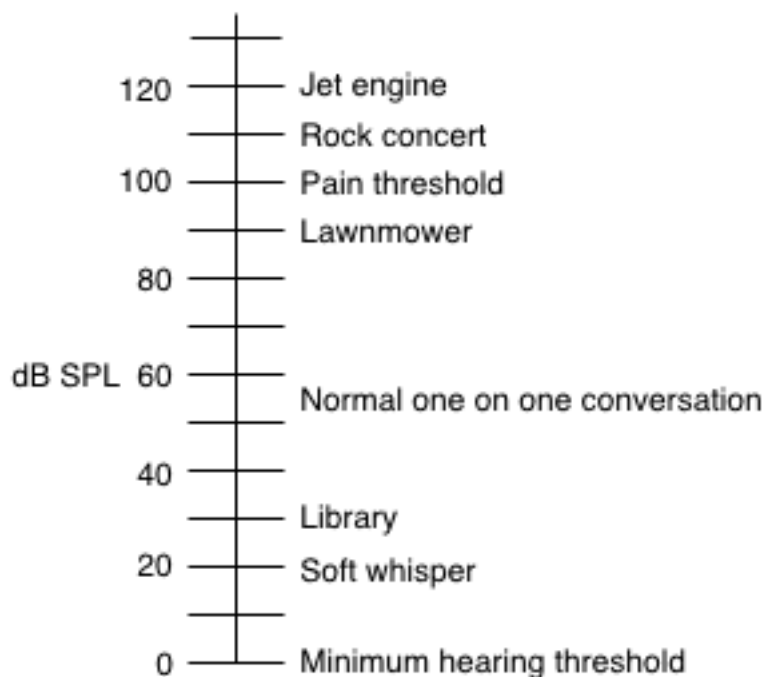


Figure 1.3: Some common sounds and their sound pressure level.

SPL can damage the hairs that react to different frequencies. With repeated damage they have a decreased ability to respond to input.

Sensineural hearing loss, the most common form of hearing impairment, is characterized by an increase in the lower threshold of hearing ability and little or no increase in the pain threshold. This is illustrated in Figure 1.4. With an increased lower threshold the user cannot hear quiet sounds without amplification. This increase in the lower threshold is generally frequency dependent, often based on the original sound that damaged the hairs. The most common form of hearing loss is a sloping loss with high frequencies more attenuated than low frequencies. Figure 1.5 shows an audiogram of an average hearing person, while Figure 1.6 shows an audiogram of someone with this common loss. The audiograms chart the level (in dB SPL) at which the individual can hear sounds based on frequency. In the normal audiogram



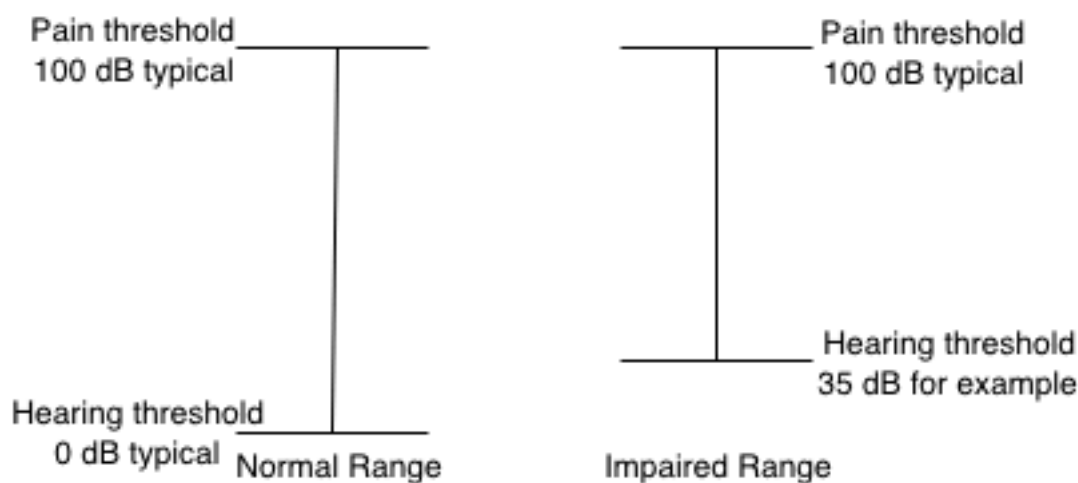


Figure 1.4: Normal hearing range compared to an impaired hearing range.

the minimum threshold level is pretty constant throughout the frequency range at around 10 dB showing that this person has a slight reduction in hearing ability. In the impaired audiogram there is significant hearing loss at the higher frequencies (i.e., the minimum threshold is 65 dB SPL at 8 kHz), while at low frequencies there is still some loss, but not as significant (i.e., the minimum threshold is 30 dB SPL at 250 Hz).

To compensate for the frequency dependence of hearing loss, modern hearing aids separate the audio signals into separate frequency bands and provide a gain that is tailored to the patient's needs within each frequency band.

Early hearing aid designs provided linear amplification, however, loud sounds can be amplified above the pain threshold as illustrated in Figure 1.7. This could allow quiet sounds to be heard, but sounds that were previously able to be heard may now damage the ear further. In an attempt to help prevent this, slowly adapting aids have been designed. The idea behind these is that generally environments have a relatively stable sound level. If a patient moves from a loud environment to a quiet environment

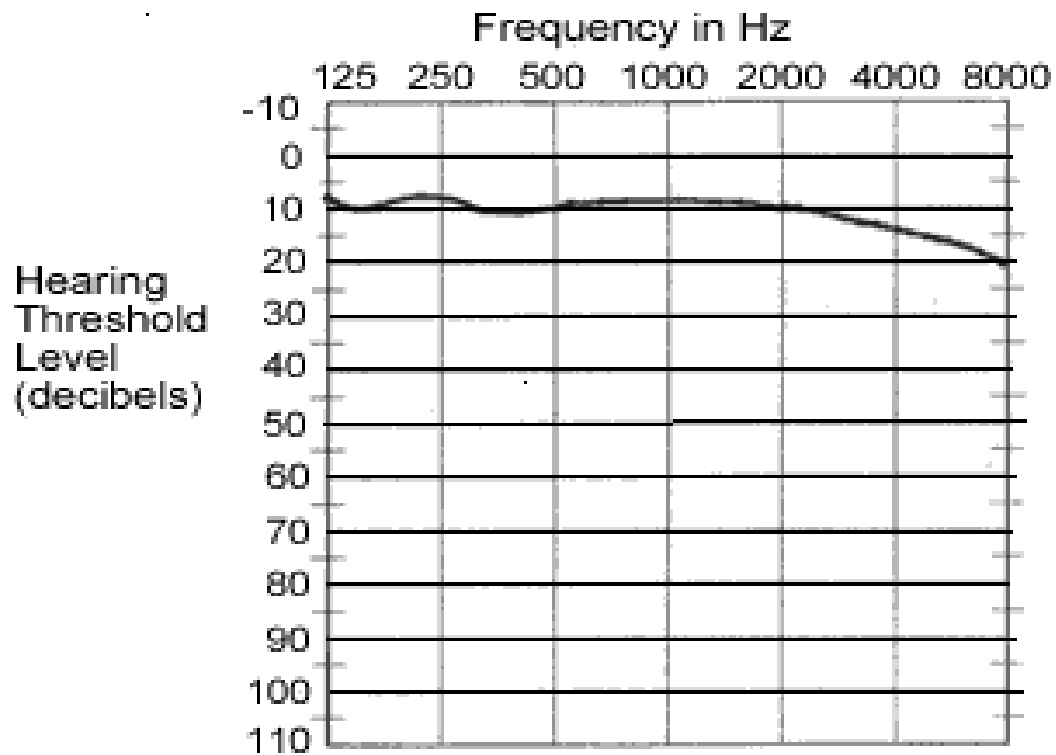


Figure 1.5: Audiogram of a normal hearing person.

the aid will slowly increase the amount that it amplifies its input. Conversely, as the patient moves from a quiet environment to a loud one the aid will decrease the amount that it amplifies. However, the slow adaptation isn't always fast enough. Sounds like a phone ringing during a quiet conversation can still be amplified above the pain threshold. To compensate for these shortcomings and to more accurately mimic the cochlea, instantaneous non-linear amplification is currently being investigated [15].

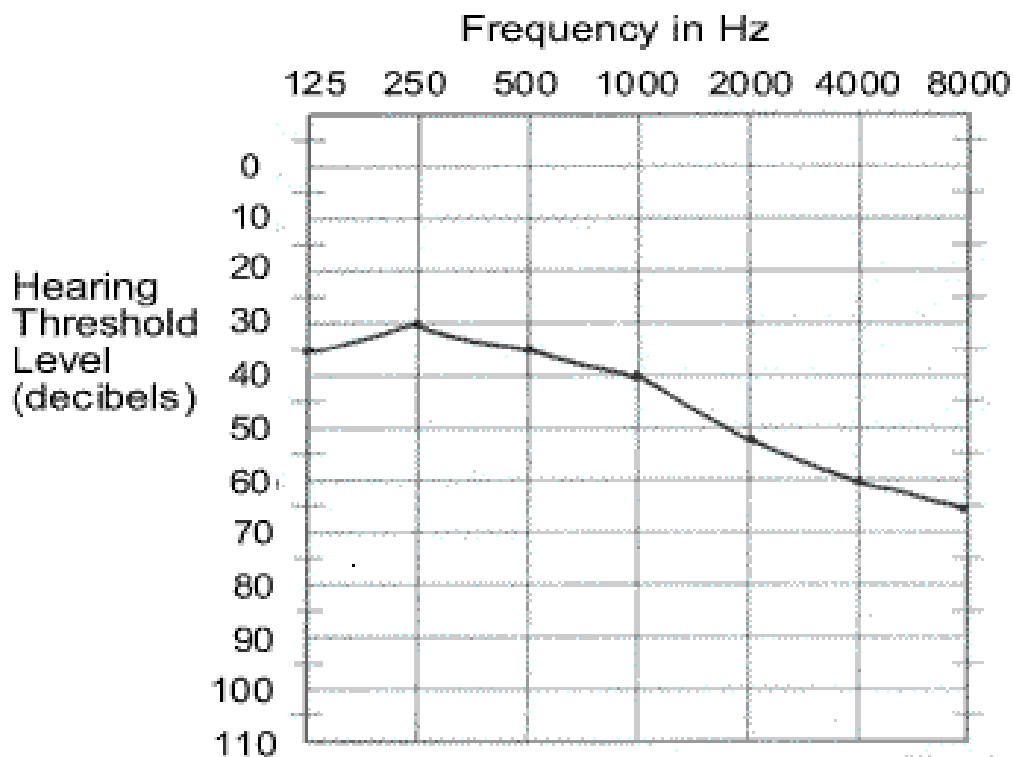


Figure 1.6: Audiogram of an impaired hearing person.

## 1.2 Hearing Aid Requirements

To mimic the normal range of human hearing a hearing aid should optimally have an input dynamic range of 100 dB. It has been shown empirically that speech understandability does not improve with signal to quantization noise ratios (SQNR) greater than 30 dB. So a hearing aid should have an SQNR of at least 30 dB [25].

Also for understandability reasons it is desirable to have the aid be able to process 32 thousand samples per second. During non-linear amplification high order odd-harmonics will be added to the signals. By having a sample rate so high above

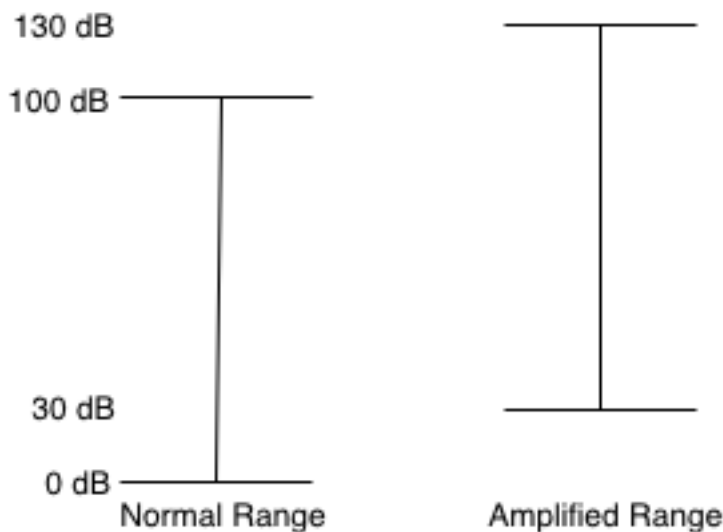


Figure 1.7: Normal hearing range compared to an amplified hearing range.

the Nyquist rate these harmonics will not be aliased down into the frequencies of speech data.

### 1.3 Hearing Aid Architecture

In an attempt to overcome the issues present in current hearing aids the design in Figure 1.8 is currently under investigation by Goldstein [15].

This design breaks the input signal into octave frequency bands, using band-pass finite impulse response (FIR) filters, which are non-linearly amplified based on the patient's prescription. From the prescription three parameters are determined which control the amplification. These parameters,  $A$ ,  $B$ , and  $p$ , can be seen in equation 1.1.

$$y = \begin{cases} Ax & \text{when } |x| \leq t \\ Bx^p & \text{when } |x| > t \end{cases} \quad (1.1)$$

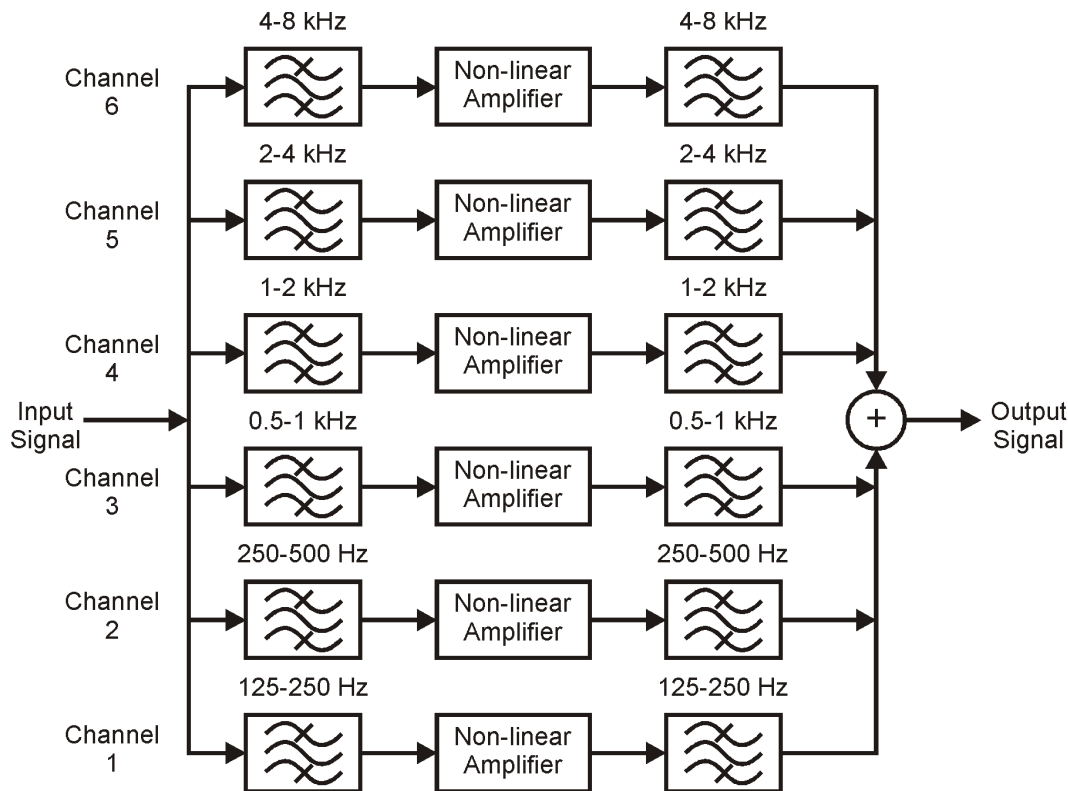


Figure 1.8: Unirate hearing aid architecture.

With  $p \leq \frac{1}{2}$ , when the input signal,  $x$ , is below the threshold,  $t$ , it is linearly amplified with gain  $A$  into the output signal,  $y$ . When the input signal is above the threshold,  $t$ , it is compressed using a power law [14]. This amplification adds odd high order harmonics to the signal which must be filtered out with the second bank of band-pass FIR filters which are identical to the first bank. The outputs of the second set of filters are added together to be converted back into sound. Each of the channels in this design run at the same rate [12] so to keep the audio quality from being degraded each lower channel FIR filters has twice as many taps as the filters in the channel above.

The current plan of the hearing aid is a multi-rate design [13] that decreases the power requirements for frequency band separation. Each channel can run at half

the rate of the channel above due to down-sampling of the input data. With only half the data to process, each lower channel consumes much less power than the channel above. Figure 1.9 shows the block diagram for the design and the blocks are described below.

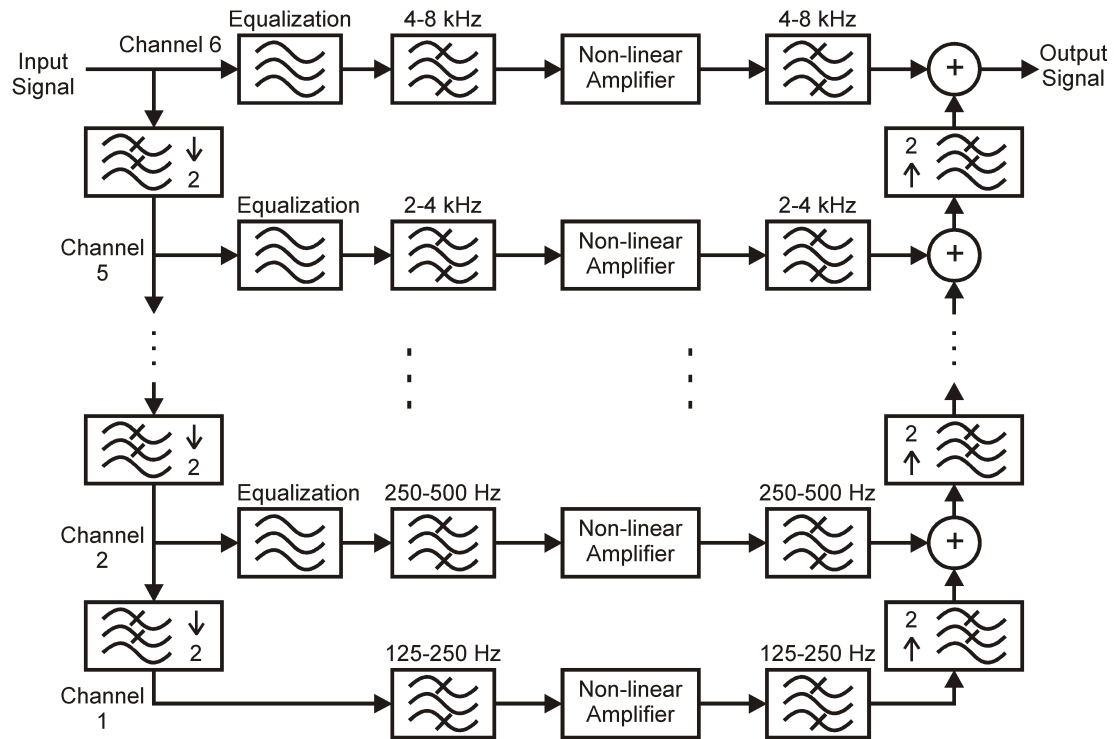


Figure 1.9: Multirate hearing aid architecture.

**Down-sampler and Low Pass Filter** Except for the uppermost channel the input signal is filtered by a finite impulse response (FIR) low-pass filter and down-sampled to allow each lower channel to run at half the sample rate of the channel above it.

**Equalization** In all channels except for the lowest, the equalization all-pass filter compensates for the group delay of the lower channels.

**FIR Filter** The first bank of band-pass FIR filters restricts the input signal to an octave frequency band for amplification.

**Non-Linear Amplifier** The non-linear amplifier performs the main function of the hearing aid. Based on the patient's prescription the different frequency bands can be amplified different amounts to compensate for specific hearing loss.

**Second FIR Filter** The second band-pass FIR filter is identical to the first in each channel. Its purpose is to remove any upper order harmonics introduced by the non-linear amplification.

**Up-sampler and Low Pass Filter** The second low-pass filter up-samples the signal for adding to the upper channels and then low pass filters to remove high frequency noise added by the up-sampling process.

This thesis will focus on the power consumed in the filters, which all have similar designs. More specifically in the 4-8 kHz band-pass filter, the multiply-accumulate function that claims the majority of the power consumed in the filter is explicitly measured and used as a model for the other filters. The impulse response of the filter is shown in Figure 1.10 and the frequency response is shown in Figure 1.11. The power consumed by the non-linear amplifiers and the channel as a whole is also being studied [7].

## 1.4 Related Work

Others have investigated the use of custom numerical representations for a variety of applications [19] and [6] including K-means clustering for images and video signal processing. These works focused on signal quality while this thesis examines the

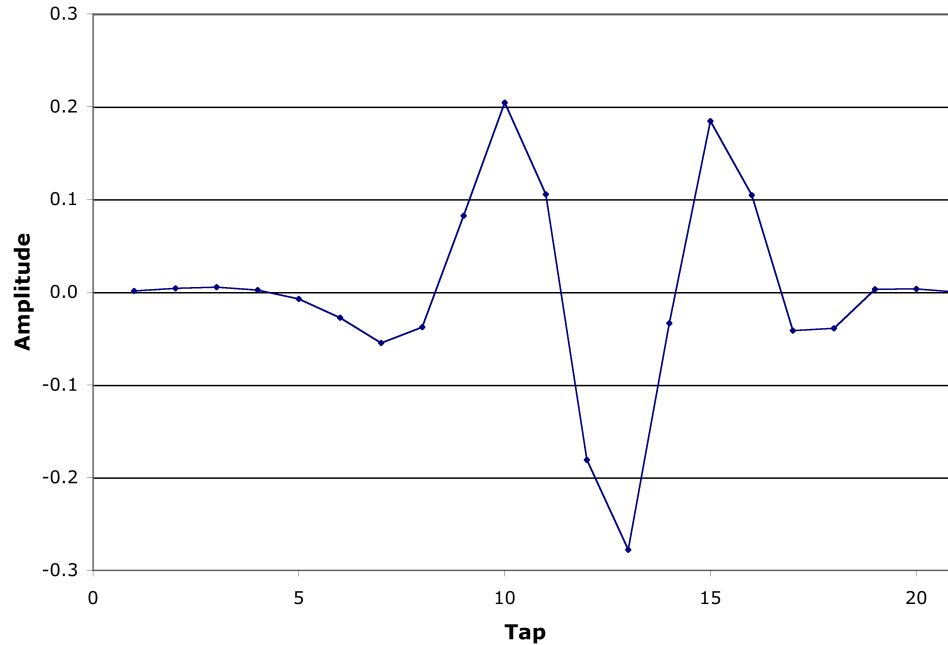


Figure 1.10: Impulse response of the 4-8 kHz band-pass FIR filter.

power consumption implications of the customization. Shang [17] examines estimating power consumption in Virtex<sup>TM</sup>-II FPGAs. They conclude that on average the dynamic power dissipation of a control logic block is  $5.9 \mu\text{W}$  per MHz. However, there is no connection to signal transition count as is made in this thesis, just clock frequency.

This investigation is based upon previous work with the logarithmic representation done by Robert Morley and his graduate students [4, 26, 23].

**Logarithmic Representation** Engel [4] discusses logarithmic encoding as an alternative to  $\mu$ -law and A-law compressive encoding. He finds that performance of a logarithmic encoder is quite similar to that shown by the compressive encoders. Through the examination of logarithmic representations Engel concludes that a 9-bit representation with a base of .9412 will provide the desired characteristics for audio



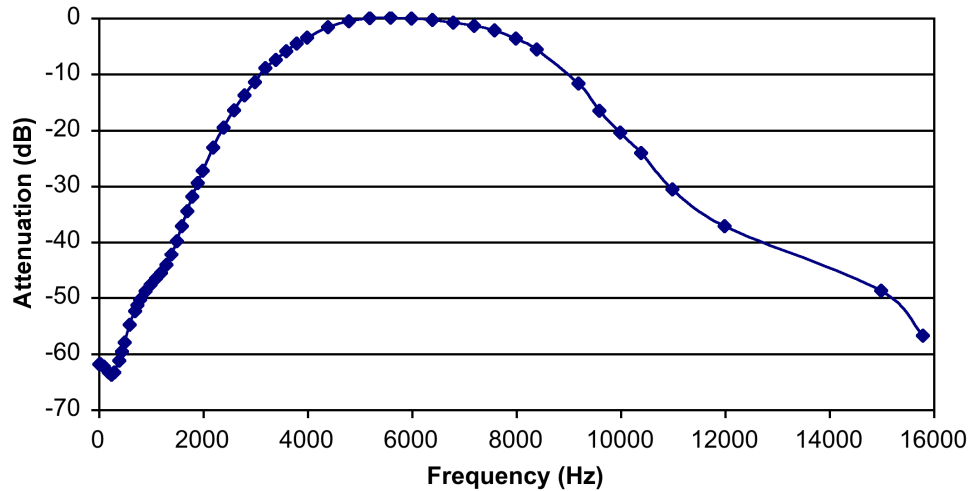


Figure 1.11: Frequency response of the 4-8 kHz band-pass FIR filter.

signals. Additionally, the majority of a logarithmic digital hearing aid was fabricated, and noise characteristics and power consumption was measured.

**Power Estimation** Sullivan [26] examines estimating power consumption for VLSI (very large scale integration) DSP (digital signal processor) design. Power consumption is computed using  $P_{cmos} = P_{dynamic} + P_{static}$ .  $P_{dynamic}$  comes from charging and discharging nodes within a circuit.  $P_{static}$  is consumed primarily by leakage currents through off transistors. Sullivan uses a simple adder circuit to test this form of power estimation. Then the building blocks of a DSP are broken down and a power estimate is derived. From these estimates it is shown that a Baugh-Wooley multiplier is more power efficient than comparable shift and add multipliers. Other parts examined include memory, bus, controller, and interconnects. This work also compares systems using a linear representation with those using a logarithmic representation. The logarithmic representation is found to use only 30% of the power necessary for the linear representation.

**Logarithmic Analog to Digital Conversion** Kwa [23] shows a method for building an analog to digital (A/D) converter that outputs in the logarithmic representation. He uses delta-sigma converters and compares power savings of a logarithmic converter to an equivalent second order linear converter. The logarithmic design provides 30 dB of SQNR over a 57 dB dynamic range by embedding the logarithmic transformation in a digital decimation filter. While this dynamic range isn't ideal for hearing aid design it shows that a direct to logarithmic A/D is possible. This examination finds over a factor of two power savings in the logarithmic filter versus the equivalent linear A/D. This savings is a "direct result of saving of the computational-power per tap as well as the reduction of the number of taps in the second-stage, FIR, decimation filter." [23]

## 1.5 Contributions

The following specifies the contributions of the author's work in this thesis.

- The original multiply-accumulate (MAC) operation for an FIR band-pass filter implemented on the FPX [10]. This was the basis for the Perl script that can now produce any of the floating point representations.
- The rewritten linear MAC with a Baugh-Wooley multiplier. This replaced a more naive MAC provided by an EE 563 group that wrote the original linear and logarithmic MACs.
- A new FIR controller for the various MACs for integration into a complete channel. This replaced the original FIR controller that interfaced with the FPX.

- Perl programs to produce random inputs for the various numerical representations.
- Perl programs to translate input files and coefficients between the various numerical representations.
- Perl golden models, to emulate the hardware, for verification of the linear and logarithmic representations. These perform the same function as the hardware in software. The two output files can then be compared to confirm the correctness of the hardware output and thus the circuit.
- An integrated circuit layout of the 1-5-5 MAC.
- Both the continuous and discrete simulations along with shell scripts to automate the process of running all the various combinations of MAC, target, and input set for the discrete simulator.
- Perl script to parse and sum signal transition count data from ModelSim output.

## 1.6 Organization

After describing why power consumption is vital to improving hearing aid performance in the areas with the most need, such as speech in noise, this chapter described the basic functioning of the ear and the issues that new hearing aids must be designed to compensate for. Then basic hearing aid architecture was discussed with specific emphasis on the new developments.

Chapter 2 will discuss the various numerical representations (linear, logarithmic, and 10 different floating point) that will be compared in terms of power consumption. This chapter will also compare the representations in terms of signal to quantization noise ratio and dynamic range.

In chapter 3 this thesis will show the block diagrams and describe the working of the FIR filters and the MACs for each representation. This chapter also discusses the input sets and verification of the output for the various circuits.

Chapter 4 shows the relationship between power consumption and signal transition counts used to compare the different representations based on a discrete-event simulation of the circuit and a continuous simulation. Then the different representations are compared using two different input sets based on signal transition counts from the discrete model. These relationships lead to some conclusions that are reported in Chapter 5 along with directions for possible future work.

## 1.7 Acronyms

- ASIC - Application Specific Integrated Circuit
- ATM - Asynchronous Transfer Mode
- CLB - Control Logic Block
- CMOS - Complementary Metal Oxide Semiconductor
- dB - deciBel
- DSP - Digital Signal Processor
- FIR - Finite Impulse Response

- FPGA - Field Programmable Gate Array
- FPX - Field-programmable Port Extender
- MAC - Multiply Accumulate Unit
- SNR - Signal to Noise Ratio
- SPL - Sound Pressure Level
- SQNR - Signal to Quantization Noise Ratio
- VHDL - VHSIC Hardware Design Language
- VHSIC - Very High Speed Integrated Circuit
- VLSI - Very Large Scale Integration
- VI - Virtual Instrument
- WUGS - Washington University Gigabit Switch

## Chapter 2

# Numerical Representations

There are three basic categories of numerical representations that this thesis will examine; a linear representation, a logarithmic representation, and several floating point representations. Each of the numerical representations is designed to represent signals that can be normalized to the range -1 to 1. In each case the representation will be carried throughout the design. As shown in Figure 2.1, an analog to digital converter would digitize the incoming audio signal directly into the proper representation and all internal signals would be in that representation until the final digital to analog converter takes the representation and converts it back to analog form. This eliminates power consumption due to conversions between representations. The ability to produce the representation directly from the analog to digital conversion is assumed to exist and is beyond the scope of this thesis. This assumption is supported by the work discussed in the introduction [23]. Throughout this chapter the numbers -.6768, .0143, .5574, and .9826 will be converted into the appropriate representations as examples.

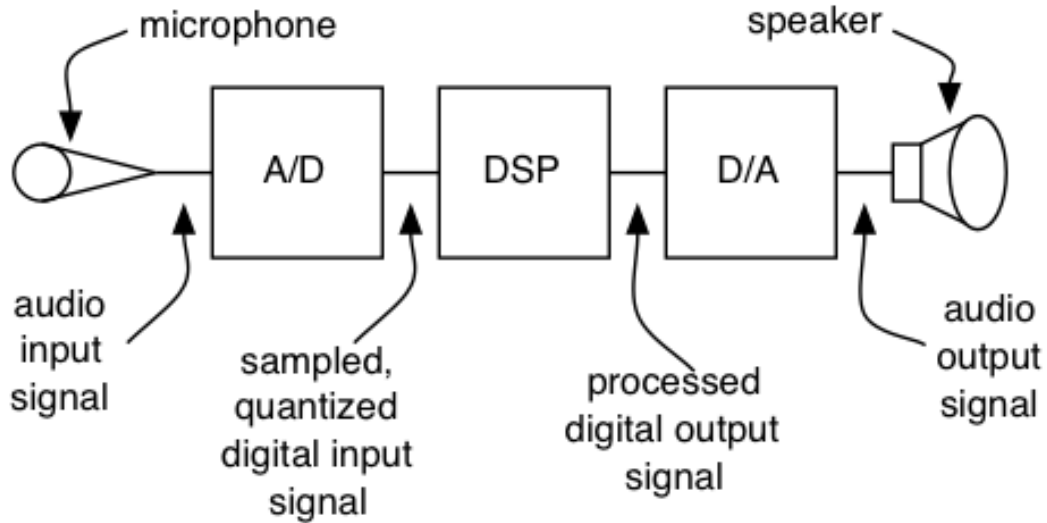


Figure 2.1: Block diagram of the hearing aid from microphone to speaker.

## 2.1 Linear Representation

The first representation that we will consider is a standard 16-bit two's complement representation. This representation is also called Q0.15. The 0 tells the number of integer bits and the 15 is the number of fractional bits. This form is often shortened to Q15 format with the 0 bits of integer understood. The input values range from  $-1$  up to  $1 - (2^{-15})$  with a constant step size of  $2^{-15}$ . This representation has been used extensively in digital signal processing applications and audio applications such as standard red book audio CDs [3]. Values in the linear representation can be computed from the binary representation to a real number by Equation 2.1.

$$x = -a_{15}2^0 + \sum_{i=0}^{14} a_i \cdot 2^{i-15} \quad (2.1)$$

**Linear Examples** For each conversion from a real number to binary number we start with a string of 16 bits:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$a_{15}$	$a_{14}$	$a_{13}$	$a_{12}$	$a_{11}$	$a_{10}$	$a_9$	$a_8$	$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$

where bit  $a_{14}$  has a value  $\frac{1}{2} = 2^{-1}$ , bit  $a_{13} = \frac{1}{4} = 2^{-2}$ , bit  $a_{12} = \frac{1}{8} = 2^{-3}$ , and so on down to bit  $a_0 = \frac{1}{2^{15}} = 2^{-15}$ . For  $-.6768$  we start by observing that it is negative and working with the absolute value. Now we have  $.6768$ .  $\frac{1}{2} \cdot \frac{1}{2^{15}}$  is added to account for any rounding issues so we now have  $.6768305177578$ . This is larger than  $2^{-1}$  so bit  $a_{14}$  gets set to 1 and  $2^{-1}$  is subtracted from the input. That leaves  $.1678305177578$  which is compared to  $2^{-2}$ . Since  $2^{-2}$  is larger, bit  $a_{13}$  is set to 0. The input is next compared to  $2^{-3}$  and is larger. Bit  $a_{12}$  becomes a 1 and  $2^{-3}$  is subtracted. The input is now  $.0518305177578$ . As  $2^{-4}$  is larger than the input bit  $a_{11}$  becomes a zero.  $2^{-5}$  is larger then the input so bit  $a_{10}$  is set to 1 and  $2^{-5}$  is subtracted to leave  $.0205805177578$ . Bit  $a_9$  is set to 1 as well since  $2^{-6}$  is smaller than the input. After the subtraction the input becomes  $.0049555177578$ . Bit  $a_8$  is set to 0 and bit  $a_7$  is made to be 1 and then the input is reduced again to  $.001049267578$ . Next bit  $a_6$  is set to 0 and bit  $a_5$  to a 1 and then input is again reduced, this time to  $.000072705078$ . The next three bits become 0 and bit  $a_0$  becomes a 1 and the final reduction in the input leaves  $.000011669922$ , which is smaller than  $2^{-15}$ . This process produces the binary number:

0	1	0	1	0	1	1	0	1	0	1	0	0	0	0	1
$a_{15}$	$a_{14}$	$a_{13}$	$a_{12}$	$a_{11}$	$a_{10}$	$a_9$	$a_8$	$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$

Remember that the initial input was negative. To take this into account the NOT of the binary number is taken and 1 added. Notice that bit  $a_{15}$  becomes a 1 signifying the sign of the number.

1	0	1	0	1	0	0	1	0	1	0	1	1	1	1	1
$a_{15}$	$a_{14}$	$a_{13}$	$a_{12}$	$a_{11}$	$a_{10}$	$a_9$	$a_8$	$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$



Table 2.1: Linear representation examples

$x$	$x$
-.6768	1010100101011111
.0143	0000000111010101
.5574	0100011101011001
.9826	0111110111000110

As an example of the reverse operation 0000000111010101 is converted to .0143 with Equation 2.1.

$$\begin{aligned}
 x &= -0 \cdot 2^0 + 0 \cdot 2^{14-15} + 0 \cdot 2^{13-15} + 0 \cdot 2^{12-15} + 0 \cdot 2^{11-15} + 0 \cdot 2^{10-15} + 0 \cdot 2^{9-15} \\
 &\quad + 1 \cdot 2^{8-15} + 1 \cdot 2^{7-15} + 1 \cdot 2^{6-15} + 0 \cdot 2^{5-15} + 1 \cdot 2^{4-15} + 0 \cdot 2^{3-15} \\
 &\quad + 1 \cdot 2^{2-15} + 0 \cdot 2^{1-15} + 1 \cdot 2^{0-15} = .014312744141
 \end{aligned}$$

The result of this, .014312744141, is the closest value to .0143 in this representation. The next smaller representable value would be .014282226563 which is slightly farther from .0143 than the represented value. Table 2.1 shows the bit representation for each of the four examples.

## 2.2 Logarithmic Representation

Looking at a logarithmic representation stems from the logarithmic perception response of the human ear. The 9-bit logarithmic representation has a base of .941 as suggested from previous research [4]. This representation is sign-magnitude with one sign bit in the most significant place and 8 magnitude bits following. Throughout this thesis logarithmic values will be denoted with a subscript  $l$  as in  $x_l$  such that  $x_l = \log_{.941}(x)$ . This representation allows signal values to range from -1 up to 1. It

is important to note that with a base of .941 if  $x > y$  in linear space, then  $x_l < y_l$  in logarithmic space. For example the maximum value, the number  $x = 1$ , is represented as  $x_l = 000000000$  while the minimum positive value,  $x = 1.8423 \cdot 10^{-7}$ , is represented as  $x_l = 011111111$ . On the other side of zero, the smallest magnitude negative number,  $x = -1.8423 \cdot 10^{-7}$ , is written as  $x_l = 111111111$  while the largest magnitude negative number,  $x = -1$ , is written as  $x_l = 100000000$ . This relation can be seen in Figure 2.2, while Figure 2.3 shows the same relationship with the linear values plotted on a log scale. In this thesis when discussing relative comparisons of numbers we will always use the linear values. Values in the logarithmic representation can be computed from the binary representation to a real number by Equation 2.2.

$$x_l = (-1)^{a_8} \cdot (.941)^{\sum_{i=0}^7 a_i \cdot 2^i} \quad (2.2)$$

**Logarithmic Examples** Each number is a string of 9 bits:

$$\begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_8 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{array}$$

The easiest way to convert into the logarithmic representation is to first convert to an integer and into binary from there. This is done by taking the  $\log_{.941}$  of the input. In the script this is done by  $\frac{\log_{10}(input)}{\log_{10}(.941)}$ . Since the first example, -.6768, is negative bit  $a_8$  is set to 1. Then the absolute value is taken to give .6768 again. Now log of this value is taken;  $\log_{.941}(.6768) = 6.41 \dots$  This value is rounded to 6 and converted to binary in a process very similar to that in the linear script. However, now the bits have the values 1, 2, 4, 8,  $\dots$ , 128 from right to left. This gives the logarithmic, binary representation of .6768 as:

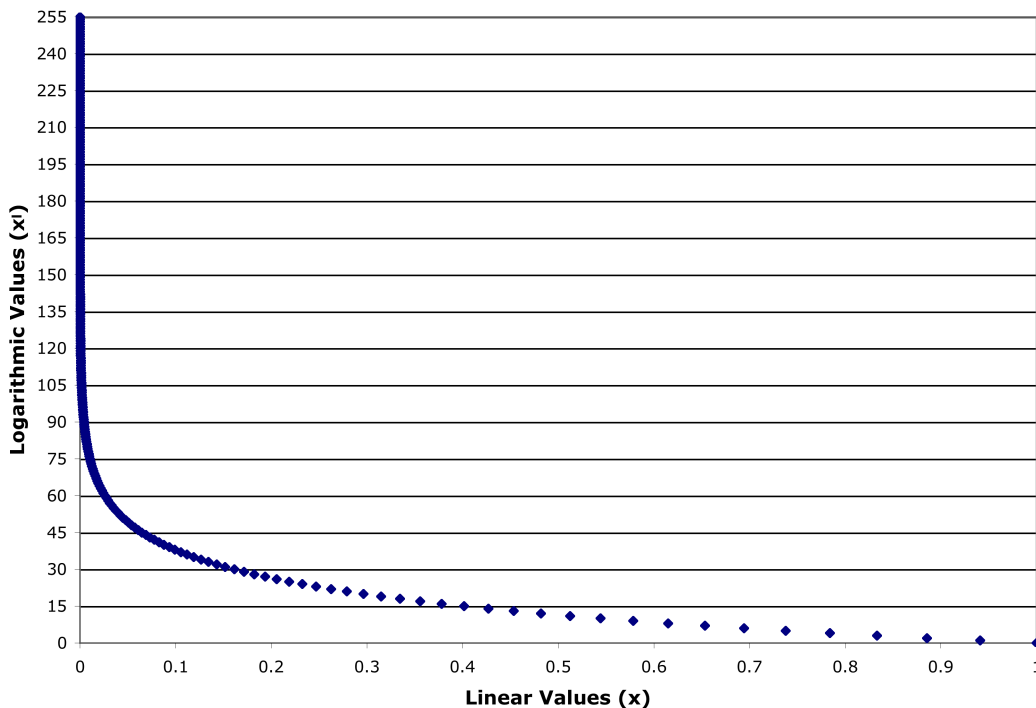


Figure 2.2: A comparison of linear inputs and their logarithmic equivalents.

1	0	0	0	0	0	1	1	0
$a_8$	$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$

Table 2.2: Logarithmic representation examples

$x$	$x_l$
-.6768	100000110
.0143	001000110
.5574	000001010
.9826	000000000

As an example of the reverse operation 001000110 is converted to .0143 with Equation 2.2.

$$-1^0 \cdot .941^{(0 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0)} = .014166861232$$

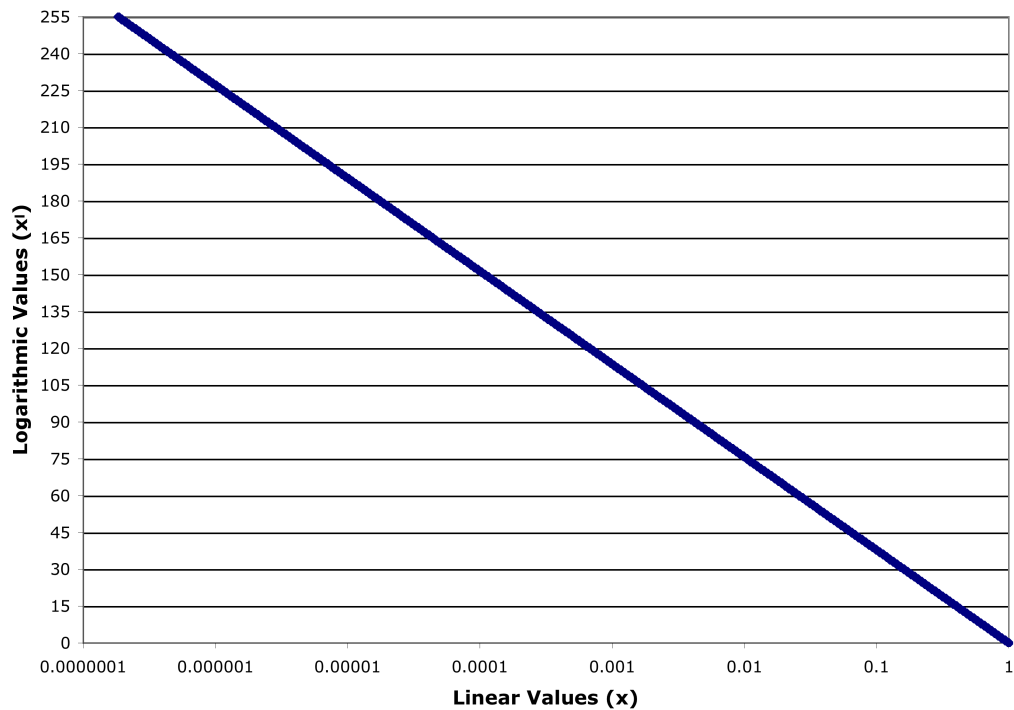


Figure 2.3: A comparison of linear inputs plotted on a log scale to their logarithmic equivalents.

The result of this, .014166861232, is the closest value to .0143 representable by this representation. Table 2.2 shows the logarithmic bit representations for each of the four examples.

Notice that as the linear, positive values increased, the binary representation decreased. In fact, .9826 is so close to one that it appears to be zero in binary. Figure 2.4 shows a close up of  $x = 1$  from Figure 2.2. .9826 is within the error bar of 1 and therefore is represented at 1 which in binary logarithmic form is 000000000.

Figure 2.5 shows the other end of the Figure 2.2. Notice that each logarithmic value has a much smaller span on the linear axis. Note, also, the fact that the smallest representable value is not identically zero.

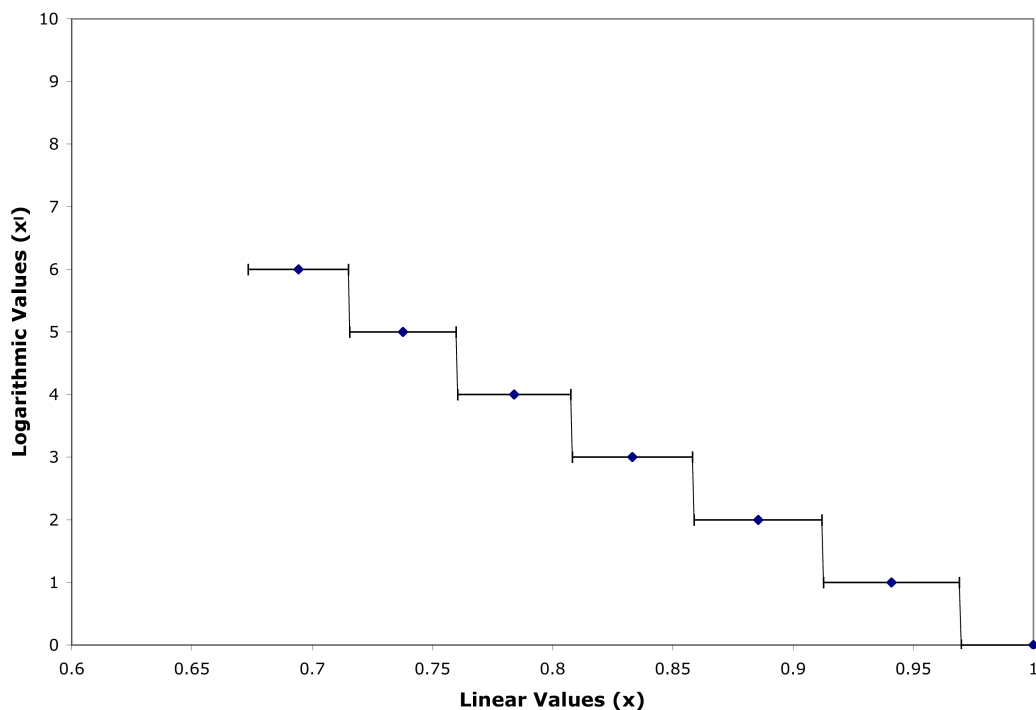


Figure 2.4: Close up of  $x = 1$  on linear vs logarithmic comparison.

## 2.3 Floating Point Representations

Floating point numbers have some characteristics of both linear and logarithmic representations as will be shown below. The mantissas act like a linear part and the exponents act like a logarithmic part. The values of these numbers aren't evenly distributed along the real line, due to the logarithmic nature of the exponents and the gaps between ranges of mantissa values. The IEEE floating point standard defines a single precision 32-bit number with one sign bit, eight exponent bits, and 23 fractional bits. The exponent is in excess notation with a bias of 127; i.e. if the exponent shows 0 then it represents  $-127$  and if the exponent shows 200 then it represents  $200-127 = 73$ . Since the mantissa is normalized it actually has 24 bits worth of data as an

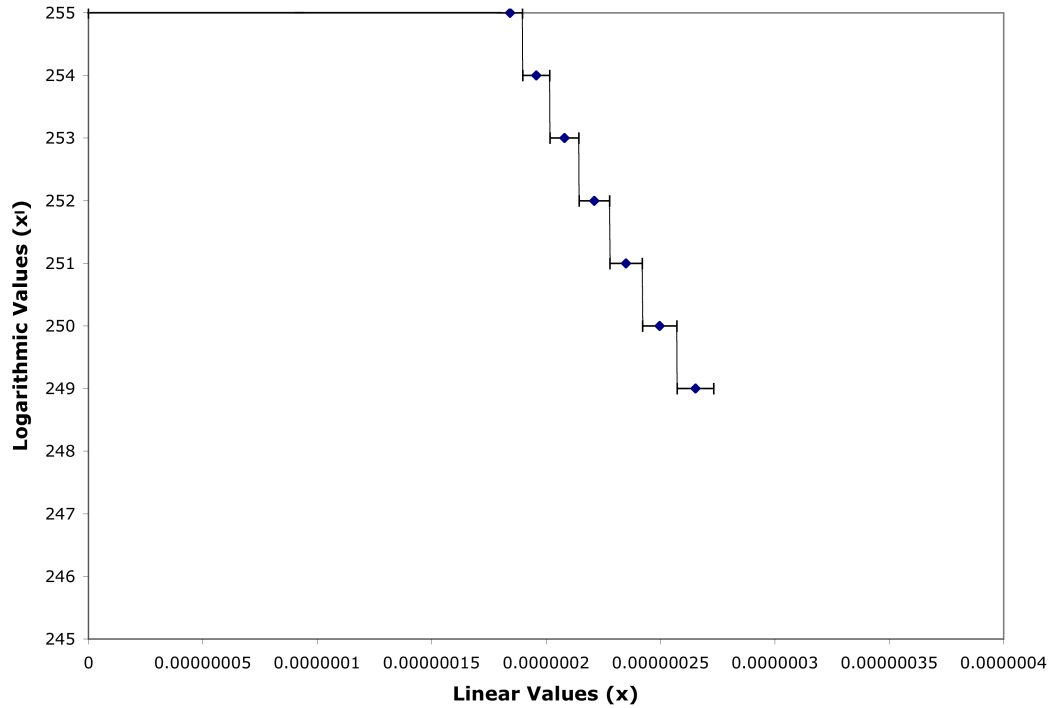


Figure 2.5: Close up of  $x$  close to 0 on linear vs logarithmic comparison.

implicit 1 is always assumed to be the first bit, but isn't stored. There are special cases defined for non-normalized mantissas, zeros, and infinities.

We investigated 10 different floating point representations. Nine of these were of the form 1 sign bit,  $E$  exponent bits,  $M$  mantissa bits and named as 1- $E$ - $M$  so that a 1-4-5 representation would have 1 sign bit, 4 exponent bits, and 5 mantissa bits. The representations considered range from 1-4-4 up to 1-6-6 with every combination of 4, 5, and 6 bit exponents and mantissas. The mantissa is a sign-magnitude value in conjunction with the sign bit. This design only supports normalized mantissas to simplify the hardware and save on computation by eliminating the comparisons that would be necessary to handle the special cases. This design also stores the leading one of the mantissa since we aren't concerned with storage space, and the bits would

need to be added back in for computation. The exponent is in excess notation based on the number of bits in the exponent.

To calculate the decimal value of the number we can use the formula

$$x_f = \frac{(-1)^{sign} \cdot \frac{mant}{2^M} \cdot 2^{exp-2^{E-1}}}{k} \quad (2.3)$$

where  $k$  (see equation 2.4) scales the input to be between -1 and 1.

$$k = \frac{2^M - 1}{2^M} \cdot 2^{(2^E - 1) - 2^{E-1}} \quad (2.4)$$

The values of  $mant$  and  $exp$  are the integer values of the appropriate bits, as seen in Equations 2.5 and 2.6 respectively, and  $E$  and  $M$  are the number of bits of the exponent and mantissa respectively. In the design under test the scaling factor  $k$  was omitted in input set creation as the scale does not impact the calculations being performed. The value of sign is either 0 for non-negative numbers or 1 for negative numbers and is just the value of the sign bit.

$$mant = \sum_{i=0}^{M-1} m_i \cdot 2^i \quad (2.5)$$

$$exp = \sum_{i=0}^{E-1} e_i \cdot 2^i \quad (2.6)$$

**Floating Point Examples** Depending upon the representation chosen there can be anywhere from 9 to 13 bits present. For these examples the 1-4-5 representation will be used.

0	0	0	0	0	0	0	0	0	0
$s$	$e_3$	$e_2$	$e_1$	$e_0$	$m_4$	$m_3$	$m_2$	$m_1$	$m_0$
sign	exponent				mantissa				

Bit  $s$  must be made into a 1 since the input,  $-.6768$ , is negative. Then the absolute value is taken. We now have:

$$.6768 = \frac{\frac{mant}{2^M} \cdot 2^{exp-2^{E-1}}}{k}$$

Next, the scaling factor,  $k$ , for the representation must be calculated (see equation 2.4). For the 1-4-5 representation  $k = (2^5 - 1) \cdot 2^{(2^4-1)-2^{4-1}} = 124$ . Now our positive input is multiplied by  $k$  to give:

$$.6768 \cdot 124 = 83.9232 = \frac{mant}{2^5} \cdot 2^{exp-2^{4-1}}$$

To find the mantissa we must first find the largest exponent that is smaller than the scaled input value. So we want  $\frac{83.9232}{2^{exp-8}} < 1$ . The exponent value must be an integer between 0 and 15 inclusive. In this case the exponent must be 15, so bits  $e_3$  down to  $e_0$  are all set to 1. We now have  $83.9232 = \frac{mant}{32} \cdot 2^{15-8}$ . This leaves  $mant = 20.9808$ . However, the mantissa must also be an integer so it is rounded up to 21. Now the rest of the bits can be filled in to make 21, i.e. bits  $m_0$ ,  $m_2$ , and  $m_4$  are set to 1.

1	1	1	1	1	1	0	1	0	1
$s$	$e_3$	$e_2$	$e_1$	$e_0$	$m_4$	$m_3$	$m_2$	$m_1$	$m_0$
sign	exponent				mantissa				

Table 2.3: 1-4-5 floating point representation examples

$x$	$x_f$
-.6768	1111110101
.0143	0100111100
.5574	0111110001
.9826	0111111110



As an example of the reverse operation 0100111100 is converted back to .0143 with Equation 2.5, 2.6, and 2.3, and that  $k = 124$ .

$$mant = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 28$$

$$exp = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9$$

$$x_f = \frac{(-1)^0 \cdot \frac{28}{2^5} \cdot 2^{9-2^4-1}}{124} = .014112903226$$

The result of this, .014112903226, is the closest value to .0143 representable by this representation. The next closest value above .0143 would be .014616935484 in this representation. Table 2.3 shows the floating point bit representations for the four examples.

## 2.4 Revised Floating Point Representation

To try and realize the power benefits of a Baugh-Wooley multiplier [23] a second floating point representation was developed. This representation uses two's complement values for both the exponent and mantissa. This unites the sign bit with the mantissa and thus converts a 1-4-5 into a 4-6 representation.

The new exponent and mantissa values can be calculated as shown in equations 2.7 and 2.8, respectively.

$$exp = -e_3 2^3 + \sum_{i=0}^2 e_i \cdot 2^i \tag{2.7}$$

$$mant = -m_5 2^5 + \sum_{i=0}^4 m_i \cdot 2^i \tag{2.8}$$

Now the total value for a floating point number is:

$$x_f = \frac{\frac{mant}{2^{M-1}} \cdot 2^{exp}}{j} \quad (2.9)$$

Where  $j$  is slightly revised from the previous  $k$ . Again this factor was not used in the design under test.

$$j = \frac{2^{M-1} - 1}{2^{M-1}} \cdot 2^E - 1 \quad (2.10)$$

### Revised Floating Point Examples

0	0	0	0	0	0	0	0	0	0
$e_3$	$e_2$	$e_1$	$e_0$	$m_5$	$m_4$	$m_3$	$m_2$	$m_1$	$m_0$
exponent					mantissa				

First take note of the sign of the number  $-.6768$  for later compensation and then work with the absolute value,  $.6768$ . The scale factor in this case is  $j = \frac{2^{6-1}-1}{2^{6-1}} \cdot 2^{4-1} - 1 = 6.78125$  so we multiply  $.6768 \cdot 6.78125 = 4.58955$ . Again, we must find the largest integer exponent that is smaller than the input value so  $\frac{4.58955}{2^{exp}} < 1$ .  $exp$  must be 3 for this to be true. Now  $4.58955 = \frac{mant}{2^{6-1}} \cdot 2^3$  and, therefore,  $mant = 18.3582$  and is rounded to 18.

0	0	1	1	0	1	0	0	1	0
$e_3$	$e_2$	$e_1$	$e_0$	$m_5$	$m_4$	$m_3$	$m_2$	$m_1$	$m_0$
exponent					mantissa				

But remember that the original value was negative so the mantissa must be adjusted to reflect this.

0	0	1	1	1	0	1	1	1	0
$e_3$	$e_2$	$e_1$	$e_0$	$m_5$	$m_4$	$m_3$	$m_2$	$m_1$	$m_0$
exponent					mantissa				

So  $0011101110 = -.663594470046$ . The next larger value would be  $0011101111 = -.700460829493$ . Table 2.4 shows the revised floating point bit representations for the four examples.

Table 2.4: 4-6 revised floating point representation examples

$x$	$x_f$
-.6768	0011101110
.0143	1101011001
.5574	0010011110
.9826	0011011011

## 2.5 Representation Characteristics

### 2.5.1 Signal to Quantization Noise

Figures 2.6 through 2.10 show the the SQNR for the linear representation, the logarithmic, and the floating point representations with an exponent of four. All of these ratios except the logarithmic were calculated using equation 2.11 assuming a sinusoidal input with the quantization noise spread evenly between  $V_i$  and  $V_{i+1}$ .

$$SQNR = 20 \cdot \log_{10} \left( \frac{\frac{V_i}{\sqrt{2}}}{\frac{V_{i+1}-V_i}{\sqrt{12}}} \right) \quad (2.11)$$

For the logarithmic representation equation 2.12 is used to account for a dead zone between the smallest representable value and zero.

$$SQNR = 20 \cdot \log_{10} \left( \frac{\frac{V_i}{\sqrt{2}}}{\max \left( \frac{V_{i+1}-V_i}{\sqrt{12}}, v_{min} \right)} \right) \quad (2.12)$$

The standard 16 bit linear representation has a variable SQNR that ranges from 0 up to 96 dB over the dynamic range. Low power input signals can be easily

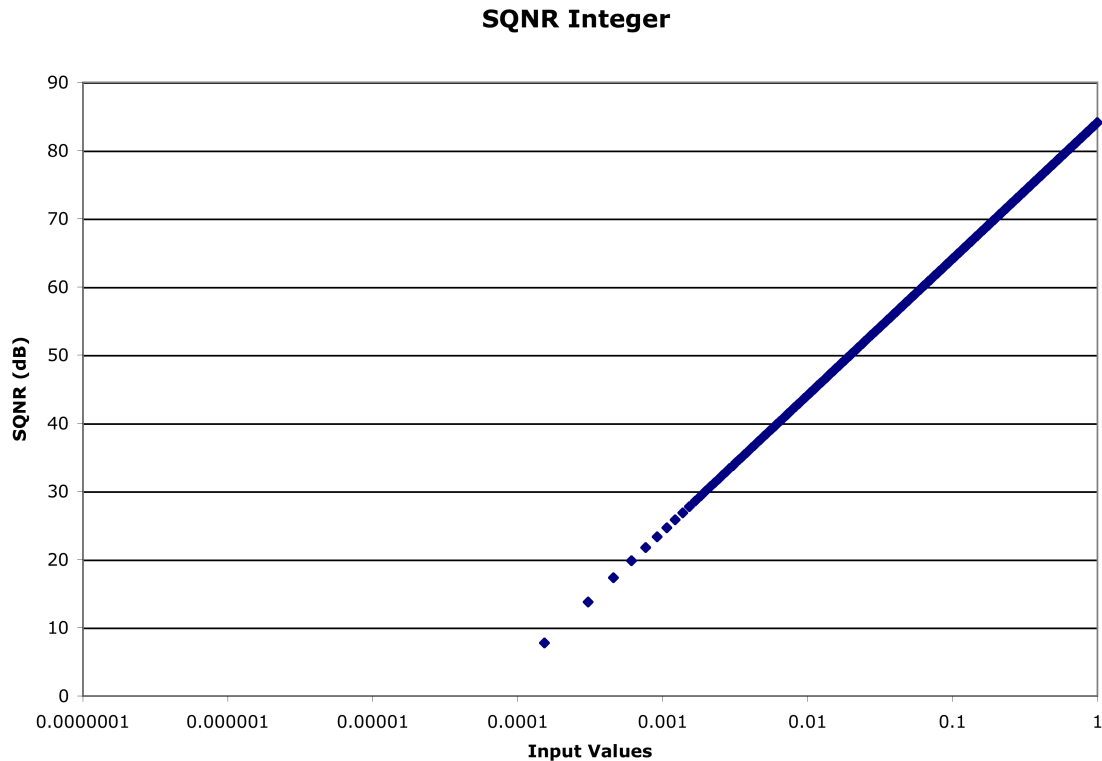


Figure 2.6: SQNR of Linear Representation.

lost in the noise with this SQNR characteristic. The logarithmic representation has an SQNR of  $\approx 33$  dB over most of the dynamic range. This represents a close match to the needs of human understanding of speech, and guided the choice of .941 as the base of the logarithm [4]. Figure 2.7 does not show the dead zone near zero where

The floating point representations with exponents of 5 and 6 have very similar SQNR to that of the representation with an exponent of 4 since the SQNR is based primarily on the mantissa value. Similarly, the revised 4-6 representation has the same SQNR as the 1-4-5 representation since it represents the same values as the 1-4-5 representation. More exponent bits just expand the dynamic range of the representation. This is shown quantitatively in the next section. We can see that the floating point SQNRs vary around a central value. This behavior is explained

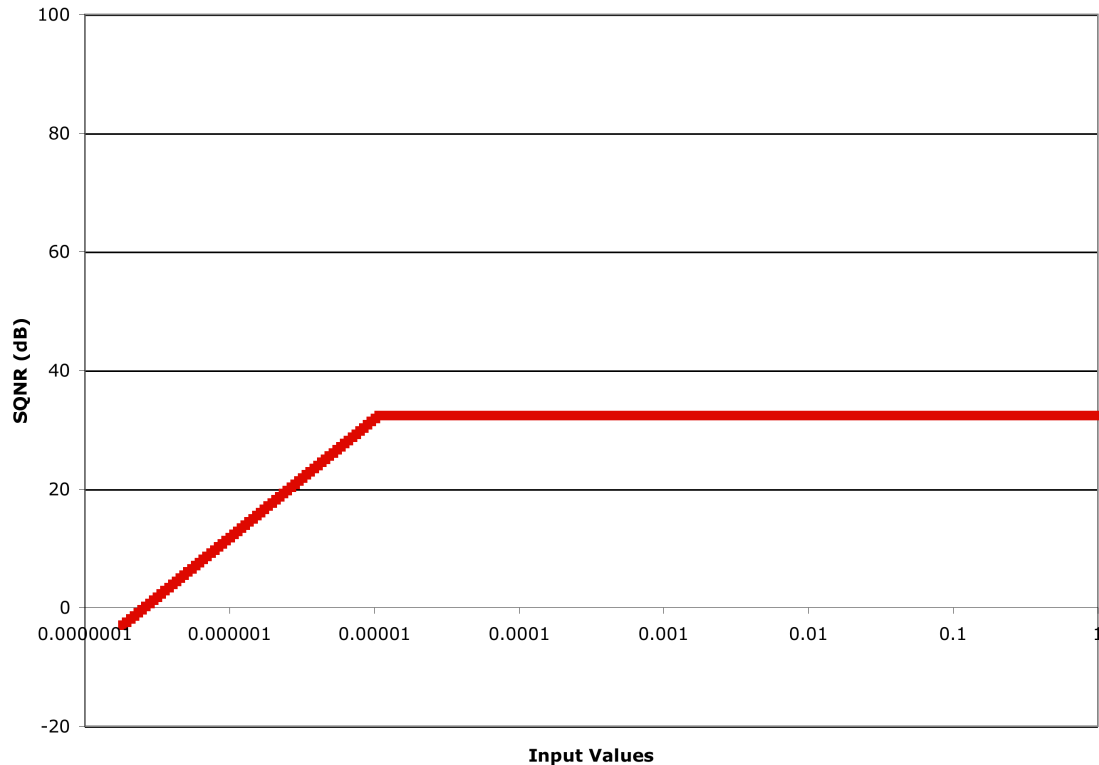


Figure 2.7: SQNR of Logarithmic Representation.

by the linear behavior of the mantissa and the logarithmic behavior of the exponent. Once the mantissa reaches the maximum value the exponent is incremented and the mantissa starts over. This causes the sawtooth effect seen in the figures.

### 2.5.2 Dynamic Range

The linear representation has a dynamic range of 96 dB which is right below the desired 100 dB. The ranges for the other representations can be seen in Figure 2.11. For the floating point representations the dynamic range is primarily dependent on exponent value. While the dynamic range rises slightly with increased mantissa range the large jumps between the groups is due to the exponent increase. Again the revised 4-6 floating point representation has the same properties as the 1-4-5 that it can replace.

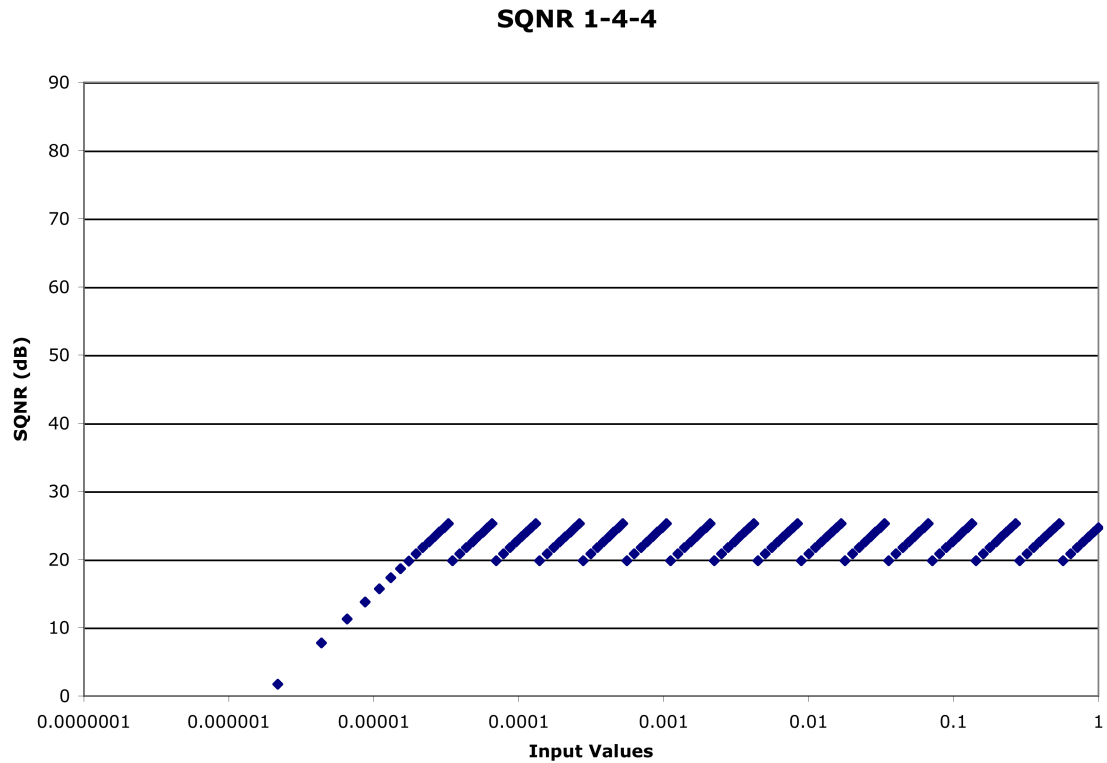


Figure 2.8: SQNR of 1-4-4 Representation.

As we can see from the figure all of the representations including the logarithmic have dynamic ranges that are as good or better than the linear representation.

These dynamic ranges were calculated with equation 2.13 where  $V_{max}$  is the maximum representable absolute value and  $V_{min}$  is the minimum non-zero representable absolute value.

$$\text{dynamic range} = 20 \cdot \log_{10} \left( \frac{V_{max}}{V_{min}} \right) \quad (2.13)$$

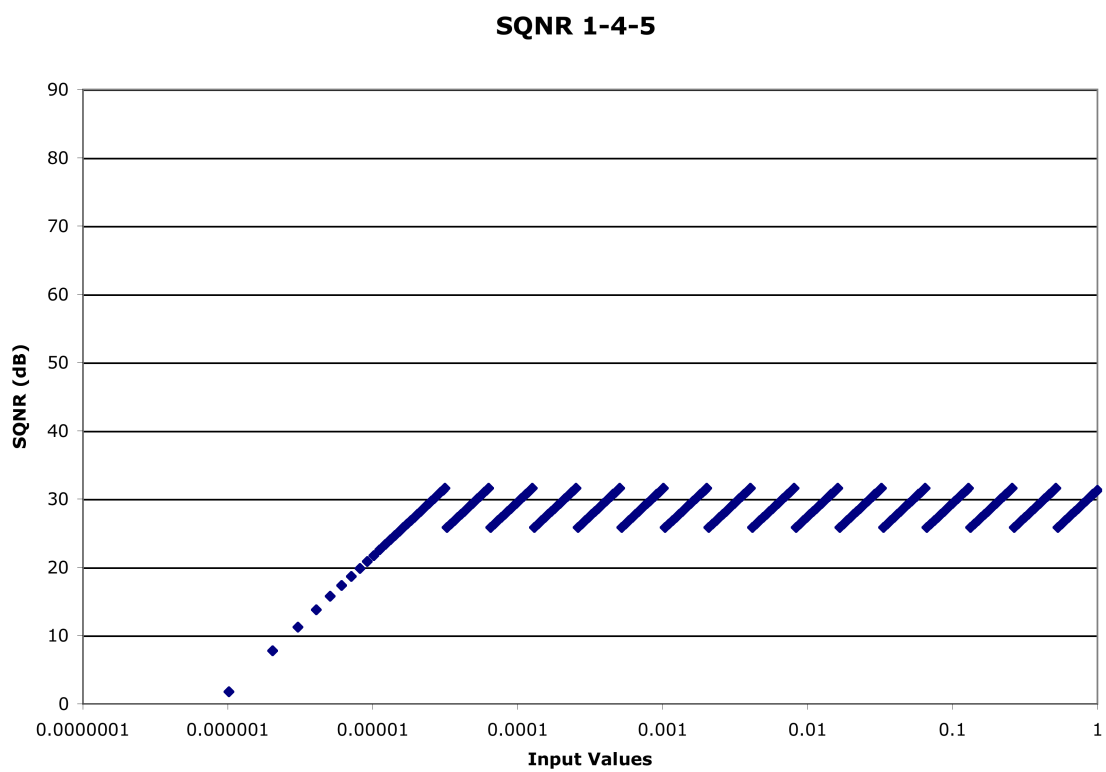


Figure 2.9: SQNR of 1-4-5 Representation.

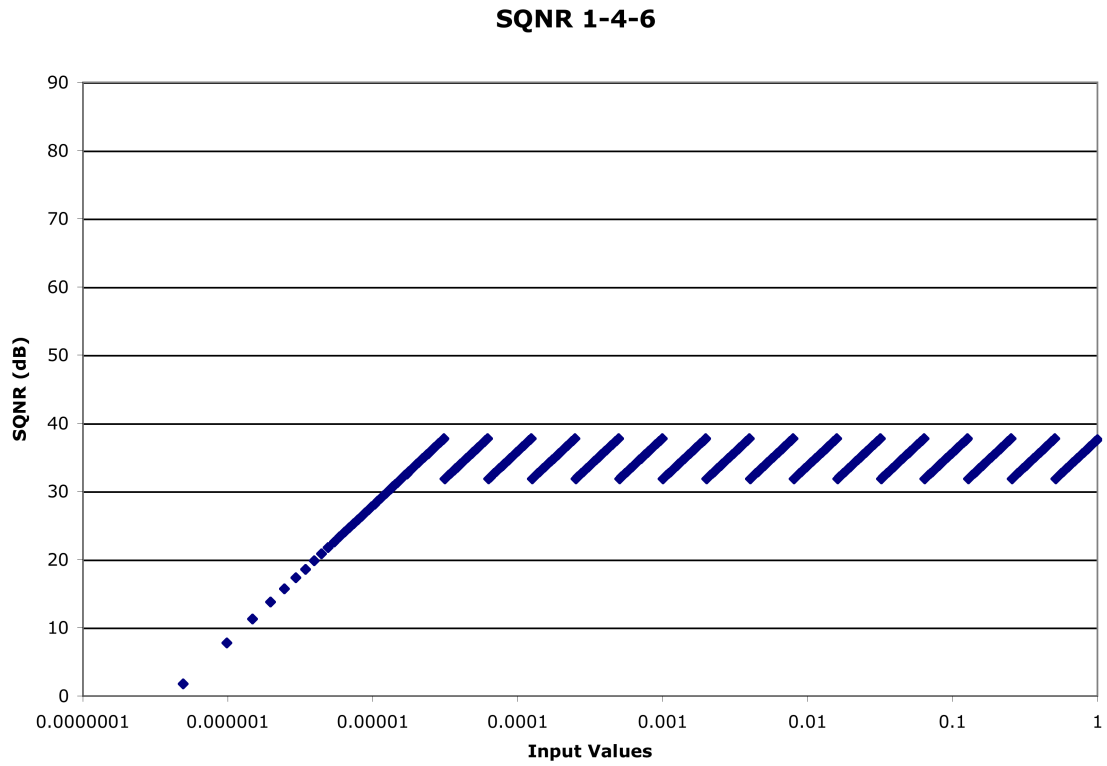


Figure 2.10: SQNR of 1-4-6 Representation.

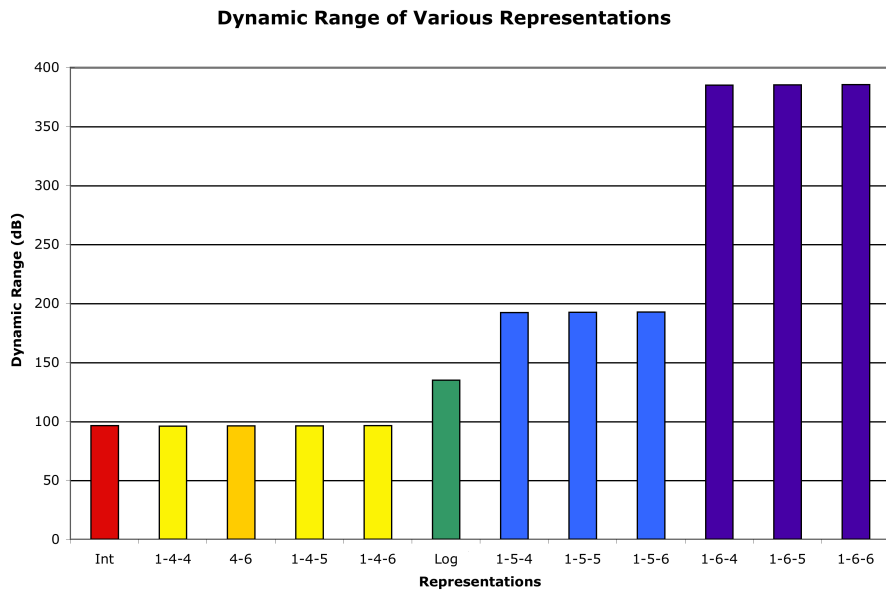


Figure 2.11: Dynamic range of various representations.



## Chapter 3

# Finite Impulse Response Filters

As shown in Figure 1.9, there are 28 filters in the multi-rate signal processing data flow. Related work has shown that the filters use the vast majority of the power consumed in the hearing aid [7], therefore, the remainder of this thesis will focus on these filters. The band pass and low pass filters are all 21 tap FIR filters which means that they all have 21 coefficients. All of the band-pass filters have the exact same design. The different frequencies passed are based on the sample rate at which the filters are run. By changing the coefficients the same basic design can be used for the low-pass filters as well. The equalizers can be implemented with circular buffers and are not considered further. These filters carry out the calculation shown in equation 3.1.

$$Y_j = \sum_{i=0}^{n-1} C_i \cdot X_{j-i} \quad (3.1)$$

where  $Y_j$  is the  $j^{th}$  output,  $C_i$  is the  $i^{th}$  coefficient, and  $X_{j-i}$  is the  $(j - i)^{th}$  last input sample and  $n$  is the number of taps. Each output value is a sum of the most recent input multiplied by the proper coefficient and the last 20 inputs multiplied by their respective coefficients. With each new input value the the oldest one is dropped

and the rest shifted to a new coefficient. The FIR filters are primarily multiply-accumulate (MAC) operators, so that is where the largest power savings might be found and where this thesis will focus.

### 3.1 Linear MAC

The linear MAC, shown in Figure 3.1, is a straightforward design. The Baugh-Wooley multiplier [2] has been shown to reduce power consumption [26] and is designed to streamline two's complement multiplies. This multiplier brings the sign calculations into a shift and add multiplier and can be built with fewer gates than comparable multipliers such as a Wallace Tree.

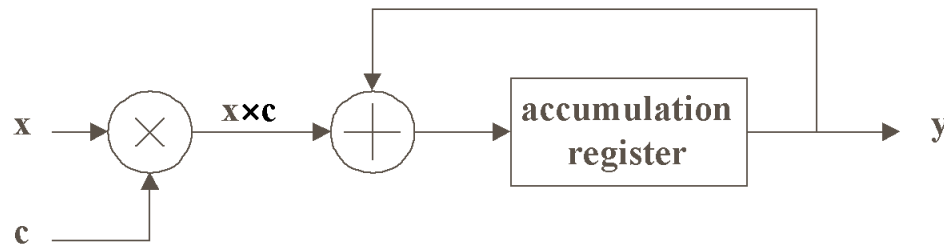


Figure 3.1: Block diagram of linear MAC.

#### 3.1.1 Baugh-Wooley Multiplier

A Baugh-Wooley multiplier takes two two's complement numbers as input, in this example the 5-bit numbers  $x = x_4x_3x_2x_1x_0$  and  $y = y_4y_3y_2y_1y_0$ . These numbers have the values

$$x = -2^4x_4 + \sum_{i=0}^3 x_i2^i$$

$$y = -2^4y_4 + \sum_{j=0}^3 y_j2^j$$

So now if  $p = x \cdot y$

$$\begin{aligned}
 p &= \left( -2^4 x_4 + \sum_{i=0}^3 x_i 2^i \right) \cdot \left( -2^4 y_4 + \sum_{j=0}^3 y_j 2^j \right) \\
 &= 2^8 x_4 y_4 + \sum_{i=0}^3 \sum_{j=0}^3 x_i y_j 2^{i+j} - 2^4 \sum_{i=0}^3 y_4 x_i 2^i - 2^4 \sum_{j=0}^3 x_4 y_j 2^j
 \end{aligned} \tag{3.2}$$

To avoid the subtraction in Equation 3.2, the two's complement of those terms is taken and added to the first two. Bitwise, with all of these additions, the final array can be seen in Table 3.1.

Table 3.1: Final Baugh-Wooley Array of Partial Products

	9	8	7	6	5	4	3	2	1	0
						$y_4$	$y_3$	$y_2$	$y_1$	$y_0$
*						$x_4$	$x_3$	$x_2$	$x_1$	$x_0$
							$x_0 y_3$	$x_0 y_2$	$x_0 y_1$	$x_0 y_0$
							$x_1 y_3$	$x_1 y_2$	$x_1 y_1$	$x_1 y_0$
					$x_2 y_3$	$x_2 y_2$	$x_2 y_1$	$x_2 y_0$		
		$x_4 y_4$		$x_3 y_3$	$x_3 y_2$	$x_3 y_1$	$x_3 y_0$			
		$\overline{x_4}$	$x_4 \overline{y_2}$	$x_4 \overline{y_2}$	$x_4 \overline{y_1}$	$x_4 \overline{y_0}$				
	1	$\overline{y_4}$	$y_4 \overline{x_3}$	$y_4 \overline{x_2}$	$y_4 \overline{x_1}$	$y_4 \overline{x_0}$				
						$x_4$				
+						$y_4$				

As the linear values are stored in two's complement form already, using the Baugh-Wooley multiplier is quite straightforward.

## 3.2 Logarithmic MAC

There are some basic facts about logarithmic math that suggest there might be significant power savings from its implementation. When dealing with logarithmic numbers what would be a multiplication in linear space becomes an addition. This is seen in

equation 3.3.

$$\log(X \cdot Y) = \log X + \log Y = X_l + Y_l \quad (3.3)$$

Since our power efficient Baugh-Wooley linear multiplier needs to do a 19 term addition as well as the 256 AND functions to find its value, reducing that to a single addition should save a large amount of power.

However, the linear addition becomes much harder in the logarithmic space. Equation 3.4 shows what calculations are necessary to perform the accumulate.

$$\log(X + Y) = \log\left(\frac{X + Y}{X}\right) + \log X = \log\left(1 + \frac{Y}{X}\right) + \log X \quad (3.4)$$

A lookup table is used to find the first term of the final sum in equation 3.4. The lookup table is indexed by the difference  $Y_l - X_l$  and produces the result of  $\log\left(1 + \frac{Y}{X}\right)$ . These values are related since  $\log\left(\frac{X}{Y}\right) = \log X - \log Y = X_l - Y_l$ .

As a further attempt to save power the lookup table can be bypassed when the result of the addition of the lookup table value and  $\log X$  doesn't depend on the lookup table value. This is the case if  $X \gg Y$ , and  $\log(X + Y)$  is within the quantization error of  $\log(X)$ . In the case of the hearing aid design, if  $Y_l - X_l > 56$  then the lookup table can be bypassed [20].

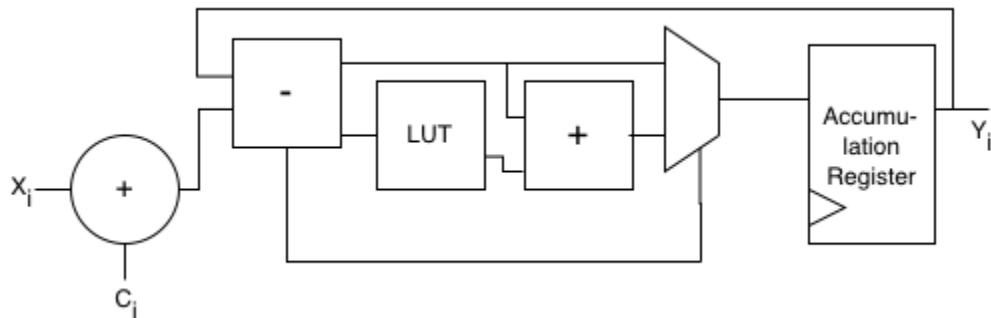


Figure 3.2: Block diagram of logarithmic MAC.

Figure 3.2 shows the block diagram of the logarithmic MAC. The input value and the coefficient are first added together. Then this value and the previous accumulated value are compared. If the lookup table is necessary the difference is passed to it and its output is added to the larger value. Otherwise the larger value is passed directly to the accumulation register. The multiplexor is controlled by the comparison.

### 3.3 Floating Point MAC

The block diagram in Figure 3.3 shows the main components of the original 9 floating point MAC designs. The figure is specific to a 1-5-5 design, but the only differences

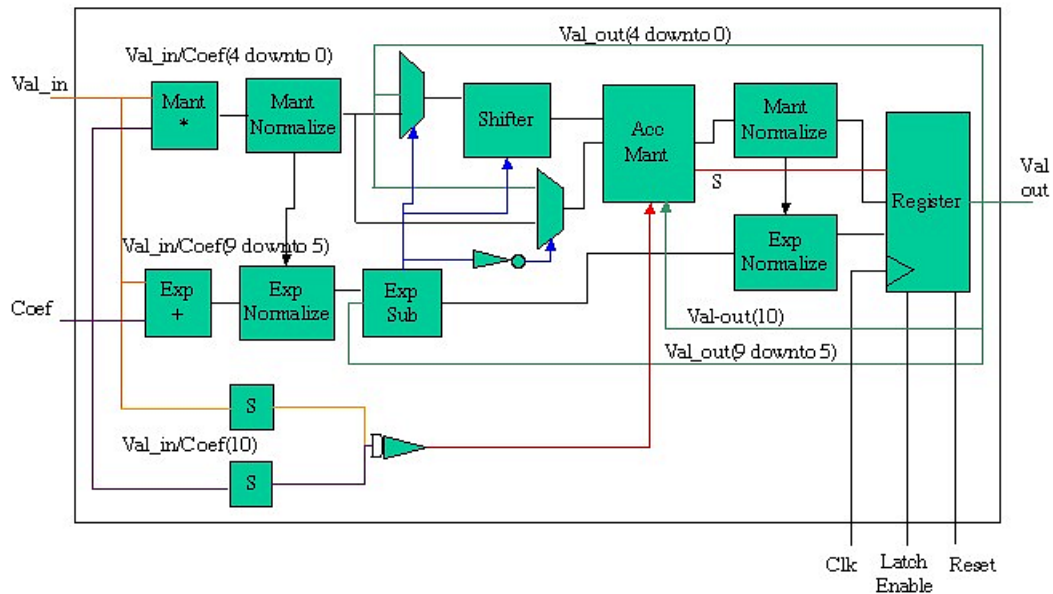


Figure 3.3: Block diagram of 1-5-5 MAC.

for the other representations are the widths of the signals and components.

For each incoming value/coefficient pair the numbers are first broken up into their sign, exponent, and mantissa pieces. The mantissas are multiplied and the

product is normalized. The exponents are added and shifted to compensate for the normalization. Meanwhile the sign bits are XOR'd to determine the sign of the result of the multiplication. Then the exponents from the multiplier and the previous accumulated value are compared to determine whether the product or the previous value need to be shifted for the accumulation. The smaller value is always shifted so that its exponent is equal to that of the larger exponent. Then the shifted mantissa and the other mantissa are added with the appropriate actions taken to account for sign to find the accumulated value. This mantissa is re-normalized if necessary along with any corresponding exponent shift that needs to occur. The new sign, exponent, and mantissa are then latched into the accumulation register to be used for the next accumulation. For an FIR filter, after the 21 multiply-accumulates are complete the accumulation register output is latched to an output register to be used by the next component in the hearing aid design.

### **3.4 Revised Floating Point MAC**

The basic function of the revised floating point MAC is the same as all of the MACs. Notice, in Figure 3.4, that the revised implementation uses a Baugh-Wooley multiplier [2] (like the linear representation) to multiply the mantissas. This also precludes the necessity of handling the sign values separately. Also the exponents now don't need the compensation for the excess notations in both values.

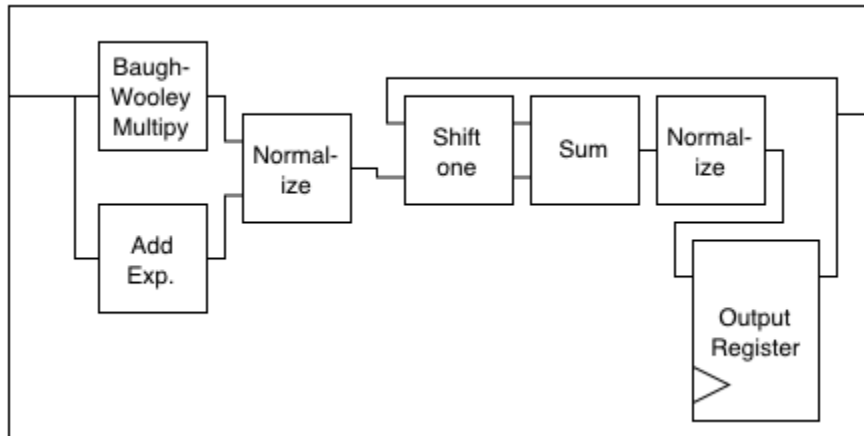


Figure 3.4: Revised floating point block diagram for two's complement representation.

## 3.5 Simulation Modeling

### 3.5.1 Input Sets

The first input set was a representative audio sample taken from the SPeech In Noise (SPIN) audiological test for human speech understandability [1, 16]. The SPIN test samples are an industry standard set of samples for testing hearing aid and patient's hearing abilities. This particular sample was a clean (i.e., no background noise) sample. The sample taken was about 3 seconds long and sampled at 32 kSamp/sec. This input set was not altered from the initial sampling so it does not cover the entire dynamic range of the representation. Normal conversation also does not cover the whole dynamic range of human ears so this limitation is justified. The second was a set of uniformly distributed, random data that spanned the entire dynamic range of the representation. The random data was generated to have the same number of samples as the 3 seconds of audio data. Figure 3.5 shows the speech data and Figure 3.6 shows the random data.

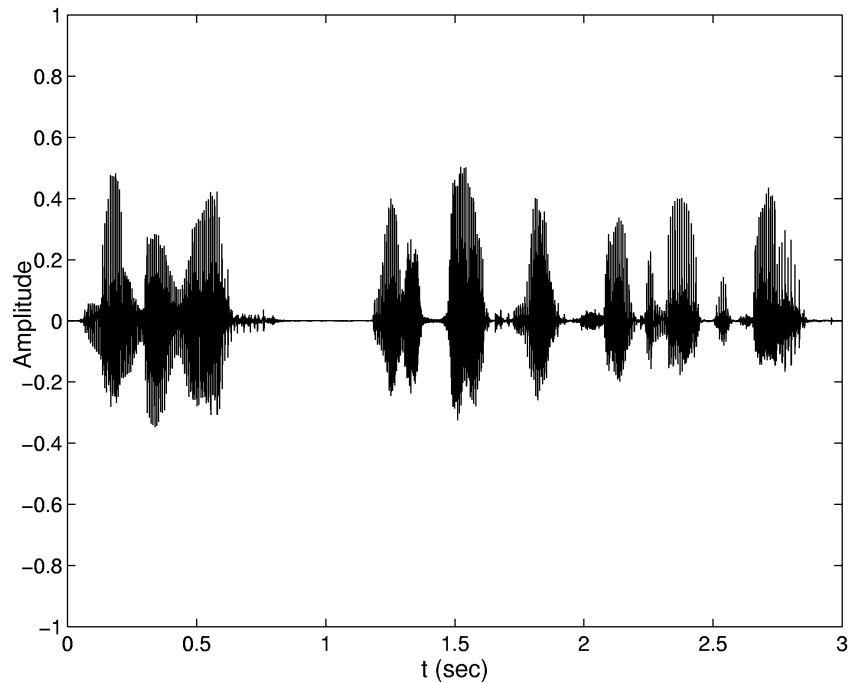


Figure 3.5: The speech input set.

The input samples started as wave files that contained decimal numbers. The initial conversion for the floating point representations was carried out by a LabVIEW virtual instrument that read in a wave file and output the data in the appropriate representation. The linear and logarithmic input was created from these files. In practice these conversions are carried out by Perl scripts that first account for the sign of the number and store the sign for later. Then the absolute value is taken to assure the input is non-negative. Then half the least significant bit value is added to the input to account for Perl's lack of rounding capabilities. The the scripts compare the input to the decimal value of each bit. If the input is larger than the value of the bit being compared then that bit is set and the value subtracted from the input. Otherwise the bit is not set and the next bit's value is compared. This continues until the final bit's value is checked. Then if the sign is negative the appropriate actions are taken depending upon representation.



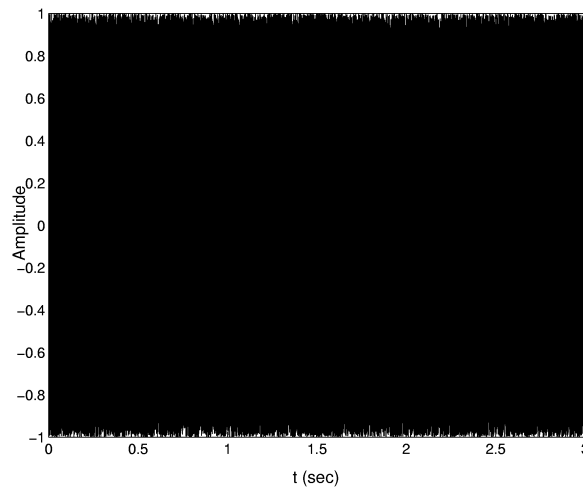


Figure 3.6: The random input set.

### 3.5.2 Verification Steps

Verifying the correctness of the output of the MACs with the above input necessitated the creation of golden models that emulate the hardware design in software. For the floating point representations one golden model that could be configured to cover the range of representations was written in LabVIEW. This virtual instrument (VI) read in the same input files that the hardware read and wrote the output to file formatted for the power estimation tool, MachPA. The file had the same input, any necessary control signals, and the output in it. When MachPA is run, the input is read in along with the control and the calculated output is compared to the output in the file. Any discrepancies are flagged and reported while the tool is running. If there are no flags then hardware output exactly matches the golden model output. For the linear and logarithmic representations a Perl script was written to act as the golden models. This script only wrote out the output so some other Perl scripts were written to build the MachPA input out of the golden model input and output files.

Further verification could be made with the discrete hardware simulation output. The golden models were written to use the same input files as the testbenches

and the testbenches wrote the output to a simple file. For the floating point verification the output had to be extracted from the MachPA input, which was done with a Perl script. Then the golden model output could be compared to the testbench output either visually or with the unix program diff. This verification flow can be seen in Figure 3.7.

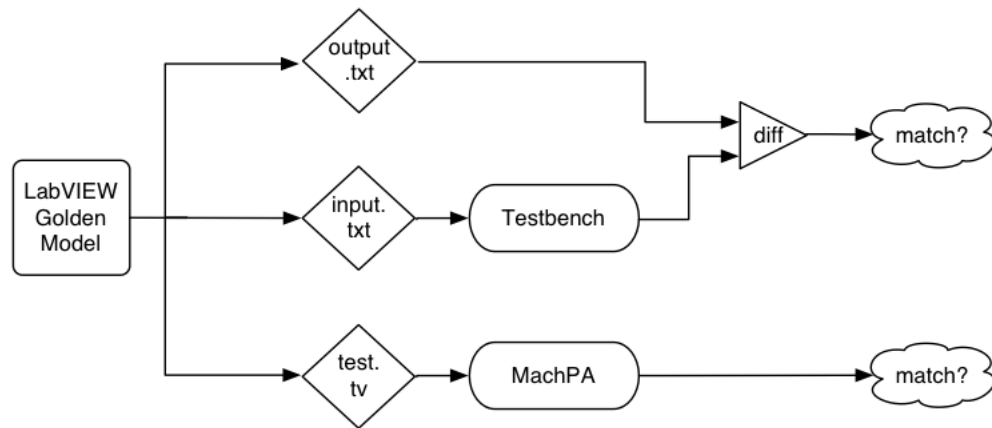


Figure 3.7: The verification flow for ASIC testing.

As one further verification the original working representation, the 1-5-5, was tested on hardware. The FPX, or field-programable port extender, is a field programmable gate array (FPGA) on a bridge between a network line card and the switch fabric of the Washington University Gigabit Switch (WUGS) [10]. Generally this platform is used for active networking applications such as routing Internet packets, compressing, encrypting, and/or buffering data, and providing different priority queues for different types of data. The 1-5-5 MAC was instantiated in an FIR controller and the hardware bit-file was loaded onto the FPGA. A sample of music was encoded into asynchronous transfer mode (ATM) cells which were streamed over the network to the filter and, after processing, the output was streamed back and recorded to a file. This file was converted to a standard wav format and played to verify the

pass-band of the 4-8 kHz filter. Also this output was compared as a waveform to the data produced by the golden model. That comparison can be seen in Figure 3.8. The apparent misalignment was shown to be a feature of the graphics capabilities of the display.

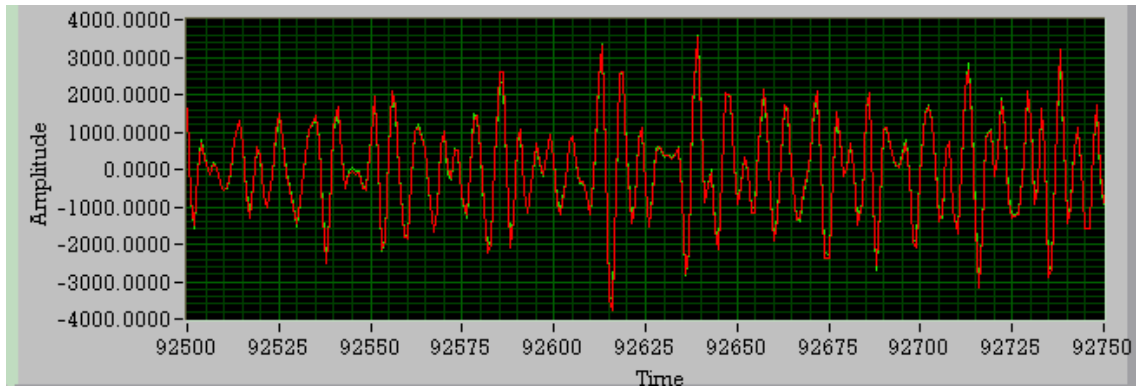


Figure 3.8: Actual 1-5-5 hardware output overlaid with golden model output for a music sample.

# Chapter 4

## Power Measurements

Two mechanisms for power estimation were used to compare the different MACs designed for different numerical representations. For each of these tests the two different input sets were used and two different complementary metal oxide semiconductor (CMOS) technologies were targeted.

### 4.1 Technologies

The MACs were targeted to both an application specific integrated circuit (ASIC) and a field programmable gate array (FPGA). The ASIC design path made use of the AMI  $.5\mu$  standard cell library [18]. This library contains building blocks that can be connected to produce the same logic as the design without the need to configure things at the transistor level. The library includes blocks such as basic AND, OR, NAND, NOR gates as well as more complicated structures such as adders and counters.

Targeting the FPGA also included a library of sorts. The synthesis tool was told what brand and model of FPGA to target. Then it could use the proper control logic blocks (CLBs) and lookup tables that the chip provides. An FPGA is made up of

many blocks like the one shown in Figure 4.1 [9]. This block contains 3 lookup tables,

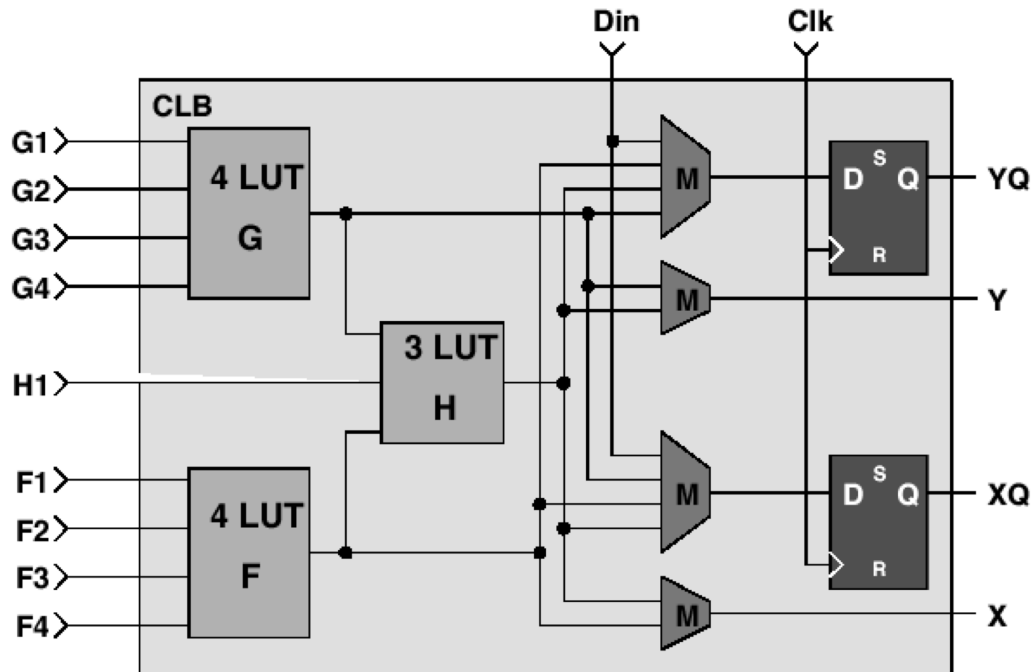


Figure 4.1: One unit of an FPGA.

two latches, and the necessary interconnects and multiplexors to connect them. By using programmable lookup tables and control blocks, FPGAs have the ability to be reconfigured. This allows for hardware tests to be performed much more conveniently than if a design had to be sent to a fabrication plant and a chip finally returned.

## 4.2 Signal Transition Counts

Basic physics tells that  $P = \frac{1}{2}fCV^2$  with  $P$  = power,  $f$  = frequency,  $C$  = capacitive load, and  $V$  = voltage. With modern digital design, frequency and voltage are primarily decided by the choice of the technology being used and the target of

the application. These decisions are choices like choosing between an FPGA and an ASIC or a  $.5\mu\text{m}$  process and a 90 nm process. The capacitive load of a single node or wire is also determined by this decision, but the overall capacitance in the circuit is determined also by the number of bits that are charged and discharged in a given amount of time.

This line of thinking says that power consumption of different circuit designs should change linearly with signal transition counts if the technology is held constant and the length of the measurement is held constant. For this reason both input sets contain the same number of values. Generally, signal transition counts are a measure of energy, but by keeping the tests a constant length power changes linearly with energy. These signal transition count tests were run with Mentor Graphics ModelSim on post synthesis VHDL. The synthesizing tool takes information about the intended target and returns a gate level VHDL file. (i.e., for an FPGA target the synthesizer would determine which lookup tables to use and what to program them with; for a standard cell ASIC the synthesizer would connect the necessary gates.) For the FPGA path, Synplicity was used and was set for a Virtex 2000E FPGA. In the case of the ASIC path, Leonardo Spectrum was used and a standard cell library for the AMI  $.5\mu$  process was targeted [18].

ModelSim is a discrete event simulator that can count the transitions on any or all signals within a design, and depending on the organization of the design it is trivial to find the distribution of power consumption based on components. This allowed for removing any transitions counted in the testbench from the total count for the circuit. The testbench was identical for each representation except for differing bit widths of signals. The testbench passed representative input values to the MAC but did not strictly emulate an FIR controller (i.e., approximately 5% of the multiply-accumulate operations were represented in the input set). This includes controlling the pairing of

coefficients and input values, controlling the accumulation and output registers, and the clock and reset signals. Each representation was run with each input set, targeted to both of the technologies. The output file with the transitions was the processed with a perl script to extract the total transition count for each test. This would take approximately 20 minutes to run the three seconds worth of data. Note that a full FIR controller that includes the entire set of multiply-accumulate operations was developed for 3 of the numerical representations. The power consumption of those systems is consistent with what follows [7].

### **4.3 MachPA Power Simulation**

To achieve more accurate estimates of power a spice-like simulator, MachPA, was used. MachPA is a continuous circuit simulator that numerically solves the differential equations that describe the circuit. To run MachPA on a design the synthesized VHDL targeted for an ASIC was placed and routed in IC Station (the 1-5-5 layout is shown in Figure 4.2), and the layout extracted. Then the golden model is used to create the appropriate output for the given input because MachPA checks its calculations against the desired output to check for correctness. Running MachPA on a complete input set can take more than 8 hours, not considering the place and route and golden model construction time.

### **4.4 Power vs. Signal Transition Counts**

Due to the large difference in processing time it is desirable to be able to focus on the signal transition counts. By running the same representations through both the discrete simulation and the continuous simulation it is possible to both verify the

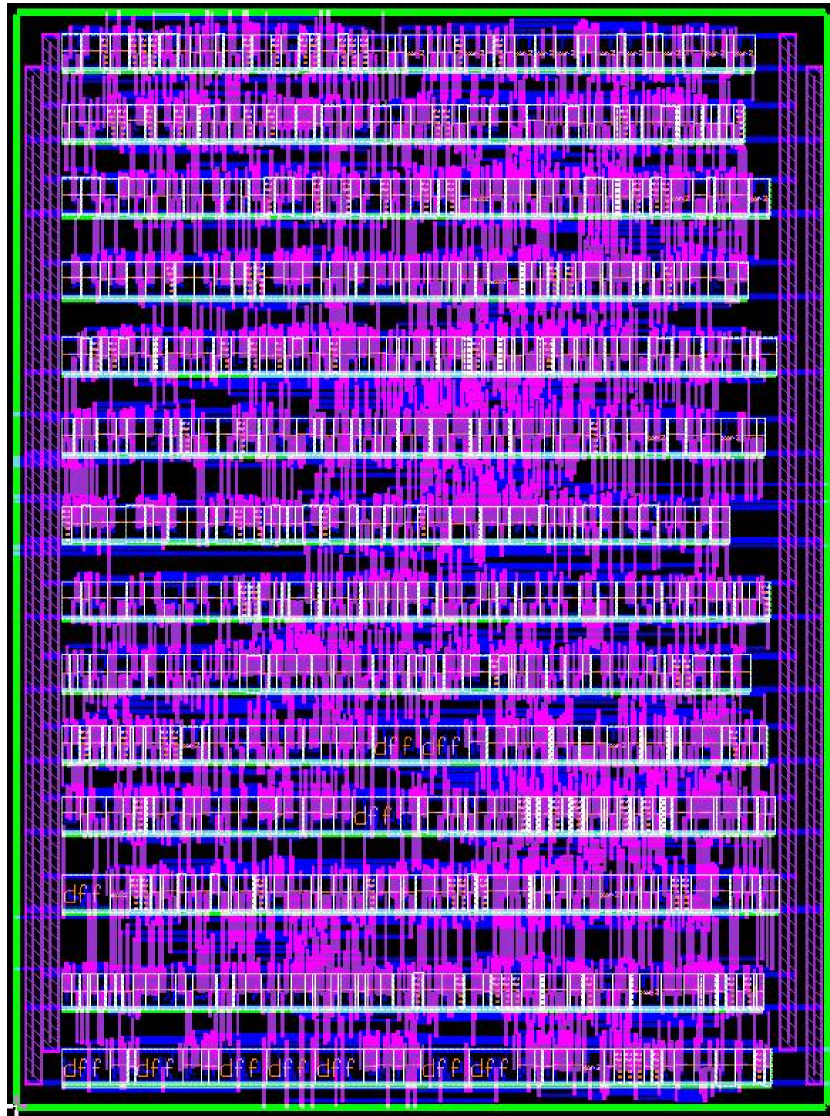


Figure 4.2: 1-5-5 IC station layout.

expected linear relationship between signal transition counts and power consumption and calibrate the conversion between the two. Figure 4.3 shows that this linear relationship is quite evident. The data points are all from the ASIC targeted VHDL and include 5 runs with the random input set and 5 with the speech input set. The representations from the random set are the logarithmic, the 1-4-4, the 1-4-5, the 1-5-4, and the 1-5-5; and from the speech set include the logarithmic, the 1-4-4, the



1-5-4, the 1-5-5, and the linear. The relationship  $y = .0649x + 1.7571$  represents

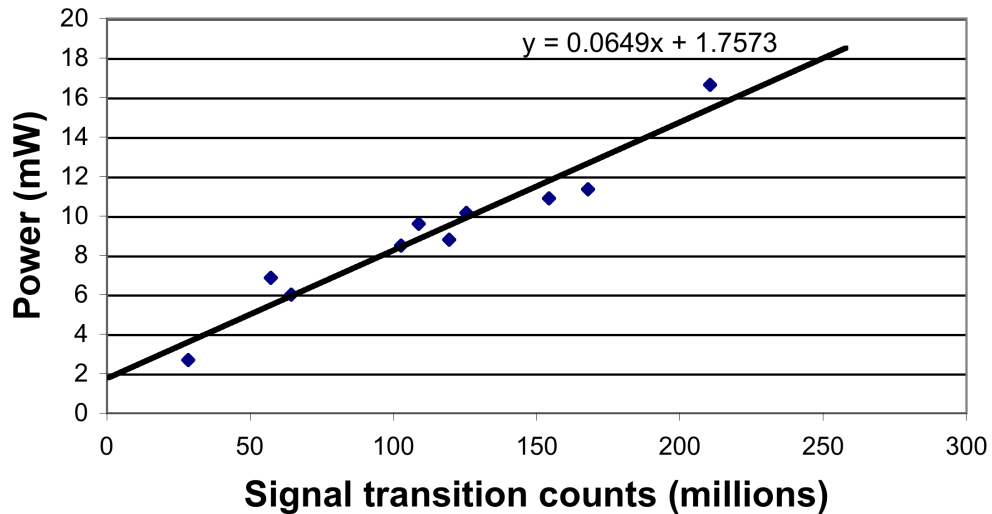


Figure 4.3: The relationship between power and signal transition count.

the least mean squared error, straight line approximation of the set of points. It takes as input the number of signal transitions divided by one million and returns the estimated power in mW. The y-intercept being at 1.7571 as opposed to zero is a result of the circuits having some signals, such as the clock, that will transition regardless of any data being present. Also the continuous circuit simulator takes into account any leakage currents present due to the feature size of the targeted technology. The relationship shown here will be used to infer power consumption from signal transition counts across all of the numerical representations. As an example of this the 1-6-5 MAC had 154,554,286 transitions with the random input set.

$$\text{power} = .0649 \cdot \frac{154,554,286}{1,000,000} + 1.7571 = 11.8mW$$

## 4.5 Power Relationship Across Representations

Below are signal transition counts and estimated power for the various combinations of representation, technology, and input set. These numbers only reflect the power consumption of the various MACs. They do not report the power consumption of a complete FIR filter as we are not modeling the controller for the MAC.

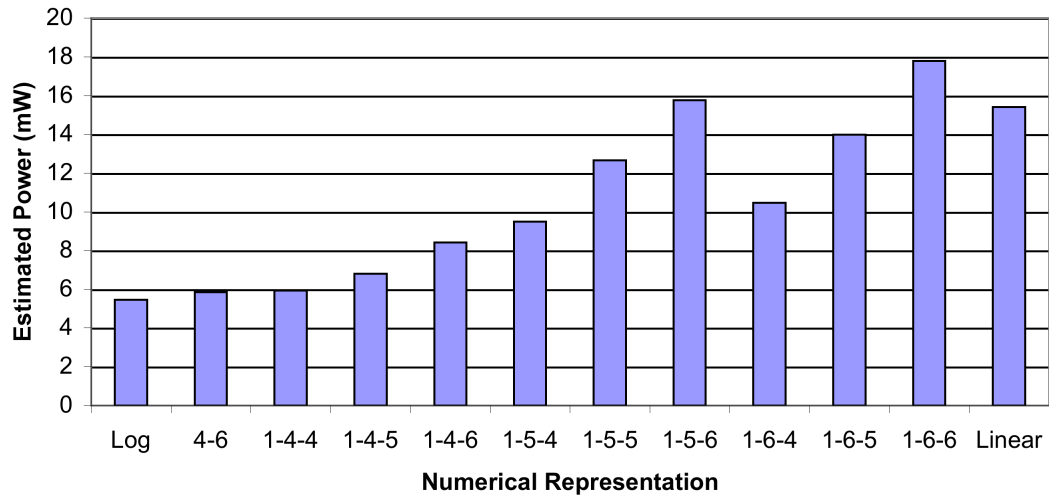


Figure 4.4: Estimated power for ASIC target with speech input for all representations.

For the ASIC target, Figures 4.4 and 4.5 show the power consumption of the various representations for the speech input set and the random input set respectively. The values shown were calculated with the linear line of best fit from Figure 4.3. The same general trends are seen in both figures. The logarithmic representation consumes the least power of any representation in both cases and the power consumption generally rises as the number of bits rise. Within the floating point representations each group of equal number of exponent bits, lengthening mantissa increases the power consumption. However, there isn't a clear trend when jumping from a mantissa of 6 bits down to a mantissa of 4 bits with a longer exponent. Table 4.1 shows the power

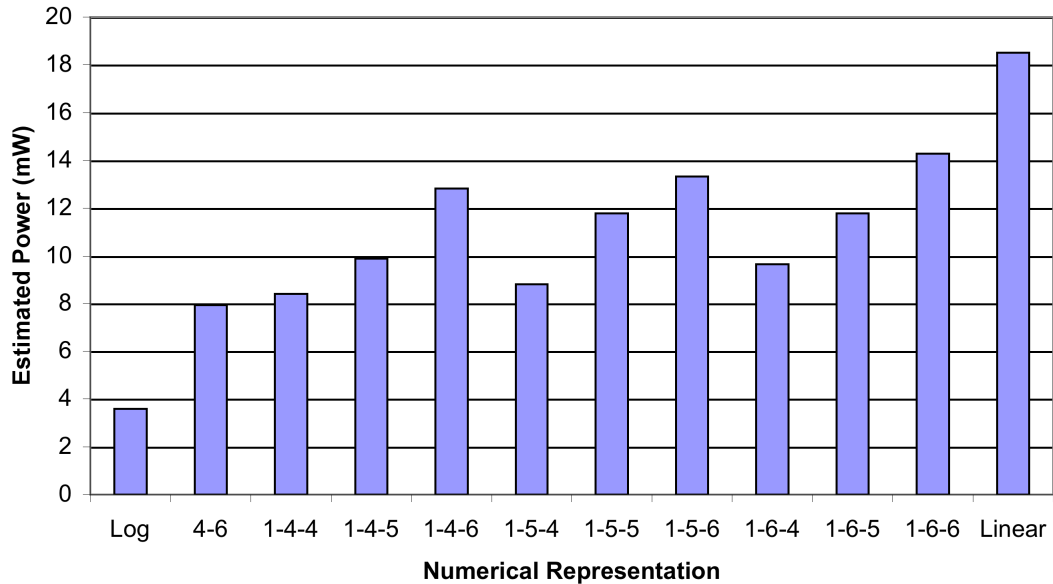


Figure 4.5: Estimated power for ASIC target with random input for all representations.

consumption of the two most power efficient representations, the revised 4-6 floating point and the logarithmic, as a percentage of the power the linear representation uses.

Table 4.1: Power consumption with respect to the linear representation.

	4-6	Logarithmic
Random	43%	19%
Speech	38%	36%

Between the sets of inputs there also isn't a clear relationship as in some cases the speech input set uses more power, and in other cases the random input set uses more power. In the case of the logarithmic representation there appears to be a much greater power savings with the random input set. This is due to the conditions in which the look up table is accessed. Speech signals are highly correlated, which makes skipping the look up table less likely, whereas the random input set would be more

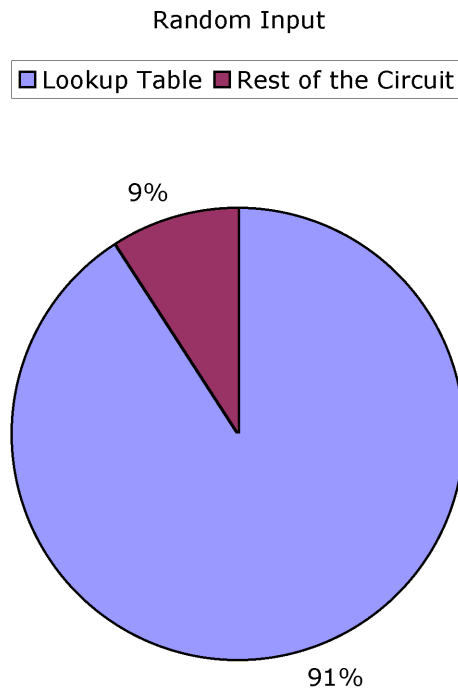


Figure 4.6: Breakdown of power consumption within logarithmic MAC with random input.

likely to have large jumps in signal level and therefore be able to bypass the look up table more often. This could have a large effect as the look up table and its control accounts for an average of  $\approx 94\%$  of the signal transition counts in the logarithmic MACs. Figures 4.6 and 4.7 show the breakdown of lookup table usage for the two different input sets.

Figures 4.8 and 4.9 show the number of signal transitions from the discrete simulations of the MAC targeted to the FPGA for the various representations. Again there is the clear upward trend with the number of mantissa bits within groups with the same number of exponent bits for the floating point representations. Also, again,

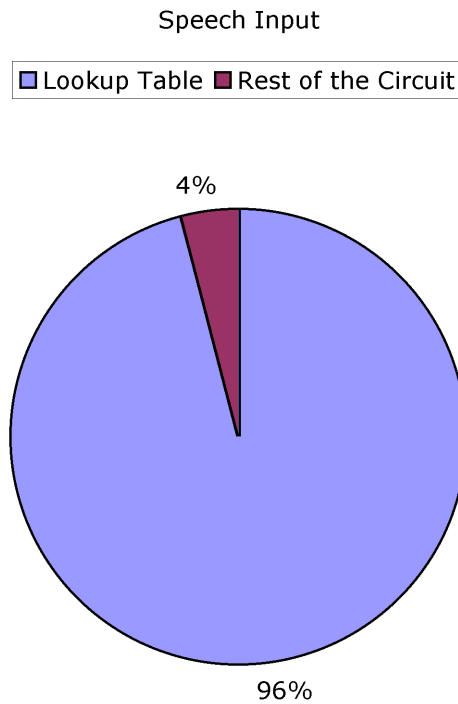


Figure 4.7: Breakdown of power consumption within logarithmic MAC with speech input.

the logarithmic representation is by far the clear choice based on power consumption alone. However, there are some relationships that don't agree with the results from the ASIC simulations. For example, with the random input, the floating point representations with 4 bits of exponent have more signal transitions than the representations with 5 and 6 bits of exponent. Since the results of the speech tests agree with what we expect this does not appear to be a design issue, however, to use these results conclusively an explanation is needed. This anomaly leads us to have higher confidence in the conclusions from the ASIC design.

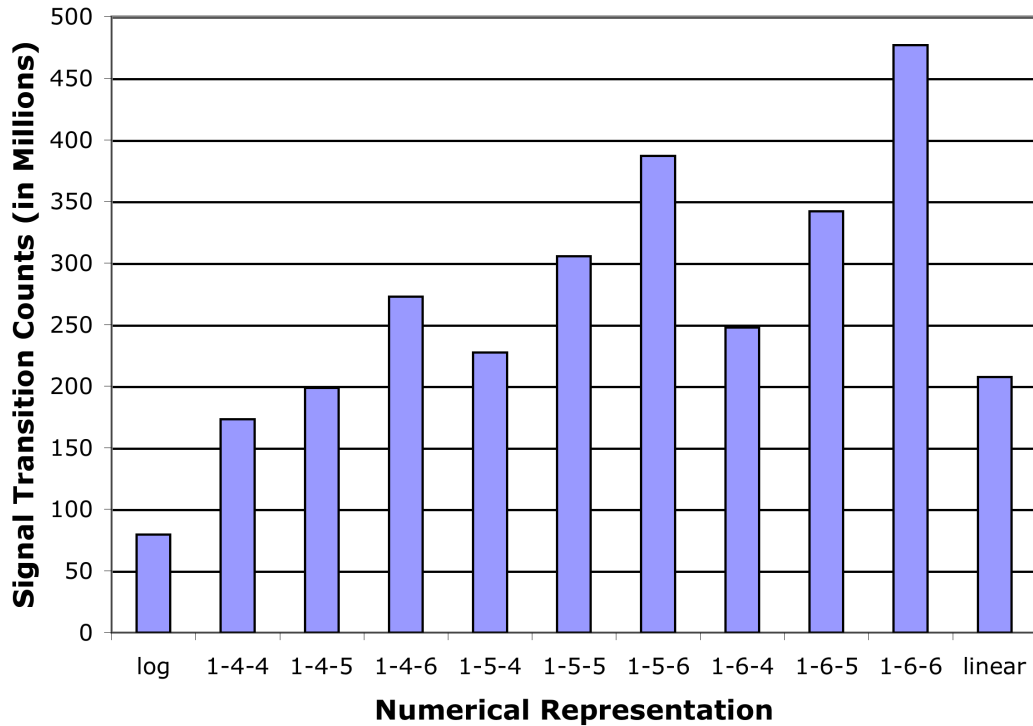


Figure 4.8: Signal transition counts for FPGA target with speech input.

## 4.6 Power Consumption for Entire Aid

For both the uni-rate and multi-rate hearing aid designs we let  $C_{total} = C_6 + C_5 + C_4 + C_3 + C_2 + C_1$  where  $C_x =$  power consumed in channel  $x$ . Also,  $C_{FIR}$  = power consumed by one 21 tap FIR filter and  $C_{LA}$  = power consumed by a hypothetical linear amplifier. For the uni-rate design:

$$C_6 = 2C_{FIR} + C_{LA}$$

$$C_5 = 4C_{FIR} + C_{LA}$$

$$C_4 = 8C_{FIR} + C_{LA}$$

$$C_3 = 16C_{FIR} + C_{LA}$$

$$C_2 = 32C_{FIR} + C_{LA}$$

$$C_1 = 64C_{FIR} + C_{LA}$$

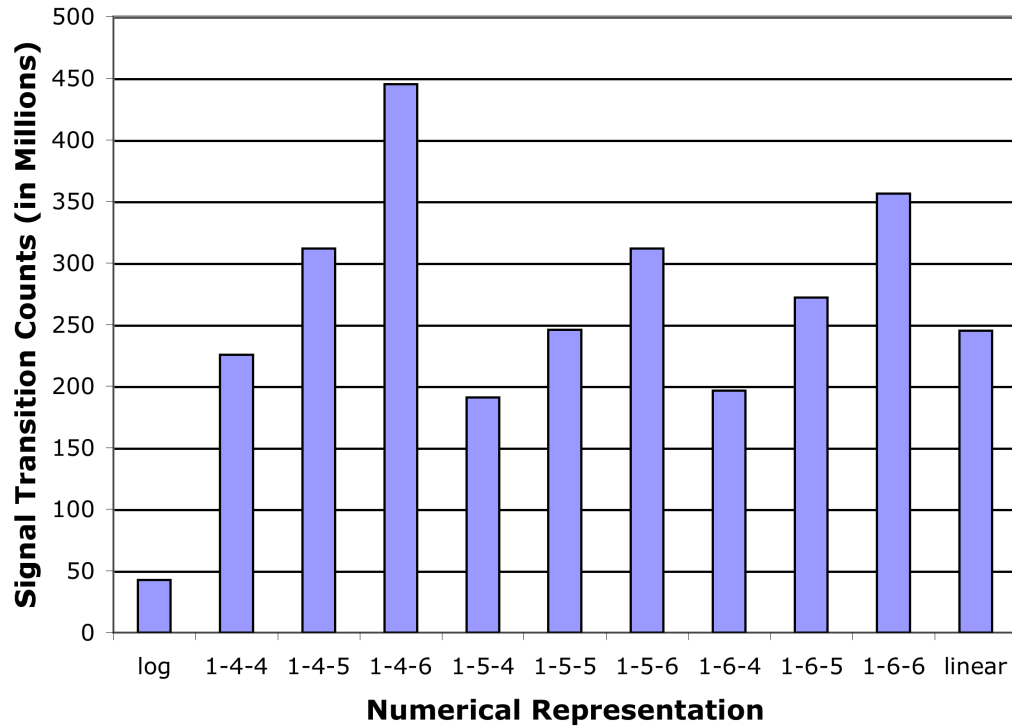


Figure 4.9: Signal transition counts for FPGA target with random input.

The number of taps doubles for each lower channel to maintain the filter's magnitude response, thus the increasing multiplier on  $C_{FIR}$ . From this,  $C_{total} = 126C_{FIR} + 6C_{LA}$  and  $C_{LA} \approx \frac{1}{21}C_{FIR}$  since a linear amplifier is just a multiply and there are 21 multiply-accumulates in one FIR. So for the uni-rate design

$$C_{total} = 126.2857C_{FIR} \quad (4.1)$$

For the multi-rate design:

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

Channels 6 and 1 only have 3 FIR filters associated with them due to the configuration of the low-pass filters. The low-pass filter at the input is associated with the channel above it since it comes before the down-sampler and therefore must run at the rate of the channel above it. The low-pass filter near the output is after the up-sampler, but every other input will be zero from the up-sampling, so it essentially runs at the rate of the channel below it. For this design  $C_{total} = 6.84375C_{FIR} + 1.96875C_{LA}$ . Again  $C_{LA} \approx \frac{1}{21}C_{FIR}$  so:

$$C_{total} = 6.9375C_{FIR} \quad (4.2)$$

From this, the multi-rate design uses only 5% of the power needed by the uni-rate design.

Table 4.2 shows the total power needed for a full hearing aid, using the equations 4.1 and 4.2 and based on the power consumption of the speech input, for the linear, logarithmic, and the 4-6 revised floating point representations. Using a logarithmic, multi-rate design uses only 2% of the power needed for a linear, uni-rate design.

While contemplating these conclusions it is important to realize that these comparisons are in terms of power only. We are not investigating the area necessary



Table 4.2: Power consumption of a full hearing aid.

	Linear	4-6	Logarithmic
Uni-rate	1947 mW	739 mW	691 mW
Multi-rate	107 mW	41 mW	38 mW

for the storage associated with the multi-rate design and its buffers. The area required is potentially significant and may rule out the multi-rate design with current technology and feature size.

## 4.7 Conclusions

From these results we can conclude a number of things on different levels. At the lowest level, it is clear that the logarithmic representation consumes the least power of the tested representations while having the audio characteristics that are necessary for a successful hearing aid. The revised 4-6 floating point representation approaches the savings of the logarithmic with speech input, however, the characteristics of the representation aren't quite as good as those of the logarithmic. While it may be possible to save more power with a floating point representation with 3 bits of exponent or mantissa the SQNR and dynamic range would be far too low. If a higher SQNR or a larger dynamic range than the logarithmic representation can provide is needed for an application the floating point representations may be able to meet the specifications while still using less power than the linear representation. We can also conclude that the multi-rate design can lower power consumption dramatically.

At a higher level, this data suggests that if power consumption is a concern in a digital signal processing application, audio or otherwise, using a customized numerical representation can provide significant benefits.

## Chapter 5

# Summary and Conclusions

This thesis has examined the power consumption ramifications of using different numerical representations for audio signals in digital hearing aids. The hearing aid design used for this examination is a multi-rate design which means that the six channels run at different sample rates (i.e. the highest frequency channel processes 32 kilosamples per second; the channel below that processes 16 kilosamples per second and so on). Within this design are 22 finite impulse response filters that consume the majority of the power used by this design. This thesis focuses on the multiply-accumulate function that is the bulk of the processing in the filters. The different numerical representations that were tested consist of the standard 16-bit two's complement linear representation, a 9-bit sign-magnitude logarithmic representation with a base of .941, 9 floating point representations with 1 sign bit and four to six bits of both exponent and mantissa, and a tenth floating point representation with a 4-bit two's complement exponent and a 6-bit two's complement mantissa. As further investigation two input sets were used in the testing of each multiply-accumulate unit. A three second speech clip from the Speech in Noise audiological test for human speech understandability was sampled at 32 kilosamples per second and a random number

generator was used to create a random input file with the same number of samples as the speech input.

A selection of these multiply-accumulate units were synthesized, placed, routed, and the layout was extracted to allow for a continuous circuit simulation of power consumption. This data was compared to signal transition counts from a discrete event simulator run on the post-synthesis design. A linear relationship was found that was used to calibrate the signal transition counts from the other combinations of numerical representation and input set. From these tests some conclusions can be drawn with regards to hearing aid design. The logarithmic representation is the clear winner in terms of audio characteristics and power consumption with both input sets. With the speech input the logarithmic representation needs only 36% of the power that the linear representation needs to process the same data. Also, by incorporating the multi-rate design the components need to process much less data which would use only 2% of the power needed for a uni-rate design. In broader terms of signal processing, or even digital circuit design, using a custom numerical representation can be very beneficial if power consumption is a constraint in the design. As the savings can be quite input dependent, for other types of signal processing the ideal representation will be highly application specific.

## 5.1 Future Work

There are a couple of directions that could take this work forward and expand upon it. The simplest include testing other clean speech samples and speech samples in noise to determine how input dependent the power savings are. More in-depth work would involve:

**Non-Linear Amplifier** If non-linear amplifiers were designed and tested for some subset of these representations then the full 6 channels could be put together into one design with some modifications for the up and down sampling filters. This would allow for further verification and calibration of the transition count to power relationship. Also the different needs of the NLA might favor a different representation which would necessitate determining which component has more of an impact on the total power consumption. This work is in progress for the linear, the logarithmic, and the 1-4-5 floating point representations[7].

**A/D and D/A** Before a hearing aid could be built to take full advantage of these results, A/D and D/A converters will be necessary. Building on the work discussed in the introduction [23], delta-sigma converters with the necessary SQNR and dynamic range could be designed.

**Hardware Testing** The most accurate determination of power requirements is through measurements made on a deployed system. One way of running on hardware would start with the existing 1-5-5 full FIR filter that has been deployed on the FPX platform and a selection of the other representations could be fully synthesized, placed, and routed and loaded onto the FPGA for physical power measurements. This would require that a full FIR controller be written for the linear and logarithmic MACs that is compatible with the current signed floating point controller. More ideally the current FIR controllers that fit in with the full channel designs would be placed-and-routed for an FPGA and the whole channel could be run on hardware. This would allow for actual measurements of current which can be easily translated into power ( $P = IV$ ). Finally, of course the desire would be to get the whole aid running on an FPGA from A/D to D/A in representative representations (i.e. linear,

logarithmic, and a few floating point representations that fit the audio characteristics well).

## References

- [1] Bilger RC, Nuetzel JM, Rabinowitz WM, Rzeckowski C. (1984). Standardization of a test of speech perception in noise. *J Speech Hear Res* 27:32-48
- [2] C. Baugh and B. Wooley. *A twos complement parallel array multiplication algorithm*. *IEEE Transactions on Computers*, 22(12):1045-1047, 1973.
- [3] CD spec. [www.disctrronics.co.uk/technology/cdaudio/cdaud\\_intro.htm](http://www.disctrronics.co.uk/technology/cdaudio/cdaud_intro.htm). Disctrronics Manufacturing (UK) Ltd.
- [4] George Engel. *Design and Analysis of Logarithmic Digital Quantizers with Applications to Low-Power Data Interfaces for Speech Processing* Doctoral Dissertation, Dept. of Electrical Engineering, Washington University. 1990.
- [5] "How Hearing Works". How Stuff Works (<http://www.howstuffworks.com/>), by Tom Harris. HowStuffWorks, Inc. 2003.
- [6] J. Dido, N. Geraudie, L. Loiseau, O. Payeur, Y. Savaria, and D. Poirier. *A Flexible Floating-Point Format for Optimizing Data-Paths and Operators in FPGA Based DSPs*. *FPGA '02*. February 2002.
- [7] Jing Lu. *Power consumption of digital hearing aid computations using customized numerical representations*. Master's Thesis, Dept. of Computer Science and Engineering, Washington University. May 2003.
- [8] Jing Lu, Eric Hemmeter, Roger Chamberlain, and Julius Goldstein. *Power Consumption Benefits of Custom Numerical Representations For Digital Hearing Aids*. Submitted to the Application Specific Annual Project. March, 2003.
- [9] John Lockwood. CS 535 Lecture Notes [http://www.arl.wustl.edu/~lockwood/class/cs535/lecture/cs535\\_lecture1.pdf](http://www.arl.wustl.edu/~lockwood/class/cs535/lecture/cs535_lecture1.pdf). Washington University, 2001.
- [10] John Lockwood , Naji Naufel, Jon Turner, and David Taylor. *Reprogrammable Network Packet Processing on the Field Programmable Port Extender (FPX)*. *ACM International Symposium on Field Programmable Gate Arrays (FPGA 2001)*, pages 87-93, Monterey, CA, February 2001.

- [11] J. L. Goldstein. *Modeling rapid compression on the basilar membrane as multiple-bandpass nonlinearity filtering*. Hear. Res., 49:39-60, 1990.
- [12] 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., August 2000.
- [13] J. L. Goldstein, M. Oz, P. Gilchrist, M. Valente. *Alternative Compressive Hearing Aid Algorithms Derived from Loudness Psychophysics and Cochlear Models*. in Proc. of Int'l Hearing Aid Research Conf., August 2002.
- [14] J. L. Goldstein, *Hearing aids based on models of cochlear compression using adaptive compression thresholds*, U.S. Patent Appl. #20020057808, May 16, 2002.
- [15] J. L. Goldstein. PI, *Hearing aids based on models of cochlear compression*, NIH SBIR grant DC04028 Phases I and II. awarded to BECS Technology, Inc., January 1999 - July 2003.
- [16] Kalikow DN, Stevens KN, Elliott LL. (1977). Development of a test of speech intelligibility in noise using sentence materials with controlled word predictability. J Acoust Soc Am 61:1337-1351
- [17] Li Shang, Alireza Kaviani, Kusuma Balthala. *Dynamic Power Consumption in Virtex<sup>TM</sup>-II FPGA Family*. FPGA '02. February 2002.
- [18] Mentor Graphics Higher Education Program: ASIC Design Kit. <http://www.mentor.com/partners/hep/AsicDesignKit/ASICindex.html>. 1997-2001.
- [19] Pavle Belanovic and Miriam Leeser. *A Library of Parameterized Hardware Modules for Floating-Point Arithmetic and Their Use*. in the Proceedings of the High Performance Embedded Computing Workshop. 2002.
- [20] Roger Chamberlain, Eric Hemmeter, Robert Morley, Jason White. *Modeling the Power Consumption of Audio Signal Processing Computations Using Customized Numerical Representations*. in the Proceedings of the 36th Annual Simulation Symposium. March 2003.
- [21] Roger Chamberlain, Yen Hsiang Chew, Varuna DeAlwis, Eric Hemmeter, John Lockwood, Robert Morley, Ed Richter, Jason White, and Huakai Zhang. *Novel Numerical Representations for Low-Power Audio Signal Processing*. Presented at the International Hearing Aid Conference. August, 2002.
- [22] Roger Chamberlain, Yen Hsiang Chew, Varuna DeAlwis, Eric Hemmeter, John Lockwood, Robert Morley, Ed Richter, Jason White, and Huakai Zhang. *Power Consumption of Customized Numerical Representations for Audio Signal Processing*. Presented at the 6th High Performance Embedded Computing Workshop. September, 2002.

- [23] S. D. Kwa. *Hybrid Sign/Logarithm Delta-Sigma Converter*. Doctoral Dissertation, Dept. of Electrical Engineering, Washington University. 1996.
- [24] S. D. Soli. *Hearing Aids: Today and Tomorrow*. Acoust. Soc. AM., 1994
- [25] S. D. Soli. *Offline simulations of logarithmic FIR filters: Measurements of processing distortion and perceptual tests of its audibility*. Bioscience Laboratory, 3M Center. October, 1988.
- [26] Thomas Sullivan. *Estimating the Power Consumption of Custom CMOS Digital Signal Processing Integrated Circuits for Uniform and Logarithmic Number Systems*. Doctoral Dissertation, Dept. of Electrical Engineering, Washington University. April, 1993.



# Vita

Eric E. Hemmeter

- Date of Birth** November 11, 1978
- Place of Birth** Red Bank, New Jersey
- Degrees** B.S. Cum Laude, Physics, Bates College, June 2001  
M.S. Computer Engineering, Washington University, St. Louis, May 2003
- Publications** Roger Chamberlain, Eric Hemmeter, Robert Morley, Jason White, *Modeling the Power Consumption of Audio Signal Processing Computations Using Customized Numerical Representations*, in the Proceedings of the 36th Annual Simulation Symposium, March 2003.
- Roger Chamberlain, Yen Hsiang Chew, Varuna DeAlwis, Eric Hemmeter, John Lockwood, Robert Morley, Ed Richter, Jason White, and Huakai Zhang, *Novel Numerical Representations for Low-Power Audio Signal Processing*, presented at the International Hearing Aid Conference, August 2002.
- Roger Chamberlain, Yen Hsiang Chew, Varuna DeAlwis, Eric Hemmeter, John Lockwood, Robert Morley, Ed Richter, Jason White, and Huakai Zhang, *Power Consumption of Customized Numerical Representations for Audio Signal Processing*, presented at the 6th High Performance Embedded Computing Workshop, September 2002.

May 2003