Modeling the Power Consumption of Audio Signal Processing Computations Using Customized Numerical Representations

Roger Chamberlain, Eric Hemmeter, Robert Morley, Jason White

Roger Chamberlain, Eric Hemmeter, Robert Morley, and Jason White, "Modeling the Power Consumption of Audio Signal Processing Computations Using Customized Numerical Representations," in *Proc. of the 36th Annual Simulation Symposium*, April 2003, pp. 249-255.

School of Engineering and Applied Science
Washington University
Campus Box 1115
One Brookings Dr.
St. Louis, MO 63130-4899

Modeling the Power Consumption of Audio Signal Processing Computations Using Customized Numerical Representations

Roger Chamberlain*†, Eric Hemmeter*, Robert Morley*, and Jason White*
*School of Engineering and Applied Science, Washington University, St. Louis, Missouri
†Hearing Emulations, LLC, and BECS Technology, Inc., St. Louis, Missouri
roger@ccrc.wustl.edu, ehemmete@ccrc.wustl.edu, rem@ee.wustl.edu, jason@earth.wustl.edu

Abstract

This paper explores the impact that numerical representation has on the power consumption of audio signal processing applications. The motivation is digital hearing aids, for which minimizing the power consumption is a critical design goal. We investigate two aspects of this problem. First, we evaluate the validity of using signal transition counts to model actual power consumption within this problem domain, and second, we compare the relative power consumption of multiply-accumulate operations for several customized numerical representations.

Keywords: audio signal processing, power consumption, numerical representation

1. Introduction

One of the major technical issues facing the designers of modern, hand-held, portable, digital systems is the need to minimize the power consumption of the system to prolong battery life. Many of these systems perform signal processing functions on audio signals (e.g., hearing aids, communications systems, MP3 players, etc). As new signal processing techniques are proposed, the computational requirements invariably grow, putting additional pressure on power consumption. In this work, we are investigating the use of non-standard numerical representations for processing audio signals, showing how the power consumption can be lowered for audio signal processing while maintaining (and even improving) overall signal quality.

Standard numerical representations used for signal processing applications include fixed-point representations (typically 16 bits) and floating-point representations (either 32- or 64-bit IEEE standard). The choices of representation available to system designers are based largely upon historical convention rather than the specific needs of the application. Our purpose is to investigate the implications of deviating from these standard representations and designing a system with a numerical representation tailored to the specific application.

Our specific motivating application is the design of digital hearing aids [1,2,3]. Here, the resource limitations can be extreme, given that the entire device (including the battery) needs to fit within the ear canal. As a result, power consumption must be held to an absolute minimum, and the use of customized numerical representations can potentially help us achieve this goal.

Focusing our attention on audio signals that communicate human speech, a dynamic range of approximately 100 dB and a signal-to-quantization-noise ratio (SQNR) of approximately 30 dB have been shown to be adequate [4]. In this investigation, we compare the power consumption of a 16-bit linear representation with several different floating-point representations (4- to 6-bit exponent and 4- to 6-bit mantissa) and a 9-bit logarithmic notation. As a representative computation, we use a series of multiply-accumulate operations. Multiply-accumulate (MAC) is the most common computation in audio signal processing. For each representation, we design a hardware MAC unit in the VHDL language and perform a standard-cell synthesis, layout, and place-and-route in the AMI Semiconductor 0.5 µm VLSI integrated circuit process.

In our previous work [5,6], the VLSI circuit design was simulated using the Mentor Graphics Mach PA™ power analysis tool with both random inputs and input vectors modeling a 21-tap finite impulse response (FIR) band-pass filter. The simulation output both verifies correct operation of the circuit and provides information on power consumption. However, this tool models the system with a continuous model, effectively solving the differential equations that describe the detailed circuit operation. While this gives highly accurate power consumption results, the execution times are unrealistically long, significantly restricting the range of candidate designs that can be effectively investigated. In this paper, we compare the results of this earlier

investigation with a logic-level simulation that models the system in a discrete-event fashion. The output of this simulation is signal transition counts, which have a linear relationship with power consumption. The increased execution speed of the logic-level simulation enables us to investigate a much wider design space, and the inherent inaccuracies present are acceptable at this point in the design cycle.

The results show a significant power savings using both the floating-point representations and the logarithmic representation. This is primarily due to the ability to either eliminate (in the case of the logarithmic representation) or significantly reduce the size of the hardware multiplier required as part of the MAC unit. The agreement between the power consumption predictions of the continuous simulation model and the discrete-event model is quite high, validating the use of the discrete-event simulation model for power estimation early in the design space exploration.

This paper provides an experimental investigation into the power consumption associated with customized numerical representations for audio signal processing applications. We report on both model fidelity for a discrete-event simulation of the system and power consumption results for a range of numerical representations. Section 2 describes the numerical representations used and the computation structures that implement the signal processing functions. Section 3 contrasts the continuous simulation model with the discrete-event model, and Section 4 compares the power consumption of the various numerical representations. Section 5 provides conclusions and directions for further work.

2. Numerical Representations and Computation Structures

For standard, fixed-point, linear, numerical representations executing MAC operations, the multiplication is significantly more expensive (in terms of power and area) than the accumulation (by about an order of magnitude). This has motivated previous investigations into logarithmic representations [7], exploiting the fact that the multiplication operations can be implemented using an adder:

$$\log(a \times b) = \log(a) + \log(b).$$

This savings must be traded off against larger power consumption in the accumulation operations, which have been implemented as look-up tables in previous work.

If a logarithmic representation gives a significant power savings in the multiplication, but costs power for the accumulation, a numerical representation that is partially logarithmic and partially linear has the potential to achieve balanced power consumption across both operations. This is precisely the description of a floating-point representation, where the exponent represents the logarithmic portion and the mantissa represents the linear portion. Rather than IEEE standard floating-point representations, however, we are interested in representations designed to more closely match the requirements of the audio signal processing application.

2.1. Properties of Numerical Representations

Figure 1 plots the dynamic range available with a 16-bit linear representation, a 9-bit logarithmic representation (using base 0.941 logarithms [8]), and nine different floating-point representations (ranging from 4 to 6 bits of exponent and 4 to 6 bits of mantissa). Here, the "e-m" notation refers to "e" exponent bits and "m" mantissa bits. For example, "6-4" is a representation with 1 sign bit, 6 bits of exponent, and 4 bits of mantissa. The dynamic range can be expressed as follows:

$$\text{Dynamic Range (dB)} = 20 \cdot \log_{10}\left(\frac{V_{\max}}{V_{\min}}\right),$$

where V_{max} corresponds to the largest value that can be represented and V_{min} corresponds to the smallest non-zero value that can be represented. Notice that for the floating-point representations, the dynamic range is determined primarily by the number of bits in the exponent. This is true because we do not support unnormalized numbers near zero due to the need to keep power consumption as low as possible.

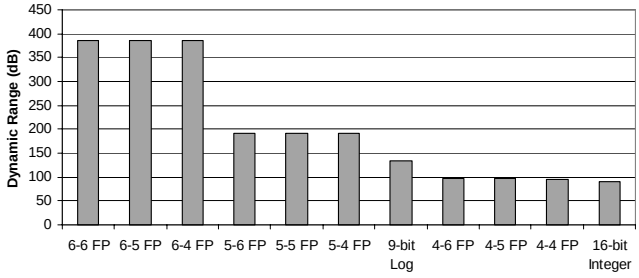

Figure 1. Dynamic range for all numeric representations.

Next we examine the SQNR associated with each of these numeric representations. SQNR is quantified as follows:

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

where V_i represents the peak value of an input sinusoid, and the quantization error is uniformly distributed between V_i and V_{i+1}, the next largest input value that can be represented in the number system. We start with the SQNR for the 16-bit integer representation, shown in Figure 2. This is followed by the 9-bit logarithmic representation in Figure 3 and three of the floating-point representations in Figure 4.

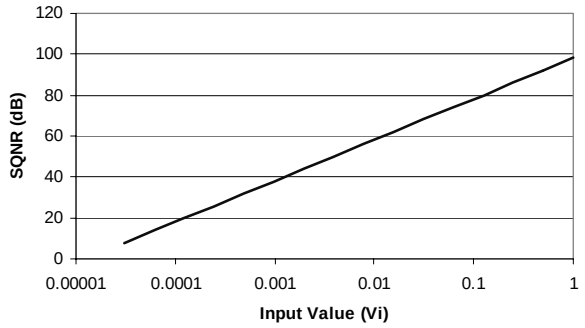

Figure 2. SQNR for 16-bit linear representation.

Figure 2 illustrates a major issue with the 16-bit linear number representation. While the signal-to-noise requirements for understanding speech are fairly constant across the available dynamic range, the SQNR of a linear representation varies significantly across the range. For low-level signals, SQNR is near 0 dB, well below the 30 dB required, and for high-level signals, SQNR is near 100 dB, well above what is needed. This points to opportunities for improved numeric properties with alternate representations.

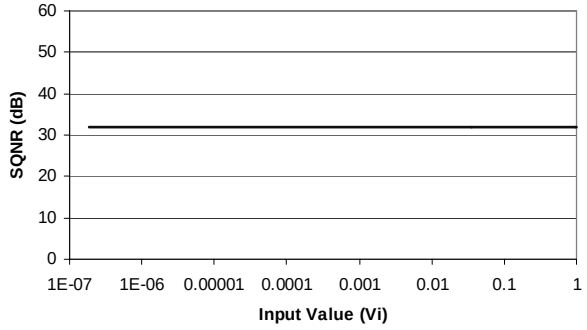

Figure 3. SQNR for 9-bit logarithmic representation.

Figure 3 illustrates one of the strong points of a logarithmic representation. Here the SQNR closely matches the requirements for understanding of human speech. It is flat at just over 30 dB over the entire dynamic range of representable values.

Figure 4 shows the SQNR for floating-point representations with 4, 5, and 6 bits in the mantissa. Although all three plots are for representations with 5 bits of exponent, the results for 4- and 6-bit exponents are the same. This is because the SQNR is not a function of the number of bits in the exponent, it is only a function of the number of bits in the mantissa. At a course level of observation, the SQNR stays relatively flat over the dynamic range of the representable values. At a finer level, however, the SQNR is seen to vary in a sawtooth fashion. This variation is due to the uneven quantization effects associated with changing one count in the mantissa versus changing one count in the exponent. The suitability of floating-point representation is still strong, however, as the SQNR can be closely matched to the needs of the application by simply choosing an appropriate number of bits in the mantissa.

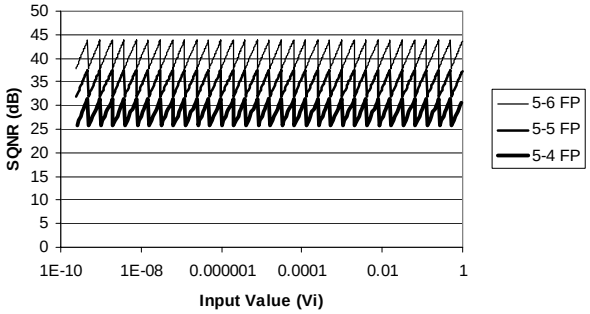

Figure 4. SQNR for three floating-point representations.

2.2. Computation Structures

The computation structure for a linear multiply-accumulate (MAC) unit is fairly traditional. Our design uses a Baugh-Wooley multiplier [9] to minimize power consumption for the multiplication operation and a standard adder leading into the accumulation register.

The MAC design for the logarithmic number representation is shown in Figure 5. In the figure, the input data is provided on the X and C inputs (typically X represents input signals and C represents filter coefficients). The multiplication operation is performed via the adder in the upper left corner of the figure. The multiplication result is compared with the current contents of the accumulator, and if the difference between the two is sufficiently large, the look-up table is bypassed and the larger of the two values is used as the new accumulation result. If the multiplication result and the current accumulator contents are sufficiently close to one another, the look-up table is accessed to determine the new accumulated value.

In the design of the look-up table, we exploit the following relationship:

$$\log(a+b) = \log(a) + \log(1+\tfrac{b}{a})$$

which, since *b/a* can be calculated via a subtraction, enables us to further limit the size of the look-up table. Essentially, the look-up table calculates the second term in the above expression.

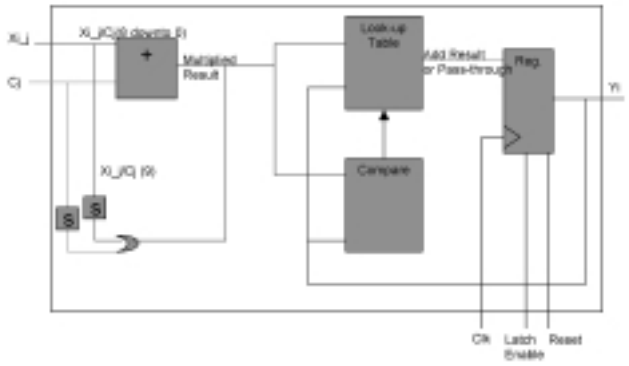

Figure 5. Functional block diagram of the 9-bit logarithmic MAC.

The structure of the floating-point MAC units is shown in Figure 6. As with the linear representations, the floating-point units are fairly traditional in design. The unit shown is for a 5-bit exponent and a 5-bit mantissa; however, the structure of the unit does not change for the other floating-point representations. The floating-point MAC units operate as follows: the data input and coefficient are separated into their three parts: sign bit, exponent bits, and mantissa bits; the mantissa bits are multiplied together using partial products, while the exponents are added; the sign bits are combined using an exclusive-or function to give the sign of the multiply result; next, the mantissa of the resulting multiply is normalized and the exponent is adjusted appropriately.

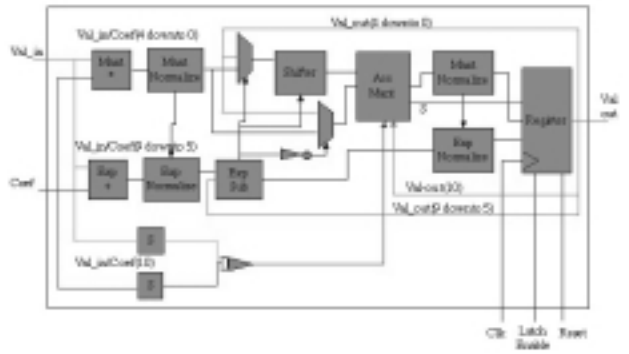

Figure 6. Functional block diagram of the 5-5 floating-point MAC.

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

In order to verify correct operation for each of the designs, a behavioral "golden model" was created using National Instruments LabVIEW™. The VHDL designs were simulated using Mentor Graphics ModelSim™, and the results compared to the behavioral model. Each MAC unit was then synthesized using Mentor Graphics Leonardo Spectrum™, targeting the ASIC Design Kit standard cell library (available through the Mentor Graphics Higher Education Program [10]) for the AMI Semiconductor 0.5 micron VLSI integrated circuit process.

3. Modeling Power Consumption

For current process technologies, the primary component of power consumption is the power required to charge and discharge the gate, drain, and interconnect capacitance associated with each signal node. For an individual signal node, the power can be expressed using the classic equation:

$$P = \tfrac{1}{2} fCV^2,$$

where f is the frequency with which the signal changes state (i.e., transitions from 0 to 1 or 1 to 0), C is the capacitive load on the signal, and V is the voltage swing of the signal transition. Expanding this equation to support multiple signals is straightforward:

$$P = \tfrac{1}{2} \sum_{i=1}^{n} C_i \alpha_i f V^2 .$$

Here, we are expressing the power consumption for n nodes, C_i is the capacitive load on node i, and $\alpha_i f$ is the signal transition frequency for node i, expressed as the product of f, the clock frequency for the circuit as a whole, and α_i, the fraction of clock cycles for which node i has a signal transition.

Knowledge of the actual values of capacitance for all n signal nodes in a circuit is not readily available during the point in the design process for which we would like to perform comparisons of the power consumption between candidate designs. As a result, it is common practice to make a few simplifying assumptions to the above equation that will enable power consumption comparisons earlier in the design cycle. Assuming the following:

1. for a gate-level representation of the circuit, the signal transitions at gate outputs are representative of the signal transitions for all signals, both internal and external to the gates, and
2. the load capacitance associated with each signal is the same,

we can simplify the above power expression to

$$P = \tfrac{1}{2} C f V^2 \frac{n}{m} \sum_{i=1}^{m} \alpha_i \ .$$

C is now the average value of load capacitance and the ratio n/m, where m is the number of gate outputs and n is the total number of signals, scales the power to account for not including the internal gate signals in the summation. The first 5 terms of the power expression $\left(\tfrac{1}{2} C f V^2 \frac{n}{m} \right)$ do not significantly change from one design candidate to the next. The upshot of this simplification is that it is now feasible to estimate the (relative) power consumption of several candidate design options by comparing just the summation expression in the above power equation. The values for m and α_i are readily available from discrete-event logic simulations of gate-level designs.

Given the gate-level designs that resulted from the synthesis described above, the following tools (all from Mentor Graphics) were used for the remainder of the design and analysis. ModelSim™ was used for discrete-event logic simulation of the gate-level designs. IC Station™ was used to perform automated place-and-route (the resulting layout for the 5-bit exponent, 5-bit mantissa floating-point MAC is shown in Figure 7). Finally, we extracted the layout in IC Station™ and ran Mach PA™ to analyze power consumption. Mach PA™ takes a test vector input file, generated by the "golden model," which contains both the inputs and expected outputs. It performs a continuous circuit simulation of the design (numerically solving the differential equations that describe the circuit operation), compares the simulated output values with the expected outputs (reporting any discrepancies, if present) and provides highly accurate current usage. With knowledge of the average current, I, power consumption is straightforward to calculate:

$$P = IV \ .$$

Upon completion of the design process, we have the ability to perform both discrete-event logic simulations of each MAC unit using the gate-level model and continuous simulations of each MAC unit using the circuit-level model. The input vectors come in two forms: a set of random inputs (uniformly distributed over the full dynamic range of the number representation) and a 21-tap finite-impulse response bandpass filter used to filter a representative speech sample (drawn from one of the sentences in the SPeech In Noise (SPIN) audiological test for human speech understandability). Each discrete-event simulation required about 10 min. to execute, and the continuous simulations each had execution times of 8 hours or more. The discrete-event simulation was executed with both input vector sets for all numerical representations, and the continuous simulation was executed for a representative subset of the input vectors and numerical representations.

Mentor Graphics IC Station Layout for the 5-5 Floating-Point Representation

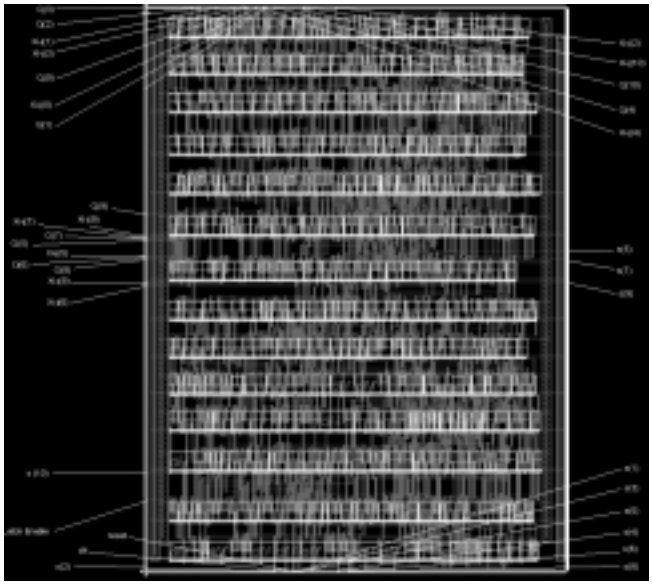

Figure 7. Standard-cell layout for floating-point MAC unit.

4. Simulation Results

The appropriateness of using the more abstract discrete-event model for evaluating power consumption is examined in Figure 8. The power consumption results from the continuous model are plotted against the signal transition counts from the discrete-event model. Note that, strictly speaking, transition counts are an energy measure rather than a power measure; however, as long as the number of clock cycles is consistent, power and energy are linearly related and either can be used. The results from 10 distinct experiments are shown, including 5 random input sets and 5 speech input sets. The numerical representations include the 9-bit logarithmic representation, the 16-bit linear representation, and several of the floating-point representations. Also shown on the plot is a least-mean-squared error linear curve fit to the data, used below to generate estimated power from transition counts.

The conclusions that can be drawn from this graph are quite clear. Although the fidelity is not perfect, signal transition counts that are measured from the discrete-event simulation model are an excellent predictor of the power consumption that is measured from the continuous simulation model. This is true independent of input data set or structural design of the MAC unit. The standard

deviation in the error between the transition counts estimating power and actual power (from the continuous simulation) is 0.9 mW. In the results that follow, we report data based on signal transition counts, the discrete-event simulation output.

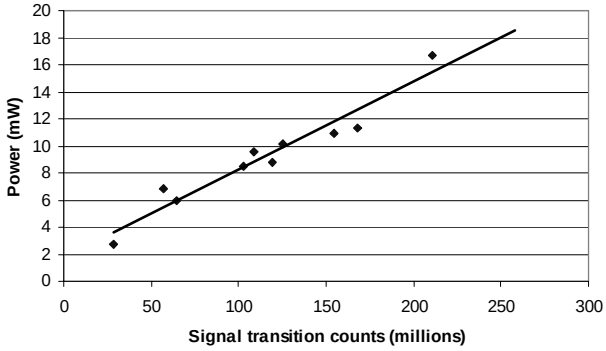

Figure 8. Continuous vs. discrete power models.

Figure 9 presents the relative power consumption (as estimated power and labeled with signal transition counts, in millions) for the eleven different numerical representations for the experiments with random input vectors. Figure 10 presents the same information for the experiments with speech input vectors. For random input vectors, the improvement in power consumption for the customized numerical representations (vs. the 16-bit integer representation) is quite dramatic, up to an order of magnitude better for the 9-bit logarithmic representation and power savings ranging from 25% to 60% for the various floating-point representations. As one would expect, the drop in power consumption is more dramatic for number representations that use fewer bits.

The power savings for speech inputs are not as dramatic, but are still significant. The 9-bit logarithmic representation again provides the greatest improvement, requiring only 27% of the power as the 16-bit linear representation. Essentially, the integer representation requires 3.7 times more power to perform the same function as the logarithmic representation. For small exponent, small mantissa floating-point representations, the power is now somewhat competitive with the logarithmic representation; however, the larger exponent, larger mantissa floating-point representations are near (or even exceed) the power consumption of the integer representation.

The results indicate that the power consumption of a MAC unit is clearly dependent upon the inputs used to exercise the system. The fact that the logarithmic representation has such a low power consumption for the random input vectors is attributable to the design optimization that enables the look-up table to be bypassed for accumulation operations that have input values sufficiently far apart in value. Since the random input vectors are uniformly distributed across the dynamic range, this bypass opportunity will be exercised much more frequently than is the case for speech signals. The improved power consumption for the linear representation when operating on speech signals can be attributed to the same fact. Since the speech signal amplitude is significantly less than the peak value of the dynamic range, the high-order bits of the integer MAC unit stay relatively stable for the speech input and are active for the random inputs.

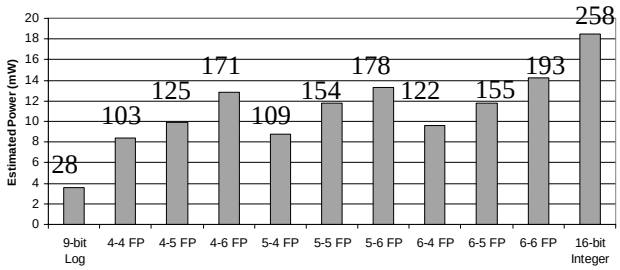

Figure 9. Relative MAC unit power consumption, random input vectors.

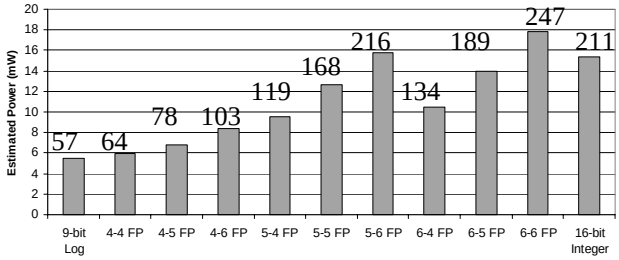

Figure 10. Relative MAC unit power consumption, speech input vectors.

Overall, the numerical properties and minimum power consumption associated with the 9-bit logarithmic representation point to it as the best option (of those considered) for our motivating application, digital hearing aids.

5. Conclusions

This paper has presented a comparison of two distinct simulation models for estimating the power consumption associated with custom numerical representations for audio signal processing applications, indicating that the less computationally intensive discrete-event simulation model is quite adequate for relative power consumption comparisons. We have also contrasted the power requirements for multiply-accumulate operations implemented using eleven different numerical representations, including a 16-bit linear representation, a 9-bit logarithmic representation, and nine floating-point

representations, ranging from 4 to 6 bits in the exponent and 4 to 6 bits in the mantissa. Overall, the logarithmic representation has the lowest power requirements for both random input vectors and speech input vectors.

Although the specific results presented here are limited to a small range of numerical representations, the general message is much broader. With the ready availability of ASIC fabrication, rigid, fixed-function computational hardware is no longer a necessity of modern digital system design, and significant power savings can result if the requirements of the application are used to specify the properties of the numerical representation. While this work was initially motivated by the needs of digital hearing aids [1,2,3], it is applicable to a wide variety of digital signal processing applications, such as cellular telephony, etc.

Our current work is expanding the investigation to include alternate designs for the floating-point representations (e.g., no longer restricting the representation to an excess notation for the exponent and signed-magnitude for the mantissa) and a wider set of computations (e.g., including the non-linear amplification functions of [3]).

Acknowledgements

This research is supported in part by NSF under grant DGE-0138624 and by NIH under grant 1R4-3DC04028-02 through BECS Technology, Inc., and Hearing Emulations, LLC. Roger Chamberlain has a financial interest in these two firms.

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

[2] H.Levitt., J.M. Pickett, and R.A. Houde, *Sensory Aids for the Hearing Impaired*, IEEE Press, NY, 1980.

[3] J.L. Goldstein, M. Valente, and R.D. Chamberlain, "Acoustic and psychoacoustic benefits of adaptive compression thresholds in hearing aid amplifiers that mimic cochlear function," *J. Acoust. Soc. Am.* **109**:2355(A), 2001.

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

[5] R. Chamberlain, Y.H. Chew, V. DeAlwis, E. Hemmeter, J. Lockwood, R. Morley, E. Richter, J. White, and H. Zhang, "Novel Numerical Representations for Low-Power Audio Signal Processing," in *Proc. of Int'l Hearing Aid Research Conf*., August 2002.

[6] R. Chamberlain, Y.H. Chew, V. DeAlwis, E. Hemmeter, J. Lockwood, R. Morley, E. Richter, J. White, and H. Zhang, "Power Consumption of Customized Numerical Representations for Audio Signal Processing," in *Proc. of High Performance Embedded Computing Workshop*, September 2002.

[7] T.J. Sullivan, "Estimating the power consumption of custom CMOS digital signal processing integrated circuits for uniform and logarithmic number systems," D.Sc. dissertation, Washington University, 1993.

[8] S.W. Kwa, "Design and analysis of a hybrid sign/logarithm delta-sigma analog-to-digital converter for audio applications," D.Sc. dissertation, Washington University, 1996.

[9] C. Baugh and B. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Transactions on Computers*, **22**(12):1045-1047, 1973.

[10] Mentor Graphics Higher Education Program:ASIC Design Kit, http://www.mentorg.com/partners/hep/AsicDesignKit/ASICindex.html, 1997-2002.