

Parameterized Module Generator for an FPGA-Based Electronic Cochlea

M.P. Leong[†], C.T. Jin^{*} and P.H.W. Leong[†]
{mpleong@cse.cuhk.edu.hk craigj@physiol.usyd.edu.au phwl@cse.cuhk.edu.hk}

[†]Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T. Hong Kong
and

^{*}Department of Electrical and Information Engineering
The University of Sydney, Australia 2006

Abstract

An FPGA-based implementation of Lyon and Mead's electronic cochlea filter and its application to a real-time cochleagram display are presented. Compared with analog VLSI implementations, an FPGA implementation offers shorter design time, improved dynamic range, higher accuracy and a simpler computer interface. The FPGA cochlea filter is generated by a tool which takes filter coefficients to compile an application-optimized design with arbitrary precision. In the process of compilation, the tool can use simulation test vectors in order to determine the appropriate scaling for each filter. The resulting model can be used as an accelerator for cochlea model research or as the front end for embedded auditory signal processing systems.

1 Introduction

It is clear that biological-based systems perform feats of signal processing that we cannot approach using even the most sophisticated computers and digital signal processing techniques. Generally, biological-based auditory systems operate with greater functionality, lower power consumption and increased robustness than their man-made electrical counterparts. This is particularly true in tasks such as speech recognition where humans are able to process signals far better than the most sophisticated computer-based systems. We can learn a lot from the elegant designs of nature.

The field of neuromorphic engineering has the long term objective of taking architectures from our understanding of biological systems to develop novel signal processing systems. This field of research, pioneered by Carver Mead [1] has concentrated on using analog VLSI to model biological systems. Research in this field has led to many biologically inspired signal processing systems which have improved performance compared to traditional systems.

The human cochlea is a transducer which converts mechanical vibrations from the middle ear into neural electrical discharges, and additionally provides spatial separation of frequency information in a manner similar to that of a spectrum analyzer [2]. It serves as the front end signal processing for all functions of the auditory nervous system such as auditory localization, pitch detection and speech recognition.

Although it is possible to simulate cochlea models in software, hardware implementations may have orders of magnitude of improvement in performance. Hardware implementations are also attractive when the target applications are on embedded devices in which power-efficiency and small-footprint are design considerations.

The electronic cochlea, first proposed by Lyon and Mead [2] is a cascade of biquadratic filter sections (as shown in Figure 1) which mimics the qualitative behavior of the human cochlea. Electronic cochleas have been successfully used in many auditory signal processing systems such as spatial localization [3], pitch detection [4], a computer peripheral [5], amplitude modulation detection [6], correlation [7] and speech recognition [8].

There have been several implementations of elec-

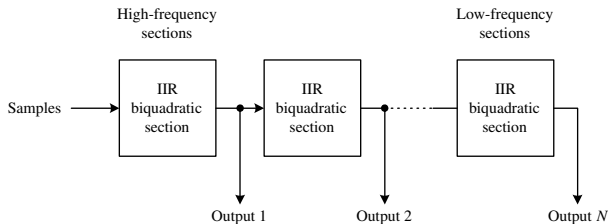


Figure 1: Cascaded IIR biquadratic section used in the Lyon and Meads cochlea model.

tronic cochleas in VLSI technology. The original implementation by Lyon and Mead was published in 1988 and used continuous time subthreshold transconductance circuits to implement a cascade of 480 stages [2, 9]. In 1992, Watts et. al. reported a 50-stage version with improved dynamic range, stability, matching and compactness [10]. A problem with analog implementations is that transistor matching issues affect the stability, accuracy and size of the filters. This issue was addressed by van Schaik et. al. in 1997 using compatible lateral bipolar transistors instead of MOSFETs in parts of the circuit [11]. Their 104-stage test chip showed greatly improved characteristics. In addition, a switched capacitor cochlea filter was proposed by Bor et. al. in 1996 [12].

There have also been several previously reported digital VLSI cochlea implementations. In 1992, Summerfield and Lyon reported an application-specific integrated circuit (ASIC) implementation which employed bit-serial second-order filters [13]. In 1997, Lim et. al. reported a VHDL-based pitch detection system which used first-order Butterworth band-pass filters for cochlea filtering [14]. Later in 1998, Brucke et. al. designed a VLSI implementation of a speech preprocessor which used gammatone filter banks to mimic the cochlea [15]. The implementation by Brucke et. al. used fixed-point arithmetic and they also explored tradeoffs between wordlength and precision.

Recently, field programmable gate array (FPGA) technology has improved in density to the point where it is possible to develop neuromorphic systems on a single FPGA. It is our thesis that many interesting neuromorphic signal processing systems can be implemented using FPGA technology, enjoying the following advantages over analog VLSI

- shorter design time
- faster fabrication time

- more robust to power supply, temperature and transistor mismatch variations
- wider dynamic range and higher signal to noise ratios
- better stability
- the chips can be reused for different application
- simpler computer interface.

In this paper, we present an FPGA implementation of an electronic cochlea which can serve as an accelerator in its own right, or as a front end preprocessing stage for embedded auditory applications. A module generator which can generate synthesizable VHDL descriptions of arbitrary wordlength fixed-point cochlea filters was developed. The module generator can also be used together with our *fp* simulation tool [16, 17] to determine the minimum and maximum ranges of all variables. This range information is then used to determine the number of fractional bits used in the variable's two's complement fraction representation. Finally, as a sample application, a real-time cochleagram display is presented.

The rest of the paper is organized as follows. In Section 2 Lyon and Mead's cochlea model is described. Section 3 describes the implementation of the filter stages using distributed arithmetic (DA). Our design methodology is presented in Section 4 followed by results in Section 5. Conclusions are drawn in Section 6.

2 Lyon and Mead's Cochlea Model

Lyon and Mead proposed the first electronic cochlea in 1988 [2, 18]. This model captured the qualitative behavior of the human cochlea using a simple cascade of second order filter stages which they implemented in analog VLSI. In this section a very superficial summary of the Lyon and Mead cochlea model is given. More detailed descriptions of the cochlea can be found in [2] and [19].

The human cochlea, or inner ear, is three dimensional fluid-dynamic system which converts mechanical vibrations from the middle ear into neural electrical discharges [2]. It is composed of the basilar membrane, inner hair cells and outer hair cells. The cochlea connects to higher levels in the auditory pathway for further processing.

The basilar membrane is a longitudinal membrane within the cochlea. The oval window is the input to the cochlea. Vibrations of the eardrum are coupled

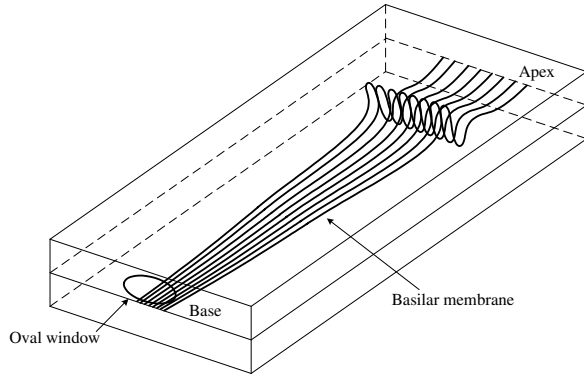


Figure 2: Illustration of a sine wave travelling through a simplified box model of an uncoiled cochlea (adapted from [2]).

via bones in the middle ear to the oval window causing a traveling wave from base to apex along the basilar membrane. The basilar membrane has a filtering action and can be thought of as a cascade of lowpass filter with exponentially decreasing cutoff frequency from base to apex.

The result of the filtering of the basilar membrane at any point along its length is a bandpass filtered version of the input signal, with center frequency decreasing along its length. Different distances along the basilar membrane are tuned to specific frequencies in a manner similar to that of a spectrum analyzer. A simplified box model showing a sinusoidal wave traveling along an uncoiled cochlea is shown in Figure 2.

Several thousand inner hair cells are distributed along the basilar membrane and convert the displacement of the basilar membrane to a neural signal. The hair cells also perform a half-wave rectifying function since only displacements in one direction will cause neurons to fire.

The outer hair cells perform automatic gain control by changing the damping of the basilar membrane. It is interesting to note that there are approximately three times more outer hair cells than inner hair cells.

In order to simulate the properties of the basilar membrane, Lyon and Mead's cochlea model used a cascade of scaled second-order low-pass filters with the transfer function

$$H(s) = \frac{1}{\tau^2 s^2 + \frac{1}{Q} \tau s + 1} \quad (1)$$

where Q represents the damping characteristic (or quality) of the filter and τ the time constant. In the cochlea filter, the τ of each filter is varied exponen-

tially along the cascade, causing filters to have exponentially decreasing cutoff frequencies. The Q of all the filters is held constant. The outputs of each filter corresponds to the displacement of different positions along the basilar membrane.

3 IIR Filters Using DA

3.1 Distributed Arithmetic

Distributed arithmetic (DA) offers an efficient method to implement a sum of products (SOP) provided that one of the variables does not change during execution. Instead of requiring a multiplier, DA utilizes a precomputed look-up table [20, 21].

Consider the SOP, S of N terms

$$S = \sum_{i=0}^{N-1} k_i x_i \quad (2)$$

where k_i is the (fixed) weighting factor and x_i is the input. For two's complement fractions, the numerical value of $x_i = \{x_{i0}x_{i1} \dots x_{i(n-1)}\}$ is

$$x_i = -x_{i0} + \sum_{b=1}^{n-1} x_{ib} \times 2^{-b}. \quad (3)$$

Substituting Equation 3 into Equation 2 yields

$$\begin{aligned} S = & - (x_{00} \times k_0 + x_{10} \times k_1 + \dots + x_{(N-1)0} \times k_{N-1}) \\ & \times 2^0 \\ & + (x_{01} \times k_0 + x_{11} \times k_1 + \dots + x_{(N-1)1} \times k_{N-1}) \\ & \times 2^{-1} \\ & + (x_{02} \times k_0 + x_{12} \times k_1 + \dots + x_{(N-1)2} \times k_{N-1}) \\ & \times 2^{-2} \\ & \vdots \\ & + (x_{0(n-1)} \times k_0 + x_{1(n-1)} \times k_1 + \dots + \\ & x_{(N-1)(n-1)} \times k_{N-1}) \times 2^{-(n-1)} \end{aligned} \quad (4)$$

The organization of the input variables are in a bit-serial, least significant bit (LSB) first format. Since $x_{ij} \in \{0, 1\}$ ($i = 0, 1, \dots, N-1$, $j = 0, 1, \dots, n-1$), each term within the brackets of Equation 4 is the sum of weighting factors k_0, k_1, \dots, k_{N-1} . On every clock cycle, one of the bracketed terms of S can thus be computed by applying x_0, x_1, \dots, x_{N-1} as the address inputs of a $2^{(N-1)}$ entry read-only memory (ROM). The contents of the ROM are precomputed from the constant k_i 's and are shown in Table 1. The output

$b_{N-1} \dots b_2 b_1 b_0$	Address	Contents
0 ... 000	0	0
0 ... 001	1	k_0
0 ... 010	2	k_1
0 ... 011	3	$k_0 + k_1$
0 ... 100	4	k_2
0 ... 101	5	$k_0 + k_2$
0 ... 110	6	$k_2 + k_1$
0 ... 111	7	$k_0 + k_1 + k_2$
\vdots	\vdots	\vdots
1 ... 111	2^{N-1}	$k_0 + k_1 + \dots + k_{N-1}$

Table 1: Contents of a DA ROM. For each address, the terms k_i for which $b_i = 1$ are summed.

of the ROM is multiplied by a power of two (a shift operation) and then accumulated. After n cycles, the accumulator contains the value of S .

3.2 Digital IIR Filters

Equation 1 can be converted from the s -domain to the z -domain via a bilinear transform. The resulting transfer function has the form

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}.$$

The corresponding time domain IIR filter can be implemented by the function

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + a_0 y(n-1) + a_1 y(n-2)$$

where $x(n-k)$ is the k 'th previous input, $y(n-k)$ is the k 'th previous output and $y(n)$ is the output. The operation is essentially the SOP of five terms, and can be directly map to a biquadratic section as shown in 3.

Figure 4 illustrates our actual implementation using distributed arithmetic (described in Section 3.1) on an Xilinx Virtex FPGA. The previous values $x(n-1)$, $x(n-2)$, and $y(n-2)$ are implemented using shift registers with the number of stages equal to the wordlength of the variables used. The shift registers are implemented by cascades of Virtex SRL16E primitives for minimum area. The DA ROM takes $x(n)$, $x(n-1)$, $x(n-2)$, $y(n-1)$ and $y(n-2)$ as inputs to generate partial sums (bracketed terms in Equation 4). As there are 5 inputs, the required number of entries in the ROM is $2^5 = 32$, leading to an efficient implementation using Xilinx ROM32X1 primitives. The scaling

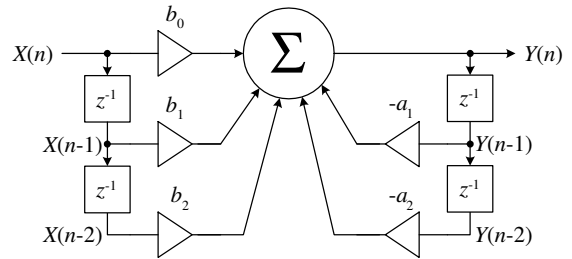


Figure 3: The architecture of an IIR biquadratic section.

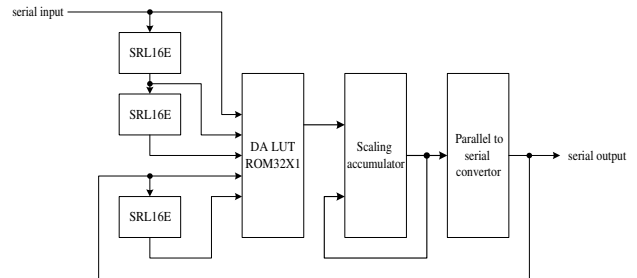


Figure 4: Implementation of an IIR biquadratic section on an Xilinx Virtex FPGA.

accumulator shifts and adds the output from the ROM (unscaled partial sum in bit-parallel organization) at every cycle to produce $y(n)$. In the last cycle of scaling and accumulation, the parallel to serial converter latches the value at the scaling accumulator. Since the scaling accumulator has a latency equal to the wordlength of the variables, the value latched by the converter is $y(n-1)$.

4 Design Methodology

Given the filter coefficients, the designer selects appropriate values of filter wordlength, and the number of bits (width) of the DA ROM's output. Note that all filter sections have the same wordlength although the allocation of integer and fractional parts used within each filter section can vary.

The cochlea filter model is written in a subset of C which supports only expressions and assignments. A compiler uses standard parsing techniques to translate expressions into directed acyclic graphs (DAG). Each node in the DAG carries out an operation on a set of operands (edges incident to the node) and produces a set of results (edges incident from the node). Each operator is mapped to a module which is a software

object, consisting of a set of parameters, a simulator and a component generator. The simulator can perform the operation at a requested precision to determine range information. It can also compare fixed-point output with a floating-point computation to derive error statistics. For this cochlea model, we defined a new operator – the IIR biquadratic section. Indeed, the sole class of operator used in this model is the IIR biquadratic section.

The coefficients for the biquadratic filters in our implementation of Lyon and Mead’s cochlea model were obtained using Malcom Slaney’s Auditory Toolbox [22]. This MATLAB toolbox has several different cochlea models, test inputs and visualization tools. The same toolbox was used to verify our designs and produce cochleagram plots.

As input, the *fp* cochlea generator takes the coefficients obtained from Auditory Toolbox, the wordlength of variables and the width of the DA ROM. Although inputs and outputs of all filter sections are of the same wordlength, their fractional wordlength can be different (two’s complement fractions are used). The dynamic ranges of inputs and outputs are determined by *fp* through simulation of a set of user supplied test vectors. The generator performs simulation using the test vectors as inputs and the range of each variable can be determined. From this information the minimum number of bits needed for the integer part of each variable is known and since the wordlength is fixed, the maximum number of bits can be assigned to the fractional part of the variable.

After deducing the best representation for each variable, the generator outputs synthesizable VHDL code that describes an implementation of the corresponding cochlea model. The fractional wordlengths of the scaling accumulator and the output variable can be different, so the operator must also include a mechanism to convert the former to the latter. Since the output of the scaling accumulator is bit-parallel while the output variable is bit-serial, the parallel to serial converter can perform format scaling by selecting the appropriate bits to serialize. The resulting VHDL description can then be used as a core in other designs.

The high level cochlea model description is approximately 60 lines of C code. From that it generates approximately 50000 lines of VHDL code for the case of a cochlea filter with 88 biquadratic sections.

5 Results

The cochlea implementation was tested on an Annapolis “Wildstar” Reconfigurable Computing En-

gine [23], a PCI-based reconfigurable computing platform containing three Xilinx Virtex XCV1000-BG560-6 FPGAs. The cochlea implementations were verified by comparing Synopsys VHDL Simulator simulations with the results produced by a floating-point software model. Synthesis and implementation were performed using Synopsys FPGA Express 3.5 and Xilinx Foundation 3.3i respectively.

5.1 Tradeoffs among Wordlength, Width of DA ROM and Precision

A series of cochlea implementations, with wordlengths from 10 to 32 bits and DA ROM width from 10 to 24 bits, were generated in order to present the tradeoffs among wordlengths, widths of DA ROMs and precisions. The coefficients of these implementations were obtained from the Auditory Toolbox using the MATLAB command `DesignLyonFilters(16000, 8, 0.25)`, which specifies a 16 kHz sampling rate, $Q = 8$ and a spacing which gives 88 biquadratic filters.

In order to present the improvement in precision with increasing wordlengths and ROM width, the frequency responses of several different fixed-point implementations are plotted in Figure 5. Figure 6 shows impulse and frequency responses obtained from a software floating-point implementation and a hardware 16-bit wordlength and 16-bit ROM width implementation.

It can be observed that the filter accuracy gradually improves with increasing wordlength or ROM width. When wordlengths or ROM widths are too small, there are significant quantization effects that may result in oscillation (as in the 12-bit wordlength implementations) or improper frequency responses at certain frequency intervals (as in the 12-bit DA ROM implementations). With 24-bit wordlength and 16-bit ROMs for example, the total quantization error is -39.46 dB, which is sufficient for most speech applications. Figure 7 shows the trend of improved quantization error with increasing wordlength and ROM width.

Area requirements, maximum clock rates and maximum sampling rates of these implementations on an Xilinx Virtex XCV1000-6 FPGA, as reported by the Xilinx implementation tools, are shown in Tables 2 and 3. For each implementation, a timing constraint, determined by the corresponding wordlength and ROM width, was supplied to the tools. An Xilinx XCV1000 FPGA has 12288 slices and the largest currently available parts, XCV3200E have 32448 slices. As a bit-serial architecture was employed, the effective

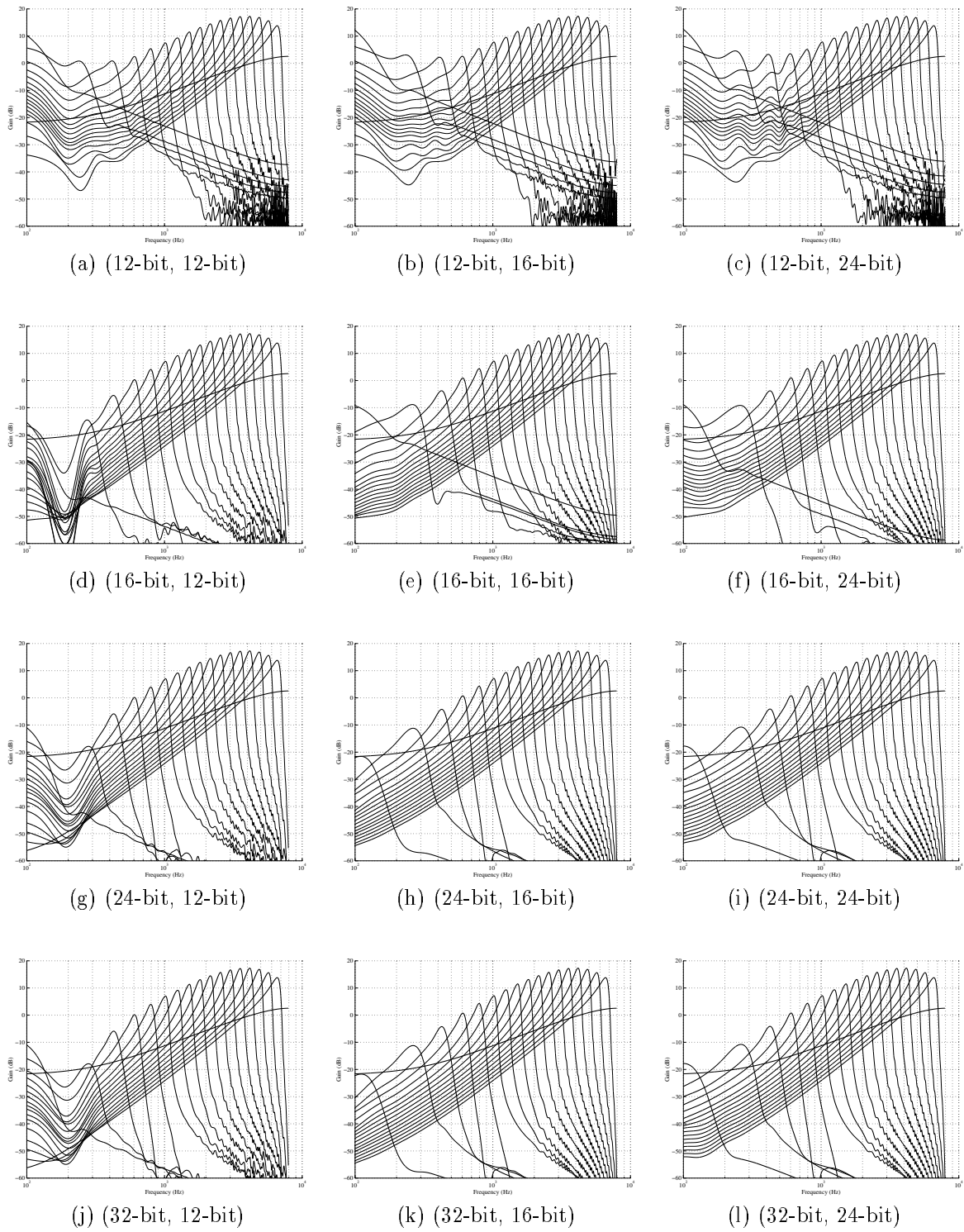
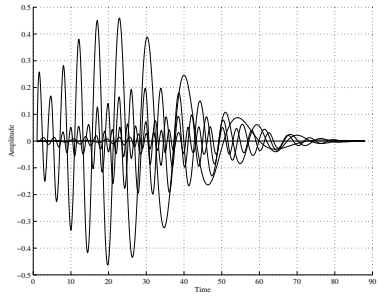
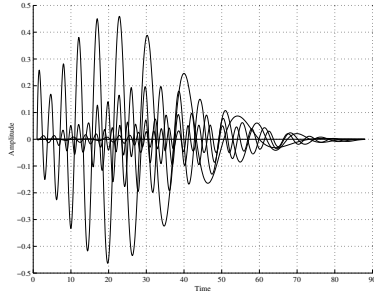


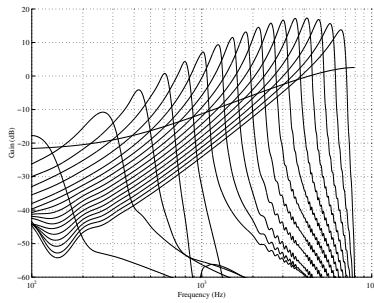
Figure 5: Frequency responses of cochlea implementations with different wordlength and width of ROMs (wordlength, ROM width).



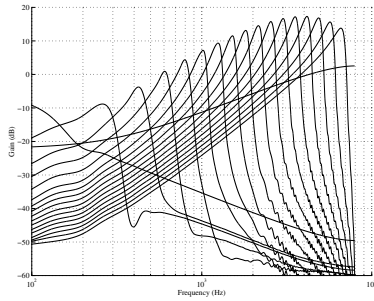
(a) Impulse response (software)



(b) Impulse response (hardware)



(c) Frequency response (software)



(d) Frequency response (hardware)

Figure 6: Impulse response of (a) a software floating-point implementation and (b) the hardware 16-bit wordlength, 16-bit ROM width implementation. Frequency response of (c) the software floating-point implementation and (d) hardware 16-bit wordlength, 16-bit ROM width implementation.

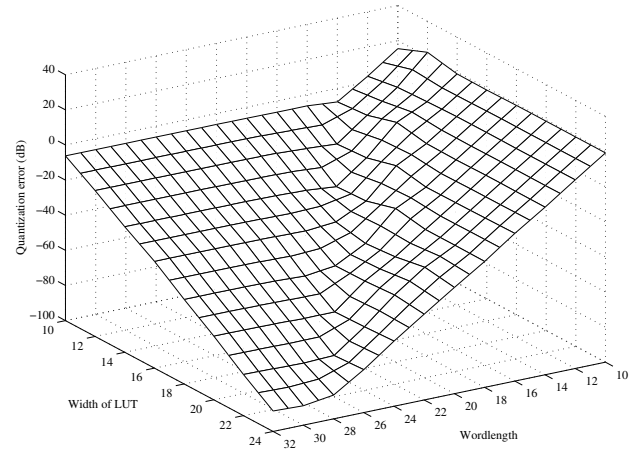


Figure 7: Mesh plot showing the quantization errors of implementations with varying wordlengths and DA ROM widths.

Wordlength	ROM Width			
	12-bit	16-bit	20-bit	24-bit
12-bit	5770	6582	7440	8340
16-bit	6160	6800	7589	8515
20-bit	6914	7343	7874	8602
24-bit	7620	8048	8578	9106
28-bit	8288	8748	9278	9805
32-bit	9297	9716	10245	10771

Table 2: Area requirements of an 88-section cochlea implementation of different wordlengths and ROM width (number of slices).

sampling rate of the implementations are their maximum clock rates divided by their wordlengths. With increasing wordlength or ROM width, an increase in area requirement and a general trend of decreasing maximum clock rate and sampling rate were observed.

5.2 Application to Speech Processing

A 24-bit wordlength, 16-bit DA ROM implementation was used to construct a cochleagram display application. This implementation was chosen because it is the smallest implementation that does not oscillate (refer to Figure 5 and Table 2).

The design of the cochleagram display is shown in Figure 8. The host PC writes input data into a dual-port BlockRAM (256 × 32-bit synchronous RAM) which passes through a parallel to serial converter

Word-length	ROM Width			
	12-bit	16-bit	20-bit	24-bit
12-bit	70.89, 5.91	68.03, 5.67	64.94, 5.41	63.91, 5.33
16-bit	67.74, 4.23	67.38, 4.21	61.60, 3.85	60.24, 3.77
20-bit	66.87, 3.34	65.60, 3.28	61.02, 3.05	59.79, 2.99
24-bit	66.15, 2.76	65.58, 2.73	60.53, 2.52	57.08, 2.38
28-bit	65.00, 2.32	63.13, 2.25	59.41, 2.12	57.01, 2.04
32-bit	64.96, 2.03	63.63, 1.99	58.00, 1.81	56.55, 1.77

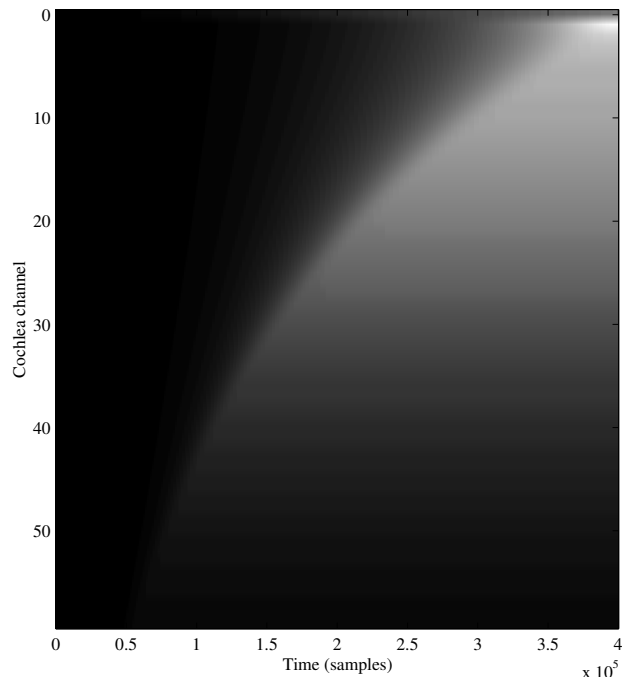
Table 3: Maximum clock rates and corresponding sampling rates of 88 section cochlea implementations for different wordlengths and ROM width (maximum clock rate (MHz), maximum sampling rate (MHz)).

and enters the cochlea core. Each of the outputs of the cochlea core undergoes serial to parallel conversion followed by half-wave rectification (to model the functionality of the inner hair cells). The outputs are accumulated to integrate its value over 256 samples. The accumulated output is read by the PC and displayed to obtain a cochleagram.

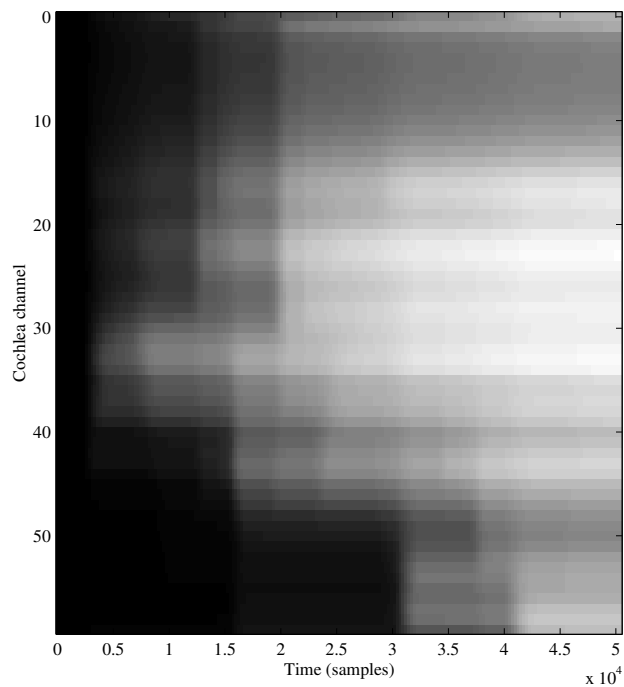
The cochleagram display was tested with several different inputs. Figure 9 shows the cochleagrams produced from swept-sine wave and the Auditory Toolbox’s “tapestry” inputs, the former being a 25 second linear chirp and the latter the speech file of a woman saying “a huge tapestry hung in her hallway”.

In addition to the cochlea model, the cochleagram display consists of half-wave rectifiers, accumulators and interface. Due to limited hardware resources on a Xilinx XCV1000-6 FPGA, only the first 60 out of the 88 cochlea sections were used in order to reduce area requirements. The resultant cochleagram display requires 10344 slices and can be clocked at 52.51 MHz, yielding a sampling rate of 2.19 MHz (or 137 times faster than real time performance). Including software and interfacing overheads, the measured throughput on the “Wildstar” platform was 238 kHz. As a comparison, the auditory toolbox achieves a 64 kHz throughput on a Sun Ultra-5 360 MHz machine.

It is interesting to compare the FPGA-based cochleagram system with a similar system developed in analog VLSI by Lazzaro et. al. in 1994 [5]. Using a 2 μm CMOS process, they integrated a 119 stage silicon cochlea (with a slightly more sophisticated hair cell model), non-volatile analog storage and a sophisticated event-based communications protocol on a single $3.6 \times 6.8 \text{ mm}^2$ chip with a power consumption of 5 mW. The analog VLSI version has improved density



(a) Swept-sine wave input



(b) “tapestry” input

Figure 9: Cochleagrams of (a) swept-sine wave and (b) “tapestry” inputs. The former has 400000 samples while the latter has 50380 samples.

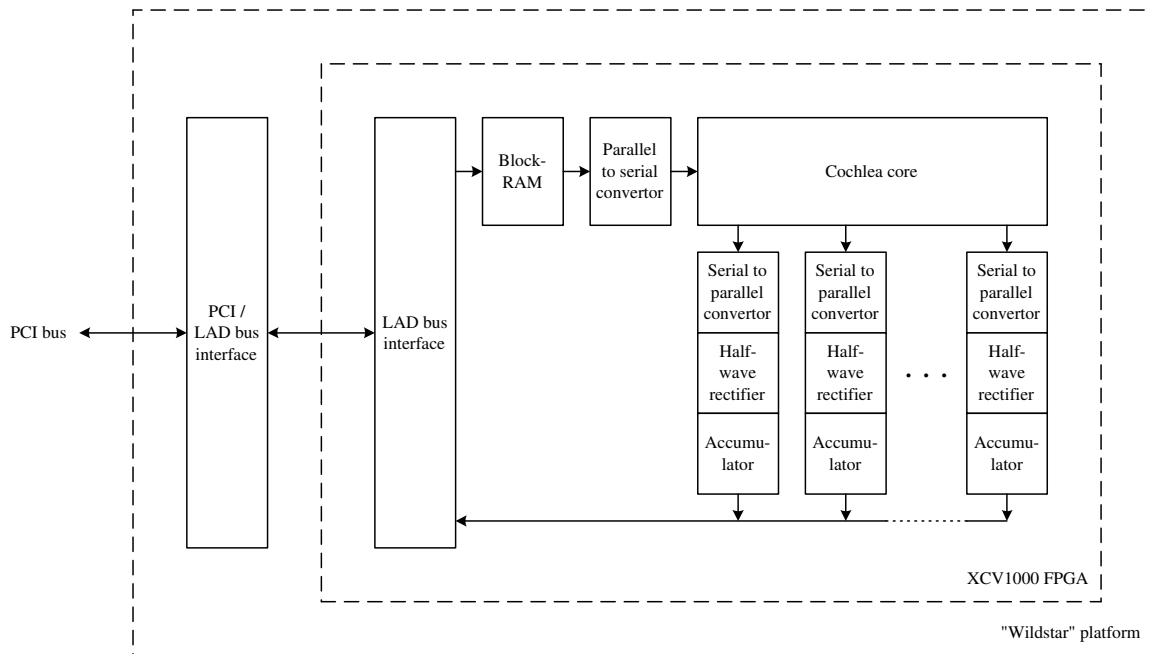


Figure 8: System architecture of the cochleagram display.

and power consumption compared with the FPGA approach. However, the FPGA version is vastly simpler; easier to modify; has a shorter design time; and is much more tolerant of supply voltage, temperature and transistor matching variations. Although qualitative results are not available, it is expected the FPGA version also has better filter accuracy; can operate at higher Q without instability; and has a wider dynamic range.

We believe that there are many applications of the FPGA cochlea, some including for audio compression, speech recognition, audio and speech visualization, models of human auditory localization, models of bat localization etc. Our next application will be to demonstrate the feasibility of an FPGA based neuromorphic isolated wordspotting system which uses the FPGA cochlea for preprocessing.

6 Conclusion

FPGAs provide a very flexible platform for the development of neuromorphic circuits and offer advantages in terms of faster design time, faster fabrication time, wider dynamic range, better stability and simpler computer interface over analog VLSI implementations.

A parameterized FPGA implementation of an electronic cochlea was developed that can be used as a building block for many systems which model the human auditory pathway. This electronic cochlea demonstrates the feasibility of incorporating large neuromorphic systems on FPGA devices. Neuromorphic systems employ parallel distributed processing which is well suited to FPGA implementation, and may offer significant advantages over conventional architectures.

Acknowledgements

The work described in this paper was supported by a direct grant from the Chinese University of Hong Kong (Project code 2050240), the German Academic Exchange Service and the Research Grants Council of Hong Kong Joint Research Scheme (Project no. G_HK010/00).

References

- [1] C. Mead, *Analog VLSI and Neural Systems*. Addison Wesley, 1989.

- [2] R. F. Lyon and C. Mead, "An analog electronic cochlea," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1119–1134, July 1988.
- [3] J. P. Lazzaro and C. A. Mead, "Silicon models of auditory localization," *Neural Computation*, vol. 1, pp. 47–57, 1989.
- [4] J. P. Lazzaro and C. A. Mead, "Silicon models of pitch perception," in *Proceedings National Academy of Sciences*, vol. 86, pp. 9597–9601, 1989.
- [5] J. P. Lazzaro, J. Wawrznek, and A. Kramer, "Systems technologies for silicon auditory models," *IEEE Micro*, vol. 14, no. 3, pp. 7–15, 1994.
- [6] A. van Schaik and R. Meddis, "Analog very large-scale integration (VLSI) implementation of a model of amplitude-modulation sensitivity in the auditory brainstem," *Journal of the Acoustical Society of America*, vol. 105, pp. 811–821, February 1999.
- [7] C. A. Mead, X. Arreguit, and J. Lazzaro, "Analog VLSI model of binaural hearing," *IEEE Transactions on Neural Networks*, vol. 2, pp. 230–236, 1991.
- [8] J. P. Lazzaro, J. Wawrznek, and R. P. Lippmann, "A micropower analog circuit implementation of hidden markov model state decoding," *Journal of Solid State Circuits*, vol. 32, no. 8, pp. 1200–1209, 1997.
- [9] R. F. Lyon, "Analog implementations of auditory models," in *Proceedings of the DARPA Workshop on Speech and Natural Language*, Morgan Kaufmann, 1991.
- [10] L. Watts, D. A. Kerns, R. F. Lyon, and C. A. Mead, "Improved implementation of the silicon cochlea," *IEEE Journal of Solid State Circuits*, vol. 27, pp. 692–700, May 1992.
- [11] A. van Schaik, E. Fragnière, and E. Vittoz, "Improved silicon cochlea using compatible lateral bipolar transistors," in *Advances in Neural Information Processing Systems 8*, MIT Press, 1997.
- [12] J. C. Bor and C. Y. Wu, "Analog electronic cochlea design using multiplexing switched-capacitor circuits," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 155–166, 1996.
- [13] C. D. Summerfield and R. F. Lyon, "ASIC implementation of the Lyon cochlea model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 673–676, 1992.
- [14] S. C. Lim, A. R. Temple, S. Jones, and R. Meddis, "VHDL-based design of biologically inspired pitch detection system," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 922–927, 1997.
- [15] M. Brucke, W. Nebel, A. Schwarz, B. Mertsching, M. Hansen, and B. Kollmeier, "Digital VLSI-implementation of a psychoacoustically and physiologically motivated speech preprocessor," in *Proceedings of the NATO Advanced Study Institute on Computational Hearing*, pp. 157–162, 1998.
- [16] M. P. Leong, M. Y. Yeung, C. K. Yeung, C. W. Fu, P. A. Heng, and P. H. W. Leong, "Automatic floating to fixed point translation and its application to post-rendering 3D warping," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 240–248, April 1999.
- [17] M. P. Leong and P. H. W. Leong, "A variable-radix digit-serial design methodology and its applications to the discrete cosine transform," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2000. in review.
- [18] R. F. Lyon and C. Mead, *Electronic Cochlea*, ch. 16 in *Analog VLSI and Neural Systems*. Addison Wesley, 1989.
- [19] J. O. Pickles, *An Introduction to the Physiology of Hearing*. Academic Press, 1988.
- [20] G. R. Goslin, *A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance*. Xilinx, Inc., 1995. Application Note.
- [21] I. Xilinx, *The Role of Distributed Arithmetic in FPGA-based Signal Processing*, November 1996. <http://www.xilinx.com/appnotes/theory1.pdf>.
- [22] M. Slaney, *Auditory Toolbox: A MATLAB Toolbox for Auditory Modeling Work*. Interval Research Corporation, 1998. Technical Report #1998-010, Version 2.
- [23] Annapolis Micro Systems, Inc., *Wildstar Reference Manual*, 2000. Revision 3.3.