CAR-Lite: A Multi-Rate Cochlea Model on FPGA

Ram Kuber Singh, Ying Xu, Runchun Wang, Tara Julia Hamilton, André van Schaik
The MARCS Institute for Brain, Behaviour and Development,
Western Sydney University,
Kingswood, NSW 2751, Australia

Susan L Denham
School of Psychology,
Faculty of Health & Human Sciences,
Plymouth University,
Plymouth, Devon PL4 8AA, U.K.

***Abstract*— Filters in cochlea models use different coefficients to break sound into a two-dimensional time-frequency representation. On digital hardware with a single sampling rate, the number of bits required to represent these coefficients require substantial computational resources such as memory storage. In this paper, we present a cochlea model operating at multiple sampling rates. As a result, fewer bits are required to represent filter coefficients on hardware as opposed to all the filters operating at a single sampling rate. Additionally, with a 108-filter cochlea implementation, up to nine times fewer coefficients are used than a single sampling rate approach across all filter sections. We present an implementation of 108 filters in Matlab and on an Altera Cyclone V FPGA with a low logic level utilization of 2.57%. Our model can thus be extended to include other auditory processing models such as loudness, pitch perception and timbre recognition on a single FPGA.**

Keywords — Neuromorphic engineering; electronic cochlea; basilar membrane; inner hair cell; auditory nerve; FPGA; multi-rate sampling

I. INTRODUCTION

In a biological auditory pathway, the cochlea breaks sound into signals with multiple frequency components and transmits them to the brain to perform complex tasks such as distinguishing sounds based on loudness, pitch, timbre, etc. [1]. Models of the cochlea offer us working hypotheses to understand this sound analysis process and are also used in automatic speech and music recognition [2].

Cochlea models attempt to capture the processing characteristics of the basilar membrane (BM), inner hair cells (IHC) and auditory nerve (AN) spiking. The BM is a coiled trapezoidal structure that mechanically vibrates along its length at specific locations in response to frequency components present in sound [3]. This motion causes IHCs positioned along the length of the BM to move as well [4]. The IHCs transduce such mechanical movement into electrical spikes that are transmitted to the brain via the auditory nerve [5].

The dynamics of a biological cochlea can be defined by cochlea models of various configurations. Prominent configurations include models with a parallel filterbank [6], transmission-line architectures [7], and cascaded filters [8]. The cascade of asymmetric resonator (CAR) filter configuration [9], [10] characterizes most biological cochlea features at a faster processing speed when compared to other configurations [11] and enables real-time implementation.

We have previously implemented a 100-section CAR model on FPGA with 400 16-bit coefficients [12], [13] using a single audio sampling rate. We will demonstrate in the upcoming sections that, by using multiple audio sampling rates for a 108-filter implementation of the CAR model, the bit width of all 432 coefficients can be reduced from 16-bits to 8-bits and nine times fewer coefficients can be used than in [12], [13]. On the FPGA, this amounts to a smaller logic utilization than the CAR model implementation with single audio sampling rate [12]–[14]. This enables other more complex models mimicking auditory pathway characteristics to be implemented in real-time on a single FPGA, thereby providing an alternative sound analysis perspective in contrast to non-real-time cochlea models.

II. ASYMMETRIC RESONATOR FILTER

The CAR model [8], [15] uses a cascade filterbank to model sound wave propagation of the basilar membrane in the cochlea [16]. Each filter in the filterbank is known as an asymmetric resonator filter (ARF) [17], which is configured as a two-pole-two-zero filter with an infinite impulse response (IIR). The ARF has an asymmetric gain response defined by a shallow gradient before its center frequency (CF) and a steep gradient after CF.

Coefficients of the ARF placed at the start of the cascade filterbank result in a resonance at high frequency. ARFs placed thereafter, have coefficients resulting in decreasing CFs. The transfer function of a resonator filter [9] is:

$$\frac{Y}{X} = g\left(\frac{z^2 + (-2a + hc)rz + r^2}{z^2 - 2arz + r^2}\right) \tag{1}$$

Coefficients a and c are the real and imaginary components of a complex signal and are computed as follows:

$$a = \cos(2\pi f_c / f_s) \tag{2}$$

$$c = \sin(2\pi f_c / f_s) \tag{3}$$

where f_c is the center frequency of the filter and f_s is the sampling rate. Coefficient h controls the distance of zeros from the frequency of the poles and is set to the real component of the ringing frequency, c. r is the pole and zero radius in the z-plane and g is the overall unity gain at DC defined by:

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

$$g = \frac{1 - 2ar + r^2}{1 - (2a - hc)r + r^2} \tag{4}$$

III. CAR-LITE MODEL

Equation (1) is split into two first-order difference equations, which are then organized in a close-coupled form [9]. Calculation of the coefficients in CAR-Lite is identical to that for the CAR model [8], [9], [12] for the first 12 sections, i.e., for the highest octave, o_1. The CFs of 12 ARFs in an octave are equally spaced on a log scale spanning from $f_s/4$ to $f_s/8$. Coefficients a, c, and g of the 12 ARFs in o_1 are pre-computed while r is empirically set to ensure constant gain response. At the next lower octave, o_2, the same set of coefficients, namely a, c, g, and r, are reused but the sampling rate is halved. This technique ensures that ARFs in o_2 to have the same gain responses as the ARFs in o_1. However, the center frequencies (CF) of the ARFs in o_2 are halved with respect to the CFs of the ARFs in o_1 in accordance with sound wave propagation in BM in decreasing CFs. On FPGA, this technique is highly advantageous as only coefficients from ARFs in o_1 require storage. Furthermore, halving sampling rate can be implemented simply by ignoring every second input sample to o_2. We apply this technique to all other octaves, reducing the sampling rate according to:

$$f_S(n) = f_S(1) / \left(2^{n-1}\right) \tag{5}$$

where n is the octave index and $f_s(1)$ is the sampling rate at the highest octave, o_1. Fig. 1 illustrates a block diagram of the CAR-Lite model.

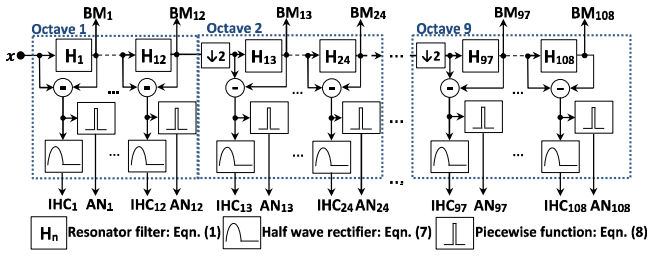

Fig. 1: CAR-Lite model with the sampling rate halved at the start of each octave.

The CAR filter outputs are low-pass filters with a resonant gain around CF, but they all have 0dB gain at DC as observed for the low frequency slopes in Fig. 3(a) and Fig. 3(b). This can be used to represent BM displacements. However, IHCs are sensitive to BM velocity, which means 6dB/octave low frequency slopes are required. One way to compute BM velocity is to implement a temporal differentiation step by taking the difference of the output at previous and current discrete times for the same section. However, this requires additional storage for previous time values. A simple solution is to take the difference between neighboring sections, since the output of the previous section is the input to the current one. Therefore, there is no need to store previous time CAR filter output. We apply this lateral section difference technique to the output of the CAR filters given by:

$$BMd(s,t) = Y(s,t) - Y(s-1,t) \tag{6}$$

where $Y(s,t)$ is the ARF output of section s at time t. As a result, we find that the low frequency slopes have the desired gains of 6dB/octave as observed in Fig. 4(a) and Fig. 4(b).

Since, we are aiming at the simplest and smallest implementation of a silicon cochlea, the inner hair cells (IHC) are modeled as simple half-wave rectifiers:

$$IHC(s,t) = \begin{cases} BMd(s,t), BMd(s,t) \geq 0 \\ 0, BMd(s,t) < 0 \end{cases} \tag{7}$$

Similarly, auditory nerve (AN) spikes are generated at positive zero crossings of BMd:

$$AN(s,t) = \begin{cases} 1, BMd(s,t) \geq 0.01 \,\&\, BMd(s,t-1) < 0 \\ 0, otherwise \end{cases} \tag{8}$$

Such binary spikes characterize periodicity in the input sound signal. An arbitrary amplitude threshold of 0.01 is introduced to avoid spike generation by sound signals with very low amplitude.

Note that the AN response in our model does not use the IHC output such as those in biologically inspired models [18] as we intend to extract periodicity information using the zero crossings of the BMd signals and the amplitude of IHC signals for future work with pitch detection. Also note that more complex IHC and AN models can be implemented on the FPGA, should they be required, as there are plenty of resources left, even on a small FPGA board.

IV. SOFTWARE SIMULATION

We simulated the CAR-Lite model with nine octaves totaling 108 sections in Matlab. The CFs of the filters span from 50 Hz to 24 kHz, which covers the range of human hearing. The sampling rate for the first octave, o_1, is set at 96 kHz and is halved for every octave down the cascade. Therefore, the second octave, o_2, operates at a sampling rate of 48 kHz, and the final octave, o_9, operates at 375 Hz sampling rate. We initially implemented the model using floating-point arithmetic to acquire a benchmark response and subsequently, converted the model to fixed-point numbers. For the fixed-point implementation, 8-bits coefficients and 16-bits signals are used to match the floating-point gain response within an error margin of 5%. Here, signal refers to input audio data, the BM output and the IHC output. The AN signal is a single bit signal. The fixed-point model was used as the reference model for FPGA implementation via Verilog.

V. FPGA IMPLEMENTATION

We used an Altera Cyclone V FPGA with a global clock speed of 250 MHz and an audio codec sampling rate set at 96 kHz. Fig. 2 displays the block diagram of the FPGA implementation of CAR-Lite. The model is characterized by two modules: a supervisor module and an equation module. The supervisor module invokes the equation module using time multiplexing, and manages the flow of computation by controlling resources. This includes selecting the cochlea section to be processed and transmission of its corresponding coefficients and data into the equation module. The equation

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

module implements resonator filter equation (1) as well as equations (6) to (8) for BMd, IHC, and AN segments respectively.

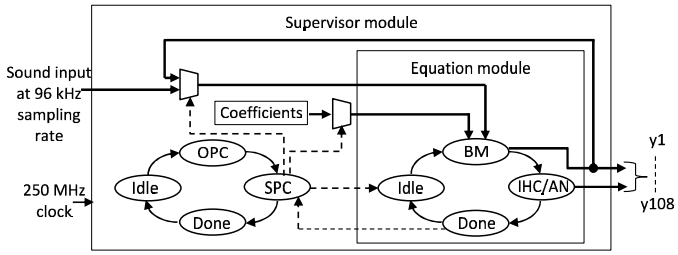

Fig. 2. FPGA implementation of CAR-Lite model.

A. *Supervisor Module*

The supervisor module oversees the running of the CAR-Lite model. A global clock rate of 250 MHz ensures that 108 sections are processed in 9.5µs, which is less than 10.42µs - the time interval between the arrivals of two successive audio samples at a rate of 96 kHz. Due to their low number, all 60 8-bits coefficients are stored on the FPGA. The module has a finite state machine (FSM) that controls which octave and which sections are processed at any time. It uses two states to achieve this: octave processing control (OPC) and section processing control (SPC). The former is used specifically for controlling multiple sampling rates across octave groups, whereas the latter is used for selecting coefficients and data entry into the equation module.

The OPC state implements sampling rate reduction by a factor of 2 for every octave from o_2, to o_9. These eight octave groups ignore every second input sample. A 9-bit register is used to administer the activation of octaves where 1-bit is allocated to an octave group i.e. bit 1 to o_1, bit 2 to o_2, etc. Two conditions are required to activate an octave as defined by a Boolean control logic:

$$B(n,t) = B(n-1,t) \cdot \overline{B(n,t-2^{n-1})} \qquad (9)$$

where B is a 9-bit register with the n^{th} bit corresponding to octave group o_n. The first of two conditions in (9) is that the preceding octave group, o_{n-1} had been activated to be processed. This ensures sequential octave group activation in the cascade. This is reinforced by the activation of o_n at previous time, $t\text{-}2^{n-1}$. In this final condition, the toggled bit generates the bypassing of every 2nd input sample to o_n. These conditions are applicable to all octave groups with the exception of bit 1 corresponding to o_1. This bit is set at the arrival of an input sample. Sections in octave groups that are not processed simply copy output values from $t\text{-}1$ to t.

When a new octave is required to run, the OPC state explicitly updates the section index in SPC state to be processed, which indicates the next section to be processed. Otherwise, the SPC tracks section indices to be processed by itself and only updates the OPC with already processed section indices. The SPC invokes the equation module with coefficients and input data for a section to be processed and once, the equation module finishes, the SPC reads back the output data and stores them in memory.

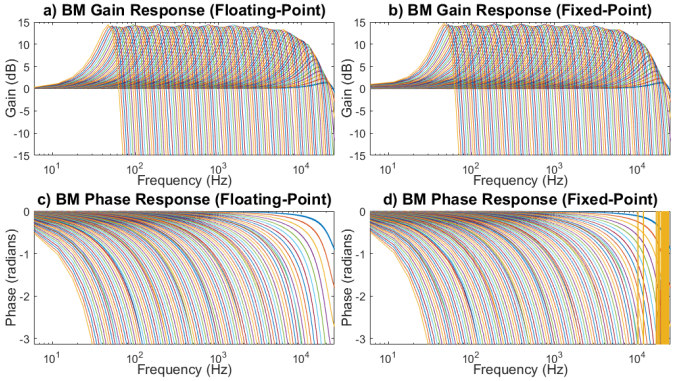

Fig. 3: BM gain and phase responses of CAR-Lite software floating-point and hardware fixed-point models. Each (colored) curve represents a single section, starting from section 108 (extreme left curve) to section 1 (extreme right curve).

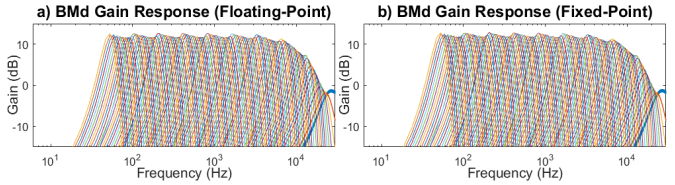

Fig. 4: BMd gain responses of CAR-Lite software floating-point and hardware fixed-point models. Each (colored) curve represents a single section, starting from section 108 (left most curve) to section 1 (right most curve).

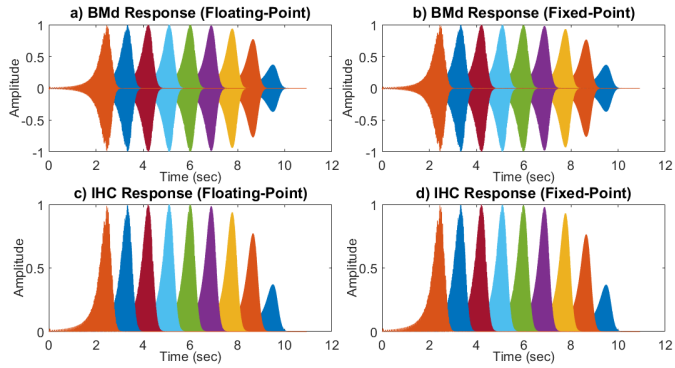

Fig. 5: Normalized basilar membrane (BMd) and inner hair cell (IHC) response based on log chirp input signal for software floating-point and hardware fixed-point models from sections 104 (extreme left signal), 92, 80, 68, 56, 44, 32, 20 and 8 (extreme right signal).

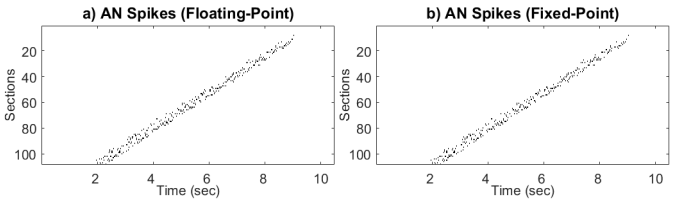

Fig. 6: Auditory nerve (AN) spike response based on log chirp input signal for software floating-point and hardware fixed-point models.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

B. Equation Module

An equation module comprises BM, IHC and AN segments of the cochlea arranged in the order of (1), (6), (7), and (8) respectively. A finite state machine (FSM) is used to achieve serial processing of these segments for a single section within the equation module. These equations are broken down into smaller parts comprising two operands. Each of these parts is processed within a discrete state of the FSM. A single section requires 22 states corresponding to 22 global clock cycles to complete processing. Individually, each BM, IHC, and AN segment takes 17, 1, and 4 clock cycles, respectively, to process. This results in a latency of 88 ns per section.

VI. Results

The hardware implementation of the CAR-Lite model was verified by comparing Altera Quartus Verilog simulations with results generated by the floating-point Matlab model. Fig. 3, Fig. 4, Fig. 5, and Fig. 6 illustrate the software floating-point and hardware fixed-point responses of the CAR-Lite model generated using a 10.93s log chirp in the range of 10 Hz to 48 kHz. The gain and phase responses of 108 CAR filter sections, i.e. basilar membrane (BM) responses, are displayed in Fig. 3. Coefficient r in (1) and (4) is responsible for the approximately constant gain responses for all sections except for the ones in the first two octaves at high frequencies, which have increasing gains as a result of the partial overlap of the resonances of neighboring sections. The effect of subjecting CAR filter output to lateral section difference leads to gain responses in Fig. 4, which is known also as BM difference (BMd) between neighboring sections.

Fig. 5 shows BMd, and IHC responses for sections 8, 20, 32, 44, 56, 68, 80, 92, and 104. The correlation coefficient [19], indicating the similarity of the floating-point and fixed-point responses of BMd, IHC and AN for all sections are 0.99, 0.99, and 0.98 respectively. These scores satisfy our expectation of matching responses between software floating-point and FPGA fixed-point CAR-Lite model implementations within 5% error margins. The few mismatches present are due to quantization errors in the fixed-point model.

In [12], 16-bits were used to represent fixed-point filter coefficients at a single sampling rate. Because of the use of a single sample rate, filter coefficients for the lowest frequencies were set close to either 0 or 1, needing a large number of bits to accurately represent them. In our model, filters operating at multiple sampling rates have enabled using only 8-bit coefficients for all filters.

The logic utilization in adaptive logic modules (ALM) on an Altera Cyclone V FPGA for the implementation was only 2.57%, with 747 registers used. With such low logic utilization, we can add more complex IHC and AN models as well as other sound processing algorithms, such as loudness, pitch and timbre auditory-based models to the model on a single FPGA. Furthermore, running models on a single FPGA reduces implementation complexities and data transmission constraints inherent in multiple FPGAs usage. Table 1 shows a comparison of FPGA resource utilization between CAR-Lite and selected other cochlea models on FPGA. CAR [12] and CAR-FAC [13] models have single audio sampling rates at 48 kHz and 32 kHz respectively. The number of registers utilized in [20] is smaller than our model, but our model accommodates more sections, which is able to create a higher resolution auditory image.

Table 1. Performance of cochlea filters on FPGA

Model	**FPGA**	***No. of filters***	***ALM utilization***	***Registers utilization***
Gambin et al. [20]	Xilinx XC3S500E	24	-	352
Leong et al. [21]	Xilinx XCV1000-6	88	-	6,800
CAR [12]	Xilinx Virtex-6	102	-	9,480
CAR-FAC [13]	Altera Cyclone V	100	18%	-
Chetan et al. [14]	Altera Cyclone V	70	3%	3899
This work	Altera Cyclone V	108	2.57%	747

VII. Conclusion

In this paper, we have presented a simplified cochlea model comprising a cascade of asymmetric resonator (CAR) filters. For an implementation of 108 sections, our CAR-Lite model with multiple sampling rates, has 8-bits coefficients rather than 16-bit ones used in our previous CAR model and uses 9 times fewer coefficients. We have simulated floating- and fixed-point implementations in Matlab, which was followed by an implementation on FPGA using Verilog. The logic utilization on an Altera Cyclone V FPGA is only 2.57%.

References

[1] S. Shamma and J. Fritz, "Adaptive Auditory Computations," *Curr. Opin. Neurobiol.*, vol. 25, no. 164–168, pp. 1–9, 2014.

[2] R. Meddis and E. A. Lopez-poveda, "Overview," in *Computational Models of the Auditory System*, R. Meddis, E. A. Lopez-Poveda, R. R. Fay, and A. N. Popper, Eds. Springer, 2010, pp. 1–6.

[3] J. O. Pickles, "The Cochlea," in *An Introduction to the Physiology of Hearing*, 3rd ed., Brisbane, Queensland: Emerald Group, 2008, pp. 25–72.

[4] J. O. Pickles, "Mechanisms of Transduction and Excitation in the Cochlea," in *An Introduction to the Physiology of Hearing*, 3rd ed., Brisbane, Queensland: Emerald Group, 2008, pp. 103–152.

[5] J. O. Pickles, "The Auditory Nerve," in *An Introduction to the Physiology of Hearing*, 3rd ed., Brisbane, Queensland: Emerald Group, 2008, pp. 73–101.

[6] E. A. Lopez-Poveda and R. Meddis, "A Human Nonlinear Cochlear Filterbank," *J. Acoust. Soc. Am.*, vol. 110, no. 6, pp. 3107–3118, 2001.

[7] S. Verhulst, T. Dau, and C. A. Shera, "Nonlinear time-domain cochlear model for transient stimulation and human otoacoustic emission," *J. Acoust. Soc. Am.*, vol. 132, no. 6, pp. 3842–3848, 2012.

[8] R. F. Lyon, "Cascades of two-pole–two-zero asymmetric resonators are good models of peripheral auditory function," *J. Acoust. Soc. Am.*, vol. 130, no. 6, p. 3893, 2011.

[9] R. F. Lyon, "The Cascade of Asymmetric Resonators," in *Human and Machine Hearing: Extracting Meaning from Sound*, Cambridge University Press, 2017, pp. 293–298.

[10] R. F. Lyon, "The CARFAC Digital Cochlear Model," in *Human and Machine Hearing: Extracting Meaning from Sound*, Cambridge University Press, 2017, pp. 293–298.

[11] A. Saremi, R. Beutelmann, M. Dietz, G. Ashida, J. Kretzberg, and S. Verhulst, "A Comparative Study of Seven Human Cochlear Filter Models," *J. Acoust. Soc. Am.*, vol. 140, no. 3, pp. 1618–1634, 2017.

[12] C. S. Thakur, T. J. Hamilton, J. Tapson, A. van Schaik, and R. F.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

Lyon, "FPGA Implementation of the CAR Model of the Cochlea," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 1853–1856.

[13] Y. Xu, C. S. Thakur, R. K. Singh, R. Wang, J. Tapson, and A. Van Schaik, "Electronic Cochlea : CAR-FAC Model on FPGA," in *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2016, pp. 564–567.

[14] C. S. Thakur, R. M. Wang, S. Afshar, T. J. Hamilton, J. Tapson, S. A. Shamma, and A. van Schaik, "Sound Stream Segregation : A Neuromorphic Approach to Solve the 'Cocktail Party Problem ' in Real-Time," *Front. Neurosci.*, vol. 9, no. September, pp. 1–10, 2015.

[15] R. F. Lyon, *Human and Machine Hearing: Extracting Meaning from Sound*. Cambridge University Press, 2017.

[16] R. F. Lyon, "Waves in Distributed Systems," in *Human and Machine Hearing: Extracting Meaning from Sound*, Cambridge University Press, 2017, pp. 219–236.

[17] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 1996.

[18] C. J. Sumner, E. A. Lopez-Poveda, L. P. O'Mard, and R. Meddis, "A Revised Model of the Inner-Hair Cell and Auditory-Nerve Complex," *J. Acoust. Soc. Am.*, vol. 111, no. 5, pp. 2178–2188, 2002.

[19] Mathworks, "corr2: 2-D correlation coefficient," 2017. [Online]. Available: http://au.mathworks.com/help/images/ref/corr2.html. [Accessed: 15-Oct-2017].

[20] I. Gambin, I. Grech, O. Casha, E. Gatt, and J. Micallef, "Digital Cochlea Model Implementation Using Xilinx XC3S500E Spartan-3E FPGA," in *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on*, 2010, pp. 946–949.

[21] M. P. Leong, C. T. Jin, and P. H. W. Leong, "Parameterized Module Generator for an FPGA-Based Electronic Cochlea," in *Field-Programmable Custom Computing Machines, 2001. FCCM '01. The 9th Annual IEEE Symposium on*, 2001.