Microprocessors and Microsystems 000 (2016) 1–10

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

A compact digital gamma-tone filter processor

Areli Rojo-Hernandez[a], Giovanny Sanchez-Rivera[a,*], Gerardo Avalos-Ochoa[a], Hector Perez-Meana[a], Leslie S. Smith[b]

[a] Instituto Politecnico Nacional ESIME Culhuacan, Av. Santana N 1000, Coyoacan, 04260 Distrito Federal, Mexico
[b] Computing Science and Mathematics University of Stirling, Stirling FK9 4LA, Scotland

ARTICLE INFO

Article history:
Received 14 January 2016
Revised 23 March 2016
Accepted 17 May 2016
Available online xxx

Keywords:
Auditory models
Cochlear implant processor
Gamma-tone Filter

ABSTRACT

Area consumption is one of the most important design constrains in the development of compact digital systems. Several authors have proposed making compact Cochlear Implant processors using Gamma-tone filter banks. These model aspects of the cochlea spectral filtering. A good area-efficient design of the Gamma-tone Filter Bank could reduce the amount of circuitry allowing patients to wear these cochlear implants more easily. In consequence, many authors have reduced the area by using the minimum number of registers when implementing this type of filter. However, critical paths limit their performance. Here a compact Gamma-tone Filter processor, formulated using the impulse invariant transformation together with a normalization method, is presented. The normalization method in the model guarantees the same precision for any filter order. In addition, area resources are kept low due to the implementation of a single Second Order Section (SOS) IIR stage for processing several SOS IIR stages and several channels at different times. Results show that the combination of the properties of the model and the implementation techniques generate a processor with high processing speed, expending less resources than reported in the literature.

1. Introduction

The development of digital artificial cochlear chips has attracted the interest of engineers for developing portable applications such as pitch detection, speech recognition and audio source localization on mobile devices, or for auditory prostheses [1]. These applications have used a model of the biological cochlea due to its capabilities for processing audio signals including natural sounds [2]. This cochlea functions as a transducer, converting the mechanical vibrations from the middle ear into electrical signals (auditory nerve spikes). These signals are sent to the human auditory system which responds to the information contained in the speech and audio signals.

Several studies show that the sound processing carried out by the cochlea, can be modeled using the over-complete Gamma-tone filter-bank, due to its resemblance to the human auditory system [2,3]. In addition, recently proposed mathematical models, reported in the literature, show that Gamma-tone filter banks designed using the impulse invariant transformation allow digital implementation of the analogue cochlea while employing reasonable computation with negligible distortion [4]. The hardware implementation of the cochlea, whether analogue or digital, is called an *artificial cochlea* chip or *silicon cochlea*, [5]. Because an efficient Cochlea chip is very important in several fields, the development of efficient cochlea chips has been an active research field. One of the first analogue silicon cochleae was developed by Lyon and Mead [4], using analogue VLSI 3 μm technology. It is reported that this cochlea chip, implemented using a cascade of 480 bi-quad filter sections, provides similar behavior than the human cochlea. A silicon cochlea which provides a good approximation of the human cochlea was proposed by Mandal et al. [6]. One of the most recent approaches was focused on building a bio-realistic analog CMOS Cochlea with high tunability and ultra-steep roll-off. The Chip response has high fidelity with respect to physiological experiments on mammalian cochlea and is 0.9 mm^2 in area and consumes $59.5 - 90.0$ μW [7].

Analogue implementations of artificial Cochlea chips, such as the above, are potentially efficient in terms of processing speed and area when compared with the digital implementations. However, the analogue approach is susceptible to other factors, such as temperature, transistor mismatch and power supply noise [7]. To solve these problems several digital implementations of cochlear chips have been proposed aiming for efficient sound processors with minimal area. One of the critical factors to be considered in

[*] Corresponding author.
E-mail address: giovas666@hotmail.com, giovanny.sanchez@upc.edu (G. Sanchez-Rivera).

http://dx.doi.org/10.1016/j.micpro.2016.05.010
0141-9331/© 2016 Elsevier B.V. All rights reserved.

the development of artificial cochlea chips is the precision of the variables of the system. This factor has been taken into account in the system proposed by Van Immersel and Peters [8], which supports the emulation of the biquadratic filters with arbitrary precision and uses a scaling of the fixed-point precision as a method to avoid overflow. In [9], a cochlea chip is designed to work in real-time data maximizing the use of Hardware resources.

This paper proposes a digital cochlear processor based on a digital Gamma-tone filter bank to improve the area compared to existing digital designs. Our strategy maximizes the utilization of a single SOS stage to implement any SOS stages and several channels employing a time multiplexing technique. Indeed, the use of the impulse invariant transformation and the application of the normalization method to the filter coefficients allow creating an efficient filter bank processor in terms of processing speed and area resources, respectively.

Evaluation results are provided to show the desirable properties of the proposed system. The rest of the paper is organized as follows. Section 2 presents the gamma-tone filter bank model. Section 3 presents the overview of the Gamma-tone Filter Processor. Evaluation results are provided in Section 4. Section 5 shows the improvement achieved by our proposal when compared with the current approaches. Finally Section 6 provides the conclusion of this work.

2. Review of the gamma-tone filter bank

Gamma-tone filters are defined in the continuous time domain, and can be mathematically modeled, in the discrete time domain, using different techniques of digital signal processing such as the impulse invariant transformation, the Z-matched transform and the bilinear transform. To select an appropriate transformation, in addition to the computational complexity, the introduced distortion must be considered. To this end, several works [8,10,11] show that the model employing the impulse invariant transformation provides the best performance because it has a low computational cost and lower distortion when compared with the Z-matched and bilinear transformations. Thus according to the previous published works [8,11] the impulse response of a Gamma-tone filter of order α, is given by the product of a gamma function multiplied by a cosine function, as follows:

$$\psi_{f_c}^{\alpha}(t) = \frac{1}{(\alpha-1)!}t^{\alpha-1}e^{-2\pi b_m t}\cos(2\pi f_c t)u(t) \tag{1}$$

where α is the filter order, b_m is the m'th filter bandwidth in Hz and f_{cm}, is the resonance frequency. Next using the Euler representation of cosine function in Eq. (1) and taking the Laplace transform of the resulting equation, after some manipulations it follows that the transfer function of the m'th gamma-tone and pass filter $H_m(s)$ can be represented in terms a cascade of α second order band pass filters as follows [8,10,11]:

$$H_m(s) = \left(\frac{K(s+4\pi b_m)}{(s+2\pi b_m - j2\pi f_{cm})(s+2\pi b_m + j2\pi f_{cm})}\right)^{\alpha} \tag{2}$$

Because the proposed system will be implemented in the discrete time domain, Eq. (2) must be transformed to the z-domain. To this end, the impulse invariant transform is used because it provides a discrete time version of Eq. (2) with less frequency distortion as compared with the bilinear and Z-matched transforms. Thus, applying the impulse invariant transform to Eq. (2), it follows that

$$H_m(z) = \left(\frac{2-2B_m z^{-1}}{1-2B_m z^{-1}+C_m z^{-2}}\right)^{\alpha}, \tag{3}$$

where

$$B = e^{-2\pi b_m T}\cos(2\pi f_{cm}T) \tag{4}$$

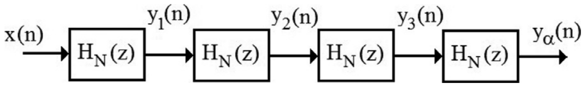


Fig. 1. Block diagram of the Gamma-tone structure implemented as a cascade of SOS IIR identical stages.

$$C_m = e^{-4\pi b_m T} \tag{5}$$

As shown in Eq. (3) the Gamma-tone filter transfer function, can be obtained as a cascade of α filters of second order with complex conjugated poles. Thus it is important to normalize the gain of each stage because it allows us to factorize the transfer function of the gamma-tone filter in α identical SOS stages. This avoids the recalculation the filter coefficients when the value of α is changed. To this end, because each stage represents a second order band pass filter transfer functions, consider the frequency response of a second stage evaluated in $2\pi f_{cm}$, which is given by

$$H_m(f_{cm}) = H_m(z)|_{Z=e^{j2\pi f_{cm}}} = \left.\frac{2-2B_m z^{-1}}{1-2B_m z^{-1}+C_m z^{-2}}\right|_{Z=e^{j2\pi f_{cm}}} \tag{6}$$

where f_{cm} is the resonance frequency. Thus from Eq. (6) it follows that

$$H_m(f_{cm}) = \frac{2-2B_m e^{-j2\pi f_{cm}}}{1-2B_m e^{-j2\pi f_{cm}} + C_m e^{-j4\pi f_{cm}}}, \tag{7}$$

whose magnitude is given by

$$|H_m(f_{cm})| = \sqrt{\frac{(2-2B_m\cos(2\pi f_{cm}))^2 + 4B_m^2\sin^2(2\pi f_{cm})}{(1-(2B_m+C_m)\cos(2\pi f_{cm}))^2 + (2B_m+C_m)^2\sin^2(2\pi f_{cm})}}. \tag{8}$$

Thus normalizing the magnitude of $H_m(f_{cm})$, from Eqs. (3) – 5 and 8 it follows that

$$H_m^N(z) = \left(\frac{A_m^N - B_m^N z^{-1}}{1-2B_m z^{-1}+C_m z^{-1}}\right)^{\alpha}, \tag{9}$$

where

$$A_m^N = \frac{2}{|H_m^N(f_{cm})|} \tag{10}$$

and

$$B_m^N = \frac{2e^{-2\pi b_m T}\cos(2\pi f_{cm}T)}{|H_m^N(f_{cm})|} \tag{11}$$

Next the normalized filter coefficients, A_N and B_N, provide the SOS stages with gain equal to one. Finally taking the inverse Z-transform of Eq. (9) we obtain the output of the gamma-tone filter which is given by

$$y_i(n) = A_i^N y_{i-1}(n) - B_i^N y_{i-1}(n-1) + 2B_i y_i(n-1) - C_i y(n-2) \tag{12}$$

The above synthesis method provides a systematic procedure that allows the implementation of the Gamma-tone filter by cascading identical second order stage independently on the value of α. It is important because it is known that the order of Gamma-tone filter can vary depending on the application and the type of signals to be processed.

Thus, because the filter can be implemented or represented by a cascade of SOS IIR band pass filters, if a given application requires the implementation of a Gamma-tone filter with $\alpha = 4$ it is necessary to use four SOS IIR band pass filters connected in cascade as shown in Fig. 1, where each block represents a second order IIR filter. Consequently the value of α is equal to the

Please cite this article as: A. Rojo-Hernandez et al., A compact digital gamma-tone filter processor, Microprocessors and Microsystems (2016), http://dx.doi.org/10.1016/j.micpro.2016.05.010

Table 1
Center frequency of 16 channel Gamma-tone filter.

Channels	1	2	3	4	5	6	7	8
F_c(Hz)	100	127.88	163.53	209.13	267.43	342	437.34	559.28
Channels	9	10	11	12	13	14	15	16
F_c(Hz)	715.21	914.61	1169.61	1495.70	1912.70	2445.97	3147.92	4000

number of SOS stages required for implementing the Gamma-tone filter. Therefore to implement the Gamma-tone filter it would be necessary to calculate the coefficients for each stage depending on the value of α. However using the mathematical model described above together with the normalization procedure, the calculations can be reduced because in this situation it is only necessary to calculate the coefficients of the first stage, because the same coefficients can be used in the synthesis of later stages, reducing the computational cost and storage.

2.1. Design parameters for the gamma-tone filter bank

According to Eq. (2) when the ear is stimulated with a given sound, different regions of the basilar membrane respond according to the frequency of the sound. These regions can be considered as a bank of cochlear filters along the basilar membrane. Because the Gamma-tone filter accurately models the basilar membrane, several studies have been carried out to determine its optimum parameters in terms of the Bark frequency scale. Thus depending on the sampling frequency, the maximum number of second order band pass filters is given by

$$N_{max} = \left\lfloor 7Ln\left(\frac{f_s}{1300} + \sqrt{\frac{f_s}{1300}}\right)^2 + 1 \right\rfloor \tag{13}$$

where $\lfloor x \rfloor$ denotes the integer part of x, the resonance frequency is given by

$$f_{cm} = 325 * \frac{e^{2m/7}}{e^{2m/7'}} \tag{14}$$

$$b_m = 25.1693\left(4.37\frac{f_{cm}}{1000} + 1\right) \tag{15}$$

is the bandwidth of the m'th band pass filter used in (11) to estimate B_m^N. Thus, taking into account that the human voice roughly has a range of frequencies from 100 Hz to 4000 Hz, designing a Gamma-tone filter bank with different values of α and 16 channels that emulates the cochlear human ear filter bank gives the resonance frequencies shown in the Table 1. It is worth noting that our approach potentially allows selecting frequencies to detect the onset of the sound [12], employing the same mechanisms described above Fig. 2.

3. Overview of the gamma-tone filter processor

The cochlea processor consists of a single SOS IIR stage, an array of internal Block RAMs and Control Unit, as shown in Fig. 3. A single SOS IIR stage is used as a part of the proposed processor to compute any filter order per channel and several channels employing the same precision in the whole system. Two design criteria have helped to build the compact processor with a single SOS IIR stage. The first is related to the technique of time multiplexing (well known as a virtualization concept) to process different virtualized SOS stages and virtualized channels at different instances of time using a single physical SOS during the sample time T_s. The second is focused on the utilization of Block RAMs. Our proposal maximizes the use of Block RAM, which is contained in the FPGA, to achieve the minimum consumption of Registers and LUTs. Modern FPGAs feature a large number of low-area BRAMs, which can store up to 36 Kbits each. In our approach, these BRAMs store the values of the internal variables (w_0 and w_1) of each SOS and the coefficients (A_N, B_N, B and C) for each channel. The number of BRAMs to store the internal variables (w_0 and w_1) of the virtualized IIR stages is a function of the number of channels and the size of each BRAM is determined by the number of virtual SOS stages. Finally, the Control Unit is responsible for synchronizing the reading and writing operations of the BRAMs enabling the internal variables (w_0 and w_1) and coefficients (A_N, B_N, B and C) of their corresponding virtualized SOS IIR stage to be processed by the physical SOS IIR stage.

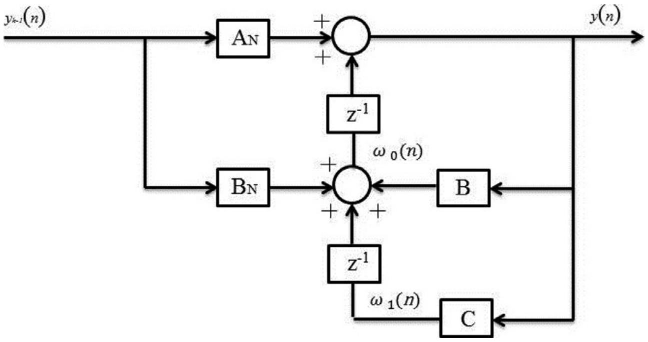


Fig. 2. Block diagram of the SOS IIR stage of the Gamma-tone filter.

The SOS IIR stage is composed of four 16 × 16 bit multipliers, two adders and two Flip-Flops, as shown in Fig. 4. The coefficients (A_N, B_N, B and C) are loaded from the BRAM coefficients into registers (A_N, B_N, B and C) in order to be updated when a particular channel is processed. Likewise, the flip-flops store the internal variables (w_0 and w_1) when the *load_ff* signal is set high (see Fig. 5). In this case, internal variables (w_0 and w_1) must be stored back after they are updated when a specific virtualized SOS stage is calculated.

As can be observed from Fig. 3, the Control Unit has an interface with the external CPU in order to load the coefficients to the BRAM. This data interface uses Gigabit-Ethernet. The user sends the coefficients, which are previously calculated, through an MS-DOS command script. Once the coefficients are loaded in the BRAM coefficients, the Control Unit distributes serially the value of the variables (w_0 and w_1) to each virtualized IIR stage reading its corresponding block of BRAM memory. The waveform diagram, which is obtained with the ModelSim ® software, shows the signals to control the BRAMs and the internal registers of the SOS IIR stage. As can be observed from Fig. 5, the time multiplexing technique can be applied because the sample duration is longer than the system clock time, allowing the calculation of every channel by its respective SOS IIR stages. Every SOS IIR stage is processed by the physical SOS IIR stage serially as shown in Fig. 5. For example, the right side of the waveform diagram of Fig. 5 shows the signal *Address_a*. This signal represents the address of the BRAM internal variables (w_0 and w_1) for each SOS of the first Channel (see Fig. 3). Once these SOS stages are processed the next internal variables (w_0 and w_1) of the SOS stages, which correspond to channel

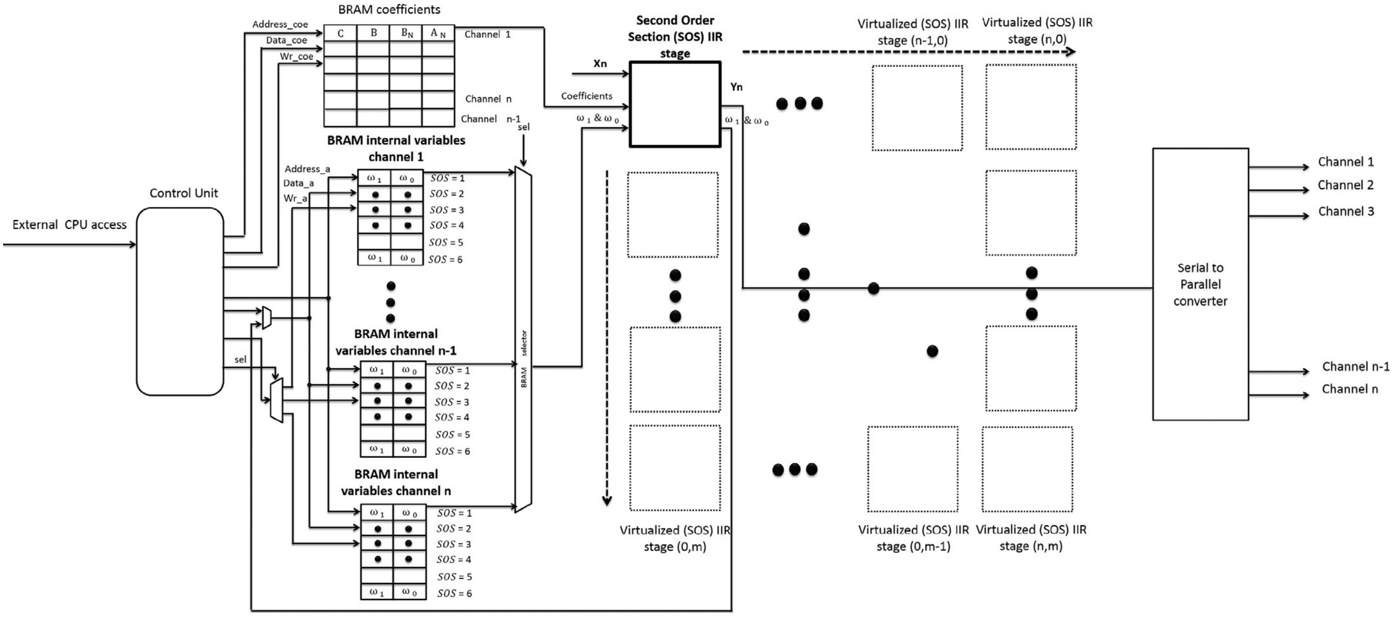

BRAM coefficients
Address_coe
Data_coe
Wr_coe
C
B
B N
A N
Channel 1
Channel n
Channel n-1
sel
BRAM internal variables channel 1
Address_a
Data_a
Wr_a
ω 1
ω 0
SOS = 1
SOS = 2
SOS = 3
SOS = 4
SOS = 5
SOS = 6
Control Unit
External CPU access
sel
BRAM internal variables channel n-1
BRAM internal variables channel n
BRAM selector
Second Order Section (SOS) IIR stage
Xn
Coefficients
ω 1 & ω 0
Yn
Virtualized (SOS) IIR stage (n-1,0)
Virtualized (SOS) IIR stage (n,0)
Virtualized (SOS) IIR stage (0,m)
Virtualized (SOS) IIR stage (0,m-1)
Virtualized (SOS) IIR stage (n,m)
Serial to Parallel converter
Channel 1
Channel 2
Channel 3
Channel n-1
Channel n

Fig. 3. Scheme of the proposed IIR biquadratic filter of order *n* with *m* channels .

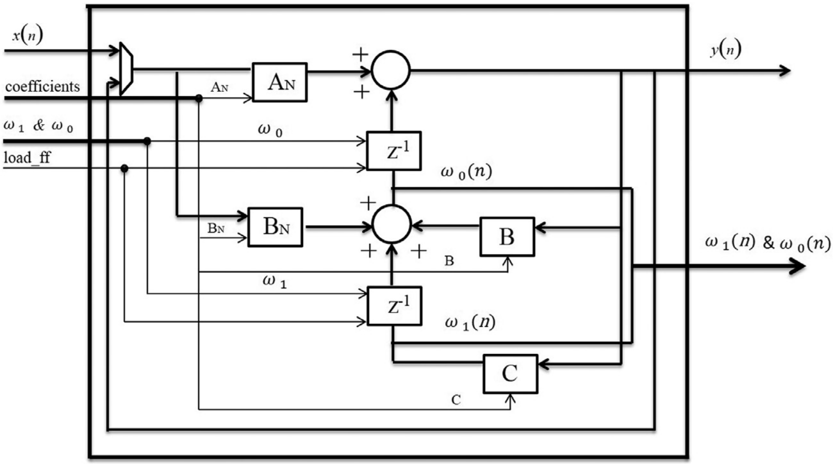


Fig. 4. Scheme of the structure of the SOS stage.

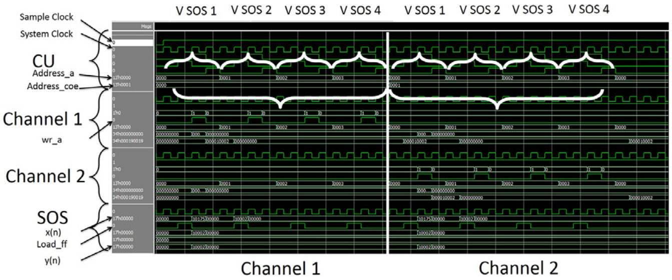


Fig. 5. Scheme showing the structure of the SOS stage.

2, are loaded by setting the signal *load_ff* high, and the *address_coe* is set to 1 to load the coefficients (A_N, B_N, B and C) into internal registers of the SOS IIR stage, as shown in Fig. 3.

An important factor to be defined is related to the maximum number of times or maximum number of virtualizations during a single Sampling Clock Cycle. The maximum number of virtual IIR stages can be found from Eq. (16):

$$T_s > (N_p * N_{SOS} * N_C) * T_{clk} \qquad (16)$$

where T_s is the sampling time, N_P are the number of clock cycles to process a single virtualized SOS IIR stage, N_{SOS} is the number of virtual SOS mudules, N_c is the number of channels, and T_{clk} is the time of the clock system.

4. Simulation and implementation and results

To verify the proposed method some filter banks were designed. The filters obtained were simulated in MatLab® to verify their response using 64 bit floating point precision and then implemented on a Field Programmable Gate Array (FPGA) employing 16 bit fixed point. The next sub-sections compare the response of the filter bank using these two representation methods.

4.1. MatLab simulation results

The filter bank was implemented as a cascade of SOS IIR filters. The center frequencies of the 16 channels Gamma-tone filter are shown in Table 1.

Fig. 6 shows the frequency response of a SOS Gamma-tone filter obtained in Matlab, where the filter coefficients were designed

Please cite this article as: A. Rojo-Hernandez et al., A compact digital gamma-tone filter processor, Microprocessors and Microsystems (2016), http://dx.doi.org/10.1016/j.micpro.2016.05.010

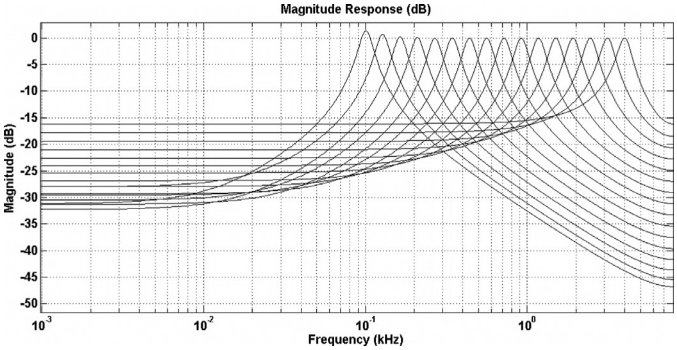


Fig. 6. Frequency response of the Gamma-tone filter using 64-bit floating point numbers.

using 64-bit floating point numbers and the sampling frequency was 16 KHz.

In order to do a simpler implementation on the FPGA the filter coefficients were converted to 16-bit integers, and the frequency response of this filter bank is shown in the Fig. 7. The results show that the integer representation does not change the response. To verify this a cascade of SOS was added to increase the order of the filters.

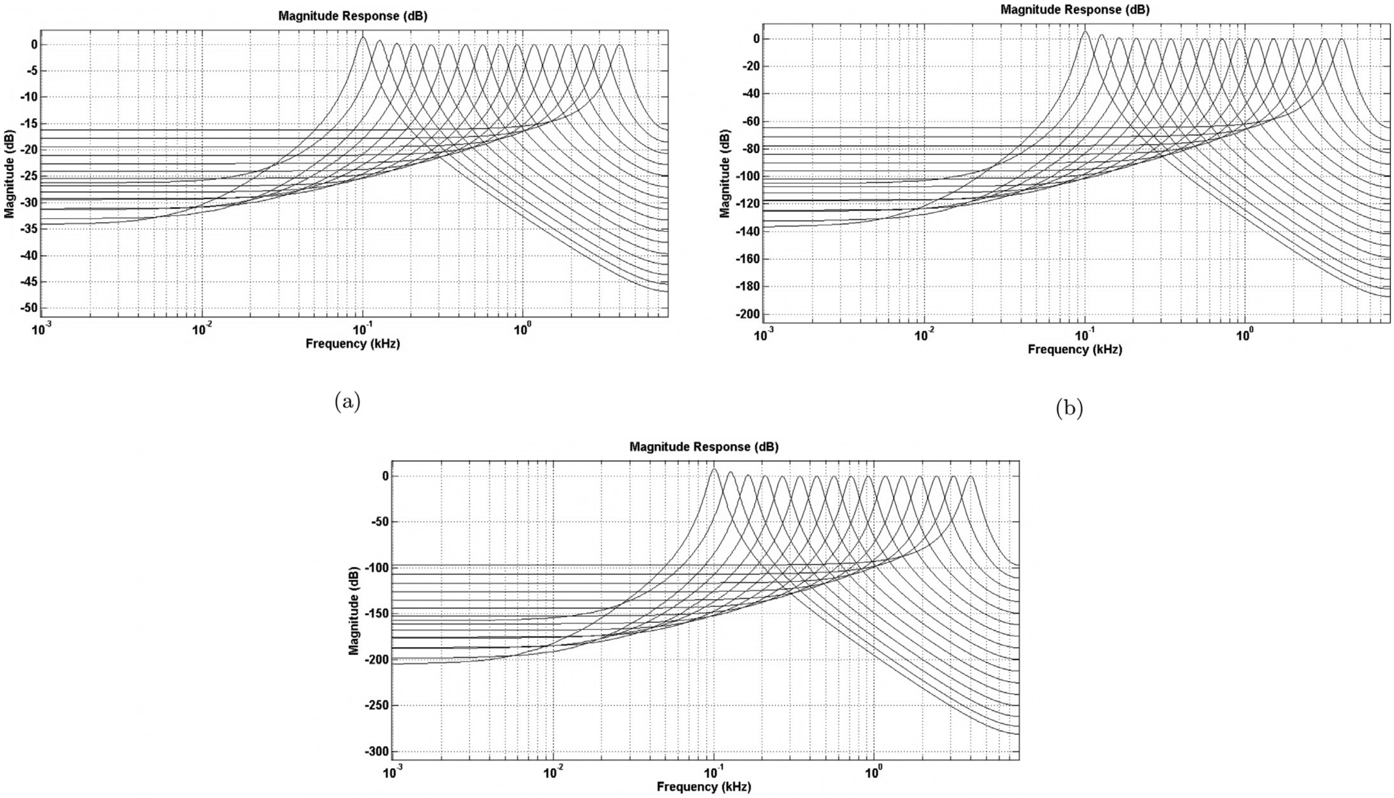
Fig. 7 shows the frequency responses of Gamma-tone filters with order 16 and 24 (equivalent to 8 SOS and 12 SOS), respectively. From the results it can be seen that increasing the filter order improves the selectivity, so that the stopband approaches its ideal characteristics.

The simulation results demonstrate that the filters with higher orders improve the stopband attenuation. Further, the integer representation of the coefficients does not change the response of the channels, and the center frequencies and bandwidth are not altered.

4.2. Implementation on a Kintex7 FPGA

The Filter Bank processor prototype was implemented on the *KC705* board kit which includes a *Xilinx Kintex7* FPGA. The Gammatone filter banks designed with 16-bit integer coefficients were implemented on the FPGA, and the frequency responses of the filters implemented were obtained with an *HP4395A* network analyzer. The fixed-point operators (multiplier and adder) were carried out by means of LUTs in order to increase the processing speed. Fig. 8 shows an arbitrary example that represents the response of the fourth channel centered at 209.13Hz for filters with order $\alpha = 4$, 16 and 24. As can be seen from this example, the higher order filters clearly provide much higher selectivity. Similarly, the selectivity response for other channels improves as the value of α increases. For a better visualization of the frequency responses of all the channels, the results obtained from the network analyzer were plotted in Matlab, the results are shown in the Fig. 9.

The results show that a simple implementation of Gamma-tone filter bank processor with higher order can be made. The only step required is to add cascade sections of the same coefficients, namely, increase the number of virtualized IIR stages. The frequency centers are not altered and the selectivity is improved,. The bands corresponding to the $F_c = 100\,$Hz and 127: 88 Hz exhibit a small gain that is less than 3 dB for the 16'th-order filter and 4 dB for the 24'th-order filter, which is not significant. The center frequencies of the other channels are around $-1\,$dB, which could be due to the connections between the board and the network analyzer.



(a)

(b)

(c)

Fig. 7. Frequency responses of 16 channel Gamma-tone filters, (a) Second-order filters, (b) 16'th-order filters, (c) 24'th-order.

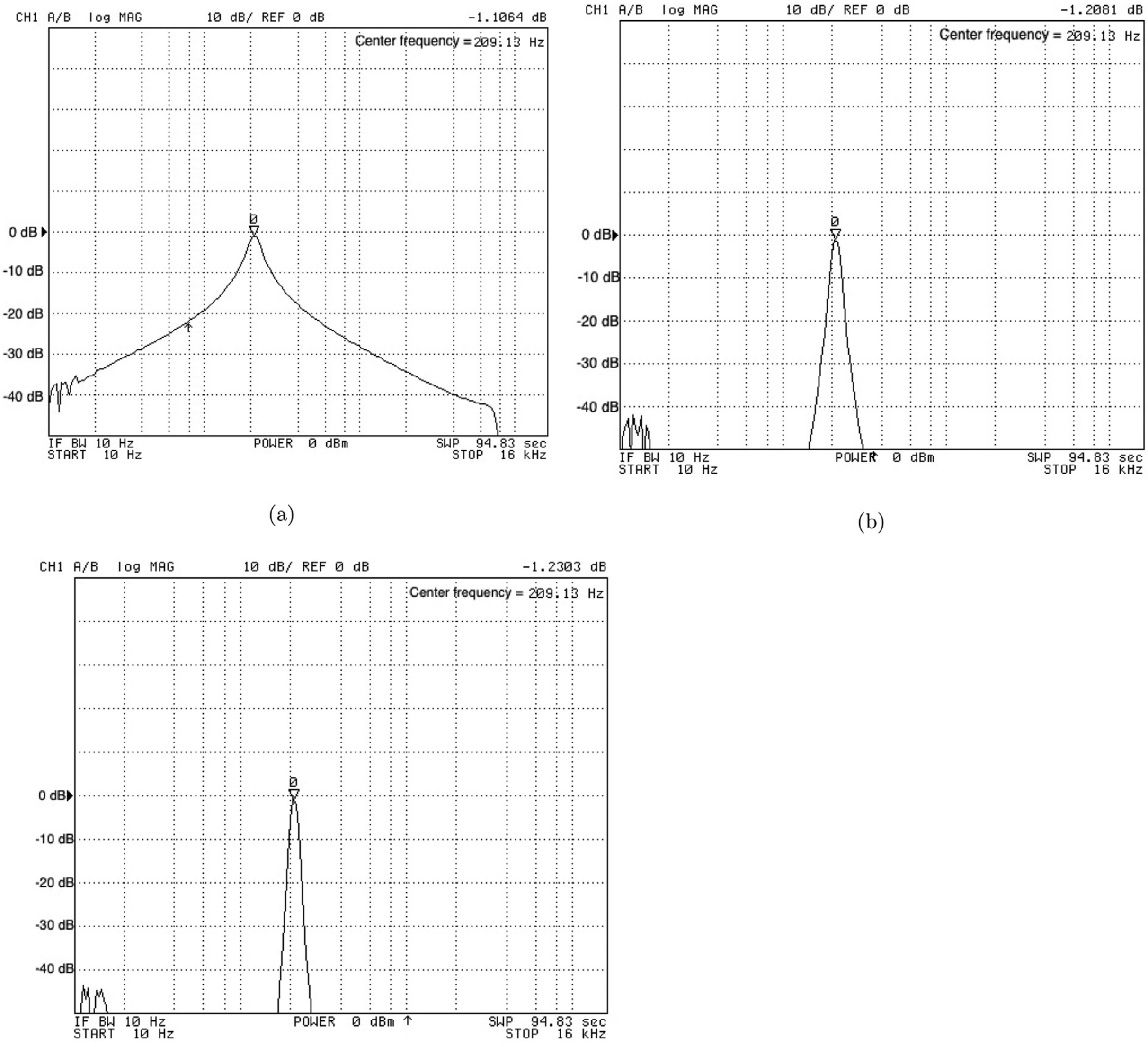


Fig. 8. Frequency responses of the fourth channel obtained with a network analyzer, (a) Second-order filters, (b) 16'th-order filters, (c) 24'th-order.

In addition to the implementation of the 16-bit fixed-point implementation, the proposed Gamma-tone filter bank processor was designed with 16-bit floating point and implemented on a Kintex-7 board in order to compare them. Clearly, the area required for implementing 16-bit floating point operators using LUTs and registers is greater than that for the 16-fixed-point operators (see Table 6). Fig. 10 shows the frequency response of an arbitrary channel that is centered at 209.13 Hz for filters with order $\alpha = 4$. As can be observed from Fig. 10, the frequency responses are quite similar using both techniques (16-bit fixed-point and 16-bit floating-point). It is worth noting that the frequency responses of the remaining channels were not dissimilar.

5. Related work

Our proposal was tested with a single SOS stage processing twelve virtualized SOS IIR stages per channel and sixteen channels on the *Kintex-7* board prototype. Theoretically, the processor is capable of computing twelve-SOS IIR stages per Channel and one hundred and fifty channels. These figures were calculated by replacing the above parameters in the Eq. (16), as shown in Table 2.

Table 2
Calculation of the performance of the current implementation and the theoretical figures.

The current version $(T_s = 62.5\ \mu\text{s},\ N_P = 4 clockcycles$ $N_{SOS} = 12$ SOS IIR stages, $N_c = 16$ and $T_clk = 8\text{ns})$	$62.5\ \mu\text{ s} > 6.14\ \mu\text{ s}$
Theoretical Values $(T_s = 62.5\ \mu\text{s},\ N_P = 4 clockcycles$ $N_{SOS} = 12$ SOS IIR stages, $N_c = 150$ and $T_clk = 8\text{ ns})$	$62.5\ \mu\text{ s} > 57.6\ \mu\text{ s}$

As can be observed from Table 2, the current implementation and the theoretical values met the condition. This illustrates the tradeoff of the proposed system between the number of SOS IIR stages and number of channels to be calculated by the processor during a sample clock cycle.

Table 3 shows a comparison between the conventional and our proposal in terms of the number of functional units and the number of registers. As can be observed from Table 3, the implementation of the SOS IIR stage further requires four registers while keeping the same number of functional units.

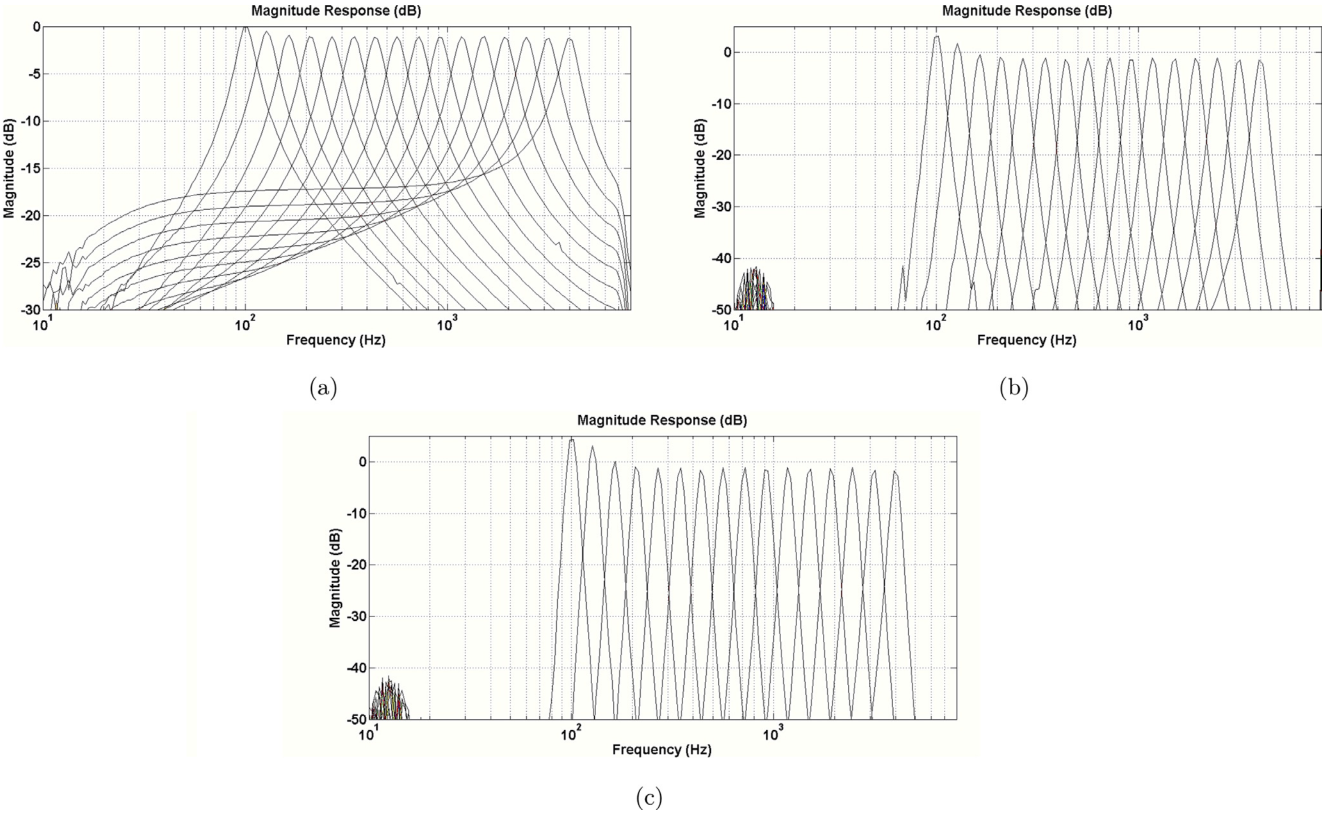


Fig. 9. Frequency responses of 16 bands of the Gamma-tone filter bank processor implemented on a Kintex-7 board, (a) Second-order filters, (b) 16'th-order filters, (c) 24'th-order.

Table 3
Comparison between conventional and the proposed SOS IIR stage.

Approach	Number of functional Units		Number of registers
	Adders	Multipliers	
Conventional	2	4	2
This work	2	4	6

Table 4
Comparison between conventional and the proposed SOS IIR stage implementing twelve-SOS IIR stages per channel and sixteen channels.

Approach	Number of functional Units		Number of registers
	Adders	Multipliers	
Conventional	384	768	384
This work	2	4	6

Table 4 compares the units required to implement the conventional SOS stages and the proposed system to process twelve-SOS IIR stages per channel and sixteen channels. The use of this technique allows improving the area consumption by a factor of 192, 192 and 64 for adders, multipliers and number of registers, respectively.

Several works attempt to emulate the artificial cochlea chip while consuming a minimum of hardware. In this section, we present a comparative analysis between our proposal and other approaches in terms of area consumption (LUTs and registers). The results of the obtained analysis are shown in Table 5.

The condition, which is given by Eq. (16), could be met for the case of requiring the implementation of 88 SOS stages per channel and seventeen channels, as shown in Table 8, expending 47.8 μs taking into account the following parameters: $T_s = 62.5\ \mu s$, $N_P = 4$ clock cycles, $N_{sos} = 88$ SOS IIR stages, $N_c = 17$, and $T_{clk} = 8ns$. Under this condition, the designed Gamma-tone filter processor could achieve a 94% and 99% reduction in consumption of LUTS and registers, respectively, when compared to [11]. In conclusion the Gamma-tone processor expends the same number of LUTs and registers to implement a range of channels from 16 to 150 and twelve-SOS IIR stage per channel meeting the condition given by Eq. (16). Table 8 shows the calculated processing time for implementing all the mentioned approaches [9–11,13] using our strategy. As can observed from Table 8, all the required processing time are under $T_s = 62.5\ \mu s$ making feasible the application of our strategy under the design parameters of mentioned approaches.

All the discussed implementations use fixed-point operators to emulate the Gamma-tone filter banks. Where 16-bit floating-point operators, which are built into LUTs and registers, are used, the area consumption is increased as shown in Table 6. This table shows the area consumption required to implement a single adder and a single multiplier under two formulaic representations (16-bit fixed-point and 16-bit floating point). Table 6 shows clearly that the required number of LUTs to implement the 16-bit floating-point multiplier and 16-bit floating-point adder is increased by a factor of 10 and 5 when compared with their counterparts, respectively. In addition, the floating-point operators require registers which are not needed for the fixed-point operators.

The number of LUTs and registers required to implement 16-bit floating-point twelve-SOS IIR stage per channel and sixteen channels with the conventional method is shown in Table 7. For this example, the required LUTs and registers are 733,824 and 1,136,952, respectively. These values have been calculated using the data of Tables 4 and 6, which exceed the available resources on the current FPGA (203,800 LUTs and 407,600 registers). Therefore, this makes their implementation on the Kintex-7 XC7K355T board infeasible.

Table 5
Comparison between the presented work and other approaches.

Approach	Number of channels	Word length (bits)	Number of SOS IIR stages	Order (α)	Number of LUTs	Number of registers
This work	16–150	16	12	24	646	85
Mishra [13]	12	8, 21	5	10	751	445
Brucke [9]	30	16, 18, 20, 22, 24, 26, 28, 30	2	4	2800	5600
Leong [11]	17	10, 12, 16, 24, 32	88	176	10,771	21,542
Dundur [10]	16	8, 12, 14, 16	1–4	2-8	20,699	823

Table 6
Comparison between 16-bit fixed-point operators and 16-bit floating-point operators in terms of area consumption.

Operator	Resources (16-bit floating-point operators)		Resources (16-bit fixed-point operators)	
	LUTs	Registers	LUTs	Registers
Adder	379	602	67	0
Multiplier	766	1,238	73	0

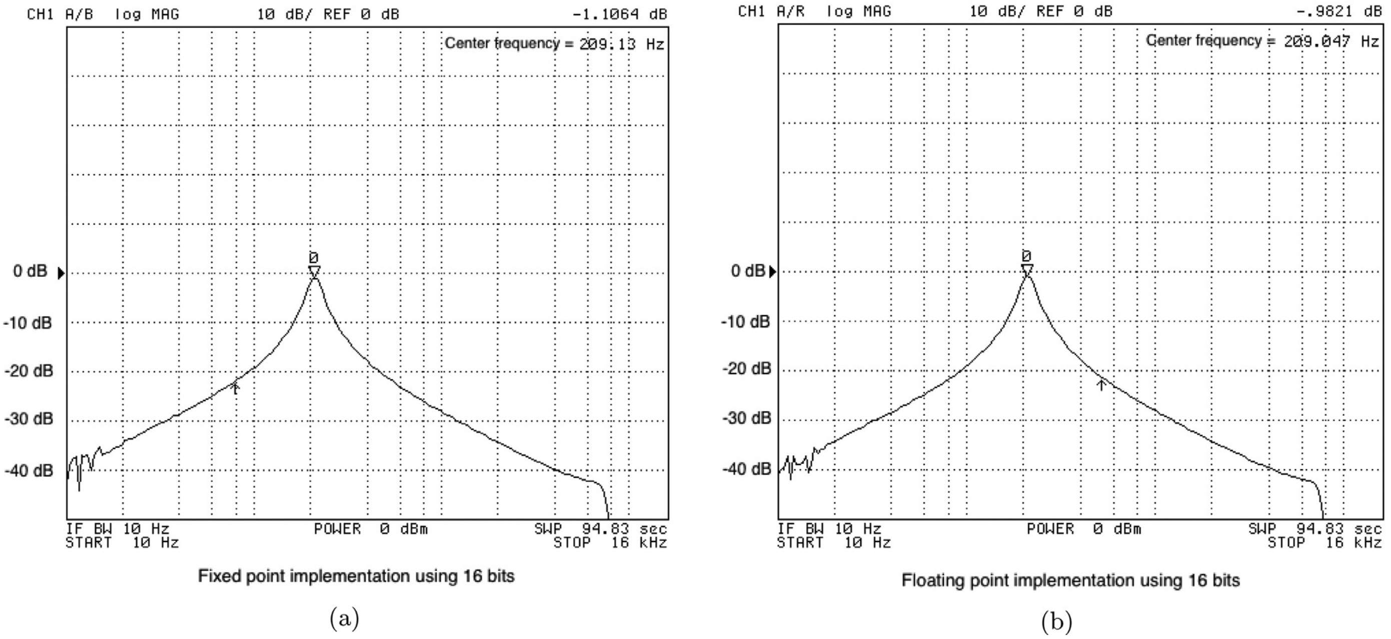


Fig. 10. Frequency responses of the fourth channel obtained with a network analyzer, (a) Second-order filters implemented with 16-bit fixed-point operators, (b) Second-order filters implemented with 16-bit floating-point operators.

Table 7
Comparison between conventional method (using 16-bit floating-point operators) and the proposed method (using 16-bit fixed-point operators) implementing twelve-SOS IIR stages per channel and sixteen channels in both cases.

Approach	Resources	
	LUTs	Registers
Conventional (16-bit floating-point)	733,824	1,136,952
This work (16-bit fixed-point)	426	0

Table 8
Processing time for implementing a variety of channels and SOS stages using the proposed strategy.

Number of channels	Number of SOS IIR stages	Processing time (μ s)
12	5	1.92
30	2	1.92
17	88	47.88
16	1–4	0.5, 1, 1.5, 2

6. Conclusions and future work

A virtualizable gamma-tone filter bank processor, which can be used for emulating human auditory models using the minimum constrains, has been developed. This electronic artificial cochlear chip demonstrates the efficient use of the available embedded resources in the FPGA with the help of the normalization method applied to the mathematical model to create compact systems. An important factor, which was taken into account to create this processor, is the precision of the variables of the system. The results show better performance regarding the selectivity of the filter bank when increasing the filter order using 16 bits. The proposed normalization of the Gamma-tone filter and the implementation strategy guarantee the minimum area consumption. Additionally, the power consumption of the proposed Gamma-tone filter processor was estimated using the VIVADO tool v2014.1. The obtained power consumption is around 5 mW to 250 mW depending on the number of BRAMs per channel. Part of our future work is the fabrication of a full custom implementation, which potentially allows further reduction of the power consumption of our processor.

Please cite this article as: A. Rojo-Hernandez et al., A compact digital gamma-tone filter processor, Microprocessors and Microsystems (2016), http://dx.doi.org/10.1016/j.micpro.2016.05.010

Acknowledgments

The authors would like to thank the Consejo Nacional de Ciencia y Tecnologia (CONACyT) and the Insitituto Politecnico Nacional of Mexico for the financial support for the realization of this work.

References

[1] V.H. Kumar, P.S. Ramaiah, Digital speech processing design for FPGA architecture for auditory prostheses, J. Comput. Sci. Eng. (JCSE) 6 (1) (2011) 33–41. URL http://sites.google.com/site/jcseuk/volumes/V6-I1-P33-42.pdf.

[2] E.C. Smith, M.S. Lewicki, Efficient auditory coding, Nature 439 (7079) (2006) 978–982, doi:10.1038/nature04485.

[3] J. Qi, D. Wang, Y. Jiang, R. Liu, Auditory features based on gammatone filters for robust speech recognition, in: 2013 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2013, pp. 305–308. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6571843.

[4] R.F. Lyon, C. Mead, An analog electronic cochlea, IEEE Trans. Acoust., Speech, Signal Proces. 36 (7) (1988) 1119–1134. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1639.

[5] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, R. Douglas, Silicon Cochleas, in: Event-Based Neuromorphic Systems, John Wiley & Sons, Ltd, 2015, pp. 71–90, doi:10.1002/9781118927601.ch4.

[6] S. Mandal, S.M. Zhak, R. Sarpeshkar, A bio-inspired active radio-Frequency silicon cochlea, IEEE J. Solid-State Circuits 44 (6) (2009) 1814–1828, doi:10.1109/JSSC.2009.2020465. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4982879.

[7] S. Wang, T.J. Koickal, A. Hamilton, R. Cheung, L.S. Smith, A bio-Realistic analog CMOS cochlea filter with high tunability and ultra-steep roll-off, IEEE Trans. Biomed. Circuits Syst. 9 (3) (2015) 297–311, doi:10.1109/TBCAS.2014.2328321. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6869048.

[8] L. Van Immerseel, S. Peeters, Digital implementation of linear gammatone filters: Comparison of design methods, Acoust. Res. Lett. Online 4 (3) (2003) 59–64. URL http://scitation.aip.org/content/asa/journal/arlo/4/3/10.1121/1.1573131.

[9] M. Brucke, A. Schulz, W. Nebel, Auditory signal processing in hardware, in: Field Programmable Logic and Applications, Springer, 1999, pp. 11–20. URL http://link.springer.com/chapter/10.1007/978-3-540-48302-1_2.

[10] R.V. Dundur, M.V. Latte, S.Y. Kulkarni, M.K. Venkatesha, Digital filter for cochlear implant implemented on a field-programmable gate array, in: Proceedings of World Academy of Science, Engineering and Technology, 33, Citeseer, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.307.4615&rep=rep1&type=pdf.

[11] M.-P. Leong, C.T. Jin, P.H. Leong, An FPGA-based electronic cochlea, EURASIP J. Appl. Signal Process. 2003 (2003) 629–638. URL http://dl.acm.org/citation.cfm?id=1283316.

[12] M.J. Newton, L.S. Smith, A neurally inspired musical instrument classification system based upon the sound onset, J. Acoust. Soc. Am. 131 (6) (2012) 4785–4798. URL http://scitation.aip.org/content/asa/journal/jasa/131/6/10.1121/1.4707535.

[13] A. Mishra, A.E. Hubbard, A cochlear filter implemented with a field-programmable gate array, IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process 49 (1) (2002) 54–60. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=996059.

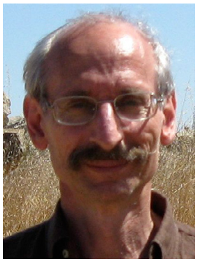
Please cite this article as: A. Rojo-Hernandez et al., A compact digital gamma-tone filter processor, Microprocessors and Microsystems (2016), http://dx.doi.org/10.1016/j.micpro.2016.05.010

Areli Rojo-Hernandez received the B.S. Degree in Electronics Engineering form the Metropolitan University of Mexico City and the M.Sc. Degree from the National Polytechnic Institute in 2011 and 2014, respectively. From April 2013 to March 2014 she was a visiting student at the University of Electro-Communications form Tokyo, Japan. In 2014 she became a Ph.D. student of the Communications and Electronics Ph.D. program of the National Polytechnic Institute where she is now a second year student. Her research interest is in the audio and speech processing fields.

Giovanny Sánchez received the M.S. degree at Instituto Politecnico Nacional, Mexico, in 2008, and the Ph.D. degree at Universitat Politecnica de Catalunya, Spain, in 2014. Currently, he is an Associate Professor in the Instituto Politecnico Nacional, Mexico. His main research interests include development of neuromorphics systems, encryption systems, auditory systems and genetic applications.

Juan Gerardo Avalos Ochoa Received the B.Sc. and the Ph.D. degrees in electronics and communications engineering from the National Polytechnic Institute, Mexico, in 2008 and 2014, respectively. He is currently working as a Professor in the Department of Computer Engineering, at the National Polytechnic Institute. His current research interests are signal processing and adaptive filtering applied to speech, audio, and acoustics.

Hector Perez-Meana received the M.S. degree from the University of Electro-Communications, Tokyo Japan, a Ph.D. degree in Electrical Engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1989. In 1981 he joined the Electrical Engineering Department of the Metropolitan University, Mexico City, where he was a Professor. From March 1989 to September 1991, he was a visiting researcher at Fujitsu Laboratories Ltd, Kawasaki, Japan. In February 1997, he joined the Graduate Department of The Mechanical and Electrical Engineering School, Culhuacan Campus (ESIME-C) of the National Polytechnic Institute of Mexico, where he is now a Professor. From 2006 to 2010 he was Dean of the Graduate Department of the ESIME-C; and from 1999 to 2006 and 2010 to 2013 Chair of the PhD program on Communications and Electronics Engineering of the IPN. In 1991 he received the IEICE excellent Paper Award, and in 1999 and 2000 the IPN Research Award. In 1998 he was Co-Chair of the ISITA'98, and general Chair of The Midwest Symposium on Circuit and Systems, 209. His principal research interests are adaptive systems, image processing, pattern recognition, information security and related fields. Dr. Perez-Meana is a member of the IEEE, IEICE, the National Researchers System of Mexico and the Mexican Academy of Science.



Leslie Smith (B.Sc. 1973, Ph.D. 1981) is Professor of Computing at the University of Stirling, Scotland. He has worked on neurally inspired computing techniques for many years, specializing in early auditory processing, neuromorphic systems and neuroinformatics in recent years. He is an SMIEEE, and was Head of Department for more than eight years, before returning to research and teaching late in 2013.