Folded Architecture for Digital Gammatone Filter Used in Speech Processor of Cochlear Implant

Rajalakshmi Karuppuswamy, Kandaswamy Arumugam, and Swathi Priya M.

Emerging trends in the area of digital very large scale integration (VLSI) signal processing can lead to a reduction in the cost of the cochlear implant. Digital signal processing algorithms are repetitively used in speech processors for filtering and encoding operations. The critical paths in these algorithms limit the performance of the speech processors. These algorithms must be transformed to accommodate processors designed to be high speed and have less area and low power. This can be realized by basing the design of the auditory filter banks for the processors on digital VLSI signal processing concepts. By applying a folding algorithm to the second-order digital gammatone filter (GTF), the number of multipliers is reduced from five to one and the number of adders is reduced from three to one, without changing the characteristics of the filter. Folded second-order filter sections are cascaded with three similar structures to realize the eighth-order digital GTF whose response is a close match to the human cochlea response. The silicon area is reduced from twenty to four multipliers and from twelve to four adders by using the folding architecture.

Keywords: Cochlear implant (CI), gammatone filter (GTF), folding, retiming, infinite impulse response (IIR) filter.

Manuscript received Apr. 10, 2012; revised Feb. 13, 2013; accepted Feb. 27, 2013.

Rajalakshmi Karuppuswamy (phone: +91 4222571477, laksharan@yahoo.com) and Swathi Priya M. (swathi.pry@gmail.com) are with the Department of Electronics and Communication Engineering, PSG College of Technology, Tamilnadu, India.

Kandaswamy Arumugam (kandaswamya@yahoo.com) is with the Department of Biomedical Engineering, PSG College of Technology, Tamilnadu, India.

http://dx.doi.org/10.4218/etrij.13.0112.0220

I. Introduction

A considerable amount of research has been conducted on the biological cochlea system because the cochlea serves as the front-end signal processor for all functions of the auditory nervous system, such as auditory localization, pitch detection, and speech recognition. The human cochlea acts as a transducer, converting the mechanical vibrations from the middle ear into neural electrical stimuli that is transmitted through the auditory system to the brain to recognize the information in the speech signal. It also provides spatial separation of frequency information, similarly to that of a spectrum analyzer [1]. The hardware model of the cochlea is known as the cochlear implant (CI), which is a stimuli-carrying electrode that is implanted inside the inner ear of a deaf patient along with a signal processor that is worn behind the ear. The hardware implementation of a CI is done in both the analog and digital very large scale integration (VLSI) domains. A CI proposed by Lyon and Mead in 1988 models the human cochlea as a cascaded series of 480 biquad filter sections in analog VLSI technology [2]. In 2009, Mandal and others [3] proposed an RF-based silicon cochlea, which is a device that mimics the functioning of the ear. Although analog electronic cochlea designs are potentially more efficient, as they use the physical current and voltage properties of transistors and avoid digitalization, digital designs have the advantage of being impervious to changes in the environment, such as in temperature or in power supply noise.

A digital VLSI electronic cochlea using a bit-serial second-order filter was reported by Summerfield and Lyon in 1992 [4]. In 1997, Lim and others implemented a first-order Butterworth bandpass filter as the cochlea filter in the pitch detection system [5]. Brucke and others, in 1998, implemented a VLSI speech

preprocessor, which uses gammatone filter (GTF) banks to mimic the cochlea [6]. In 2002, a tenth-order recursive cochlea was implemented by Mishra and Hubbard, using FPGA technology [7]. In 2003, an FPGA-based module generator that uses distributed arithmetic to implement the biquadratic filters was proposed by Leong and others [8]. Van Immerseel and Peters [9] performed the digital implementation of a GTF for a finite impulse response (FIR) filter design and for an infinite impulse response (IIR) filter design. As functions of order and computational cost of the filter for these digital filter designs, such filter properties as impulse response, magnitude, and phase response, were compared in detail to those of the analog GTF. The researchers in [9] also reported that filters designed with the base-band impulse invariant transformation gave the best overall approximation of the analog properties with reasonable computational cost. In 2008, Dundur and others proposed the design and implementation of the cochlear filter on a single device XC3S500 FPGA [10]. The filter gave a desirable fit to real-time data with efficiency of hardware usage. Kumar and Ramaiah implemented a digital speech processing algorithm on FPGA architecture for auditory prostheses [11] and improved the speech intelligibility for the speech processor.

II. Review of Gammatone Auditory Filter Bank

The most commonly used auditory filter is the GTF, which provides the spectral analysis of the signals in the cochlea. It is a sufficient model in the spectral analysis of human speech signals at a moderate sound level, which makes it suitable to simulate the signal processing in the auditory system and for sound processing in the CI. The GTF, originally introduced by Johannesma in 1972 [12] to describe the cochlea nucleus response, is the most frequently used auditory filter in cochlea modeling and speech recognition experiments. Its popularity within the auditory modeling community results from its ability to provide an appropriately shaped "pseudo-resonant" frequency transfer function that can be used to reasonably match measured physiological responses [13]-[16]. The gammatone impulse response is as follows:

$$g(t) = At^{N-1}e^{-bt}\cos(2\pi f_c t + \varphi). \quad (1)$$

The name "gammatone" was given by Aertsen and Johannesma [17] after observing the impulse response of the filter, which consists of a gamma-distribution envelope "$At^{N-1}e^{-bt}$" multiplied by a sinusoidal tone "$\cos(2\pi f_c t + \varphi)$," as represented by (1). The parameter A is the arbitrary gain factor, N is the order of the filter, b is the bandwidth of the filter, f_c is the center frequency, and φ is the starting phase. When N is equal to 4, b is 1.019 multiplied by the equivalent rectangular bandwidth (ERB). The ERB is a psychoacoustic measurement of the width of the auditory filter at each point along the cochlea; for the human auditory filter, it is approximated as

$$\mathrm{ERB} = 24.7\left(\frac{4.37 f_c}{1000} + 1\right). \quad (2)$$

Equations (1) and (2) together represent the gammatone auditory filter bank. By expanding the cosine term in the gammatone impulse response as the sum of the two exponential terms given in (3) and using the Laplace transform, the transfer function of the GTF is given by (4).

$$g(t) = \frac{A}{2}t^{N-1}\left(e^{j\varphi}e^{(-b+j\omega_c)t} + e^{-j\varphi}e^{(-b-j\omega_c)t}\right). \quad (3)$$

$$G(s) = \frac{A\Gamma(N)}{2}\left[\frac{e^{j\varphi}\left(s-(-b+j\omega_c)\right)^N + e^{-j\varphi}\left(s-(-b-j\omega_c)\right)^N}{\left[(s+b)^2+\omega_c^2\right]^N}\right]. \quad (4)$$

Expressing b and ω_c in terms of pole frequency ω_0 and quality factor Q (constant ratio of bandwidth and center frequency) and neglecting the term $\frac{A\Gamma(N)}{2}$ without loss of generality, we obtain $G(s)$ with a complex conjugate pole pair, as in (5).

$$G(s) = \left[\frac{e^{j\varphi}\left(s-\left(-\frac{\omega_0}{Q}+j\omega_0\sqrt{1-\frac{1}{4Q^2}}\right)\right)^N + e^{-j\varphi}\left(s-\left(-\frac{\omega_0}{Q}-j\omega_0\sqrt{1-\frac{1}{4Q^2}}\right)\right)^N}{\left[(s+b)^2+\omega_c^2\right]^N}\right]. \quad (5)$$

The transfer function of the GTF is approximated to the form in (6) without losing the characteristics of the filter response, as done by Slaney [18] and used by Van Immerseel and Peters in [9].

$$G(s) = \prod_{n=1}^{N} G_n(s) = \prod_{n=1}^{N}\frac{(s-s_n)}{(s-p_m)(s-p_m^*)}. \quad (6)$$

This shows that the N-th order GTF is implemented as a cascade of $N/2$ second-order filter sections, all having the complex conjugate pole pair (p_m, p_m^*) and one zero (s_n) for each section. The digital implementation of the auditory filter is realized as an FIR filter using Kaiser windowing [19], [20] and an IIR filter [8]. The FIR filter is implemented as a cascade of second-order sections of the analog GTF using a truncation criterion [9], [18], as given by (6). Using a higher-order GTF with a small truncation level, the FIR becomes noisy and the hardware complexity increases. Hence, in this study, the GTF is realized as an IIR filter [21]-[23].

The impulse invariant method is used to convert the analog

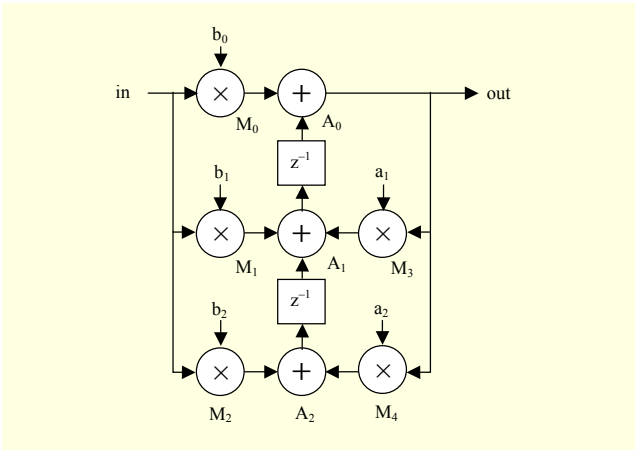

Fig. 1. IIR filter realization of second-order digital GTF.

GTF equation to a digital GTF, which increases the order to twice that of the former due to the presence of complex conjugate pole pairs. Therefore, the fourth-order analog GTF becomes an eighth-order digital GTF whose response matches with the biological cochlea. This eighth-order filter is implemented as a cascade of four biquad sections for each cochlea channel. The transformation converts two complex conjugate poles in each biquad into real coefficients, and the single zero makes the coefficient b_2 take the value zero in the digital domain. Hence, the digital transfer function for each biquad is represented in general terms as

$$G(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}. \quad (7)$$

The digital GTF is realized as an IIR filter, and its structure is given in Fig. 1, where M_0, M_1, M_2, M_3, and M_4 and A_0, A_1, and A_2 represent the multipliers and adders, respectively, and b_0, b_1, b_2, a_1, and a_2 are the coefficients. This filter is implemented as a bandpass filter for each channel of the CI, which in turn is provided with 16 to 24 channels [1], [2]. The conventional IIR filter realization of a second-order digital GTF requires five multipliers and three adders.

1. Structure of IIR Filter for Eighth-Order GTF

The direct form implementation is not usually advisable for high-order IIR filters, owing to its high sensitivity to coefficient quantization, which can lead to, among other things, severe stability problems, especially for the high-gain and tuned responses that we are targeting. Equation (8) represents the transfer function of direct form I of an IIR filter.

$$H(z)=\frac{b_0+b_1z^{-1}+....+b_{N_{b-1}}z^{-(N_{b-1})}}{1+a_1z^{-1}+.....+a_{N_{b-1}}z^{-(N_{b-1})}}. \quad (8)$$

The filter implemented by directly using the structure defined by (7) is sensitive to errors introduced by coefficient quantization and by computational precision limits. Also, the filter with an initially stable design can become unstable with increasing coefficient length. The filter order is proportional to the coefficient length. As the filter order increases, the filter becomes more unstable. The sensitivity of a filter can be reduced by writing (8) as a ratio of z transforms, which divides the direct form transfer function into lower-order sections or filter stages. By factorizing (8) into second-order sections, the transfer function of the filter becomes a product of second-order filter functions, as shown in (9).

$$H(z)=\prod_{k=1}^{N}\frac{b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2}}{1+a_{1k}z^{-1}+a_{2k}z^{-2}}. \quad (9)$$

The eighth-order digital GTF response closely matches the biological cochlea response [5], [6]. Equation (9) can be viewed as a cascade of four second-order filters. Figure 2 illustrates cascade filtering. The basic second-order section has five multipliers and three adders, and the cascading of four such similar structures leads to twenty multipliers and twelve adders for each cochlea channel. This large number of functional units increases the silicon area for the speech

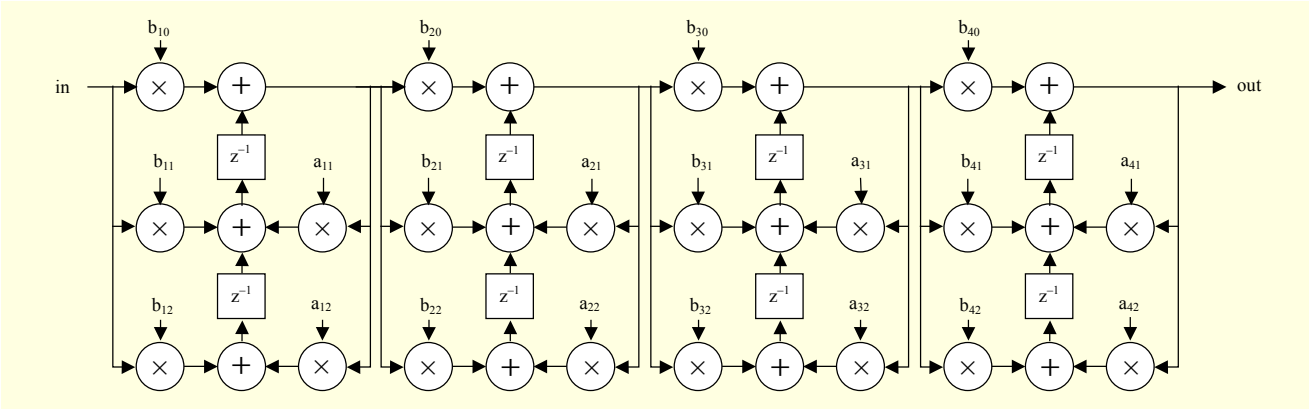

Fig. 2. IIR filter realization of eighth-order digital GTF.

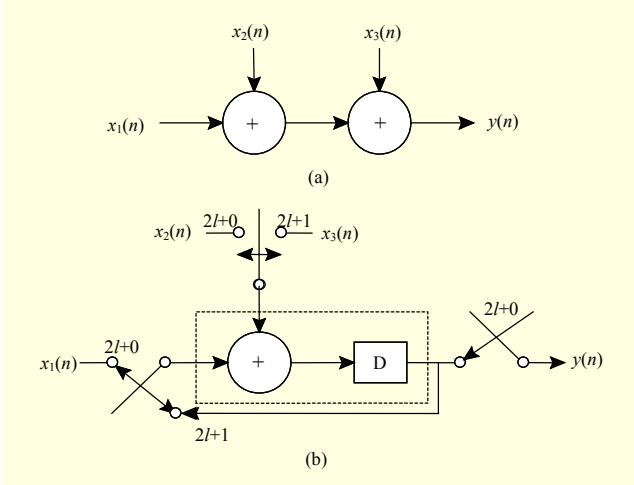

Fig. 3. (a) DSP program with two addition operations and (b) folded architecture in which two additions are folded into single hardware adder with one stage of pipelining.

processor. The folding algorithm is applied to reduce the silicon area without losing the characteristic properties of the filter.

III. Proposed Folding Architecture for Second-Order GTF

Folding is a technique to reduce the silicon area by time multiplexing many algorithm operations into single functional units, such as adders and multipliers. The folding transformation [24] provides a systematic technique to design control circuits for hardware in which several algorithm operations are time-multiplexed on a single functional unit. It is applied to a second-order GTF to reduce the silicon area where five multipliers and three adders are reduced to one each. Folding a circuit requires the following such steps: derivation of folding equations, minimizing the number of registers, data allocation, and synthesis. Here, an example is used to illustrate the folding transformation. A DSP program for adding two samples shown in Fig. 3(a) is gven by

$$y(n) = x_1(n) + x_2(n) + x_3(n). \tag{10}$$

The input should be valid for two clock cycles ($2l$+0 and $2l$+1, where l represents the iteration) so that one addition of two input samples is done in each clock cycle, and one output sample is therefore produced every two clock cycles. In the first clock cycle $2l$+0, $x_1(n)$+$x_2(n)$ is performed. In the second clock cycle $2l$+1, the third sample $x_3(n)$ is added to the intermediate result $x_1(n)$+$x_2(n)$, stored in the pipelining register. Hence, the output sample at the beginning of the next clock cycle $2l$+0 is $x_1(n)$+$x_2(n)$+$x_3(n)$. Figure 3(b) shows the folded architecture in which two additions are folded or time multiplexed to a single pipelined adder [24], [25].

1. Folding Equations

A folding set is an ordered set of K operations executed by the same functional unit. The operations are ordered from 0 to K–1. K is the number of operations folded into a single functional unit and is referred to as the folding factor. The folding order of a node is the block of time to which the node is scheduled to execute the operations in the hardware. The folding sets for the second-order GFT shown in Fig. 1 are given by

$$S_{\mathrm{A}} = \{\mathrm{A}_0, \mathrm{A}_1, \mathrm{A}_2, \varnothing_1, \varnothing_2\}, \tag{11}$$

$$S_{\mathrm{M}} = \{\mathrm{M}_0, \mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4\}. \tag{12}$$

This filter is folded with folding factor K=5, using the above folding sets. The folding factor K=5 means that the iteration period of the folded architecture is 5 units of time (u.t.), that is, each node of the GTF is executed exactly once every 5 u.t. in the folded architecture; it also means that a funtional unit in the folded hardware executes five operations. The folding set S_{A} contains only three addition operations and two null operations $(\varnothing_1, \varnothing_2)$ in positions 3 and 4 during which no operations are performed by the adder. The folding set S_{M} contains only multiplication operations, and the nodes in this folding set are executed by the same hardware multiplier. The folding equations, which are the basis for this technique, are derived for each edge of the GTF, using (13), as described in [24].

$$D_{\mathrm{F}}(U \rightarrow V) = Kw_{\mathrm{e}} - P_{\mathrm{U}} + v - u, \tag{13}$$

where K is the folding factor, w_{e} is the delay element of the edge between the source node U and the destination node V, P_{U} is the number of pipelined stages, and u and v represent the folding orders of nodes U and V, respectively. The folding equations for the second-order GTF are as listed below.

$$D_{\mathrm{F}}(\mathrm{IN} \rightarrow \mathrm{M}_0) = 5(0) - 0 + 0 - 0 = 0$$
$$D_{\mathrm{F}}(\mathrm{IN} \rightarrow \mathrm{M}_1) = 5(0) - 0 + 1 - 0 = 1$$
$$D_{\mathrm{F}}(\mathrm{IN} \rightarrow \mathrm{M}_2) = 5(0) - 0 + 2 - 0 = 2$$
$$D_{\mathrm{F}}(\mathrm{M}_0 \rightarrow \mathrm{A}_0) = 5(0) - 2 + 0 - 0 = -2$$
$$D_{\mathrm{F}}(\mathrm{M}_1 \rightarrow \mathrm{A}_1) = 5(0) - 2 + 1 - 1 = -2$$
$$D_{\mathrm{F}}(\mathrm{M}_2 \rightarrow \mathrm{A}_2) = 5(0) - 2 + 2 - 2 = -2$$
$$D_{\mathrm{F}}(\mathrm{M}_3 \rightarrow \mathrm{A}_1) = 5(0) - 2 + 1 - 3 = -4$$
$$D_{\mathrm{F}}(\mathrm{M}_4 \rightarrow \mathrm{A}_2) = 5(0) - 2 + 2 - 4 = -4$$
$$D_{\mathrm{F}}(\mathrm{A}_0 \rightarrow \mathrm{M}_3) = 5(0) - 1 + 3 - 0 = 2$$
$$D_{\mathrm{F}}(\mathrm{A}_0 \rightarrow \mathrm{M}_4) = 5(0) - 1 + 4 - 0 = 4$$
$$D_{\mathrm{F}}(\mathrm{A}_2 \rightarrow \mathrm{A}_1) = 5(1) - 1 + 1 - 2 = 3$$
$$D_{\mathrm{F}}(\mathrm{A}_1 \rightarrow \mathrm{A}_0) = 5(1) - 1 + 0 - 1 = 3$$

Retiming must be performed before folding to force causality in the system. Retiming is a transformation technique used to change the locations of the delay elements in a circuit

Table 1. Lifetime table for each node.

Node	$T_{input} \rightarrow T_{output}$
IN	0 → 2
M_0	2 → 4
A_0	1 → 6
M_1	3 → 5
A_1	2 → 7
M_2	4 → 6
A_2	3 → 8
M_3	5 → 5
M_4	6 → 6

without affecting the input and output characteristics of the circuit. The negative values of the above folding equations are made positive by using cutset retiming [24] of the delay elements. The equations are derived with an assumption that addition and multiplication operations require 1 and 2 units of time, respectively ($T_A = 1$ u.t., $T_M = 2$ u.t.).

2. Register Minimization

The folded structure contains a higher number of registers because the intermediate results need to be stored. The focus of this subsection is to minimize the number of registers used in the GTF architecture so that the silicon area occupied by the registers can be minimized. The procedure for computing the minimum number of registers and for allocating the data to these registers is as follows:

- A lifetime table is constructed using the folding equations;
- A lifetime chart is drawn to determine the number of registers.

The life period for each node is shown in Table 1, which is calculated using (14) and (15).

$$T_{input} = u + P_U. \quad (14)$$

$$T_{output} = T_{input} + \max\{D_F(U \rightarrow V)\}. \quad (15)$$

A linear lifetime chart for the GTF is shown in Fig. 4, which graphically represent the lifetime of each variable in a linear fashion. A data sample (variable) is said to be alive from the time it is produced until it is consumed. After the sample is consumed, it is dead. A variable occupies one register when it is alive. In lifetime analysis, the number of live variables at any time unit is determined, which gives the minimum number of registers required to implement the GTF architecture.

In the linear lifetime chart shown in Fig. 4, which is based on a chart presented in [23], each horizontal line represents a clock cycle and each vertical line represents the lifetime of a variable. From the lifetime chart, the minimum number of registers that can be used to implement the architecture of a second-order GTF is the maximum number of live variables at any time step. The minimum number of registers to implement the second-order GTF is max{0, 1, 2, 3, 5, 5, 5, 4, 4}= 5.

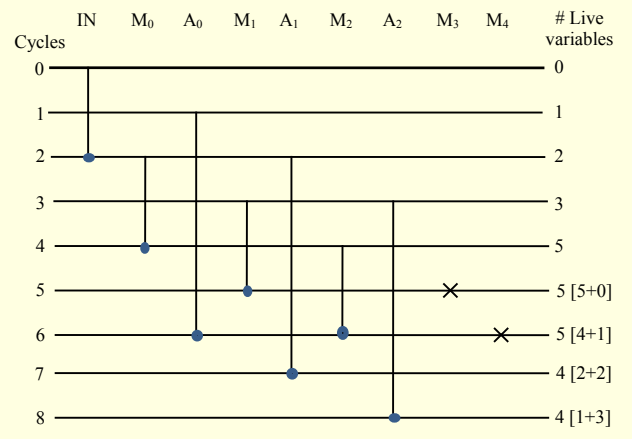

Fig. 4. Lifetime chart

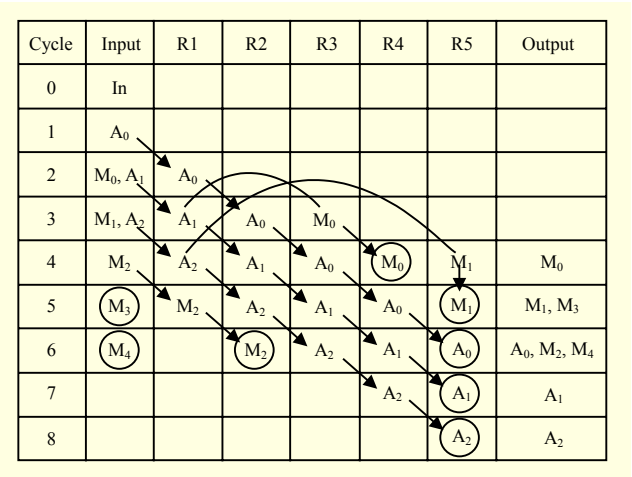

Fig. 5. Allocation table.

3. Data Allocation Using Forward-Backward Register Allocation

The registers are represented as R1, R2, R3, R4, and R5 in the allocation table (Fig. 5). In this scheme, the variables are allocated all the way forward and then to an appropriate backward register, thus the name "forward-backward." In this allocation technique, the registers are reused and dead variables are not stored.

4. Folded Architecture

The architecture corresponding to the folding equations in subsection III.1 and the register allocation in Fig. 5 give the

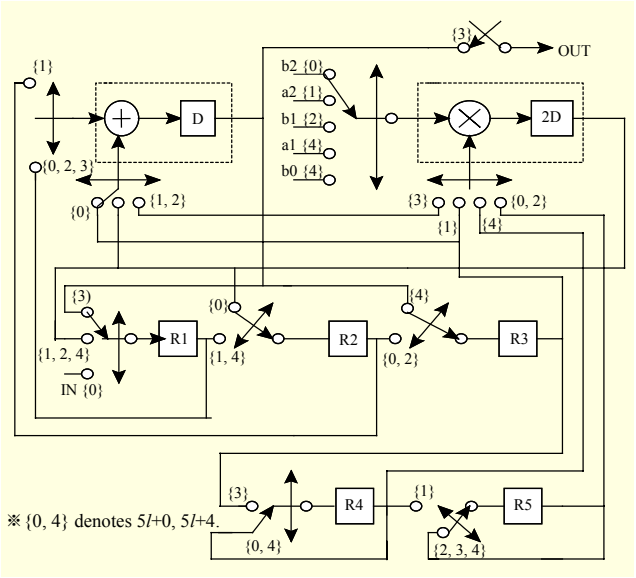

Fig. 6. Folded architecture for second-order GTF.

Table 2. Comparison between conventional and folded second-order GTFs.

	No. of functional units		No. of registers
	Adders	Multipliers	
Conventional	3	5	2
Folded	1	1	5

Table 3. Comparison between conventional and folded eighth-order GTFs.

	No. of functional units		No. of registers
	Adders	Multipliers	
Conventional	12	20	8
Folded	4	4	20

final folded architecture for the second-order GTF shown in Fig. 6. The switching instances for the architecture to perform multiplication and addition must be of the form $5l+m$ for $0 \le m \le 4$, where l denotes the iteration of the functional units.

Table 2 shows a comparison between the conventional and folded second-order GTFs in terms of the number of functional units and the number of registers. As shown, the folding transformation results in a lower number of functional units at the cost of a slight increase in the number of registers.

5. Folding Architecture for Eighth-Order GTF

Four sections of folded second-order filters are cascaded to realize the eighth-order digital gammatone filter. The magnitude and phase response of this filter is similar to the cascade structure of the conventional filter shown in Fig. 3. The folding sets for the eighth-order GTF shown in Fig. 2 are as listed below.

$$S_{A1} = \{A_0, A_1, A_2, \varnothing_1, \varnothing_2\}$$
$$S_{M1} = \{M_0, M_1, M_2, M_3, M_4\}$$
$$S_{A2} = \{A_3, A_4, A_5, \varnothing_3, \varnothing_4\}$$
$$S_{M2} = \{M_5, M_6, M_7, M_8, M_9\}$$
$$S_{A3} = \{A_6, A_7, A_8, \varnothing_5, \varnothing_6\}$$
$$S_{M3} = \{M_{10}, M_{11}, M_{12}, M_{13}, M_{14}\}$$
$$S_{A4} = \{A_9, A_{10}, A_{11}, \varnothing_7, \varnothing_8\}$$
$$S_{M4} = \{M_{15}, M_{16}, M_{17}, M_{18}, M_{19}\}$$

The folding factor is 5 (K=5) with time taken for addition equal to 1 unit of time (T_A = 1 u.t.) and for multiplication equal to 2 units of time (T_M = 2 u.t.). The folded second-order architecture shown in Fig. 6 is replicated four times and cascaded to obtain the folded architecture for the eighth-order GTF. Since it is replicated four times, using the same folding equations, the lifetime table, and the lifetime chart, the number of registers is found to be 20. Table 3 shows a comparison between the conventional and folded GTFs in terms of the number of functional units and the number of registers.

IV. Implementation of Proposed Work and Results

1. Implementation

The cochlea filter is implemented as a cascade of four sections of the folded structure of the second-order GTF. In this work, the designed GTF for a cochlea channel has a bandpass filter with a center frequency of 1,000 Hz and sampling frequency of 16 kHz. The speech signal bandwidth is approximately between 10 Hz and 3.5 kHz; hence, the filter is designed with a center frequency of 1,000 Hz. Moreover, the phase response of the GTF is unaltered for a narrow bandwidth of a speech signal around this center frequency [13], [16], [20], [22]. The coefficient of each of the second-order sections is determined using (7) and quantized to get an 8-bit coefficient. Since the folding phenomenon does not change the overall characteristic of the filter, the stability of the structure remains unaltered [25].

The frequency response of the cascaded filter shown in Fig. 7, which is obtained using MATLAB, mimics the nonlinear response of the biological cochlea. This filter is modeled in Verilog HDL and synthesized by using the Synopsys Design-Vision tool in 0.13-μm technology. To minimize the power consumed as a result of the huge adder circuitry, a ripple carry adder is used. Also, the application of

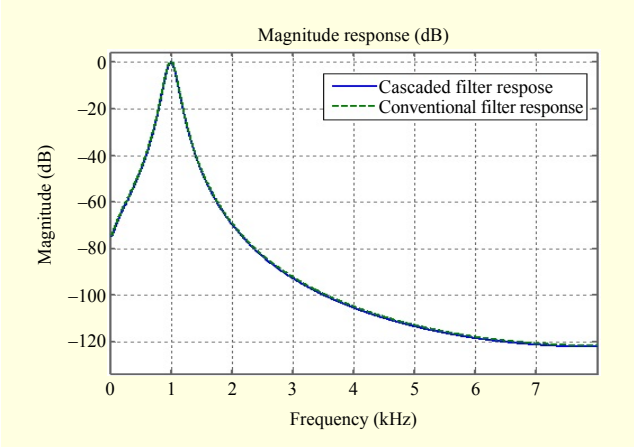

Fig. 7. Frequency response of cascaded and conventional GTF.

Table 4. Comparison of conventional and proposed structures.

	Conventional	Proposed
Area (μm^2)	10,464	4,301
Power (μW)	65.95	43.91
Delay (ns)	96.83	99.54
Power-delay product ($\times 10^{-15}$ J)	6,385.9385	4,370.8014

modified booth multipliers [26] with a regular partial product array results in an adequate improvement in the area, delay, and power consumption, compared with conventional GTFs.

2. Results

Tables 2 and 3 show comparisons between the conventional and folded GTFs in terms of the number of functional units and the number of registers. As shown, the folding transformation results in a lower number of functional units at the cost of an increase in the number of registers by 12. These registers are designed with minimal propagation delay [27], [28]. Table 4 shows a comparison between the conventional and proposed GTFs with respect to area, power, and delay. The delay represented in Table 4 indicates the total propagation delay needed for the input signal to become the output signal after being processed by the folded architecture of the filter. Simulation results show that the designed filter achieves a 58.89% reduction in area, a 33.4% reduction in power consumption, and a 31.55% reduction in power-delay product, compared to the conventional filter, but with a tradeoff of a 3% increase in the delay. The tradeoff is compensated by the reduction in area, power consumption, and power-delay product. The implemented digital filter can be fabricated to realize bionic ear processors of superior performance for application in speech recognition front-ends, portable health care devices, and implants for the hearing impaired.

V. Conclusion

In this work, a suitable auditory gammatone filter was realized in a digital environment using 0.13-μm technology and notable performance adders and multipliers. The presence of a GTF provides an appropriately shaped "pseudo-resonant" frequency transfer function, which has a very simple description in terms of its time-domain impulse response. This implemented digital filter can be used to realize the auditory filter bank that covers the entire audible band. Our results show that the folded architecture of GTF per cochlea channel significantly reduces the area of the filter, but there is a tradeoff with the delay. This increase in delay can be overcome in future work using retiming concepts. Additionally, the production cost of a speech processor can be reduced by reducing the silicon area of the filter.

References

[1] E. de Boer, "Synthetic Whole-Nerve Action Potentials for the Cat," *J. Acoust. Soc. Am.*, vol. 58, no. 5, 1975, pp. 1030-1045.

[2] R.F. Lyon and C. Mead, "An Analog Electronic Cochlea," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 36, no. 7, July 1988, pp. 1119-1134.

[3] S. Mandal, S.M. Zhak, and R. Sarpeshkar, "A Bio-Inspired Active Radio-Frequency Silicon Cochlea," *IEEE J. Solid-State Circuits*, vol. 44, no. 6, June 2009, pp. 1814-1828.

[4] C.D. Summerfield and R.F. Lyon, "ASIC Implementation of the Lyon Cochlea Model," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, Mar. 1992, pp. 673-676.

[5] S.C. Lim et al., "VHDL-Based Design of Biologically Inspired Pitch Detection System," *Proc. IEEE Int. Conf. Neural Netw.*, vol. 2, June 1997, pp. 922-927.

[6] M. Brucke et al., "Digital VLSI-Implementation of a Psychoacoustically and Physiologically Motivated Speech Preprocessor," *Proc. NATO Adv. Study Institute Comput. Hearing*, Il Ciocco, Italy, July 1998, pp. 157-162.

[7] A. Mishra and A.E. Hubbard, "A Cochlear Filter Implemented with a Field-Programmable Gate Array," *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 49, no. 1, Jan. 2002, pp. 54-60.

[8] M.P. Leong, C.T. Jin, and P.H.W. Leong, "An FPGA-Based Electronic Cochlea," *EURASIP J. Appl. Signal Process.*, no. 7, 2003, pp. 629-638.

[9] L. Van Immerseel and S. Peters, "Digital Implementation of Linear Gammatone Filters: Comparison of Design Methods,"

Acoust. Research Lett. Online, vol. 4, no. 3, Mar. 2003, pp. 59-64.

[10] R.V. Dundur et al., "Digital Filter for Cochlear Implant Implemented on a Field-Programmable Gate Array," *PWASET*, vol. 45, Sept. 2008, pp. 468-472.

[11] V.H. Kumar and P.S. Ramaiah, "Digital Speech Processing Design for FPGA Architecture for Auditory Prostheses," *J. Comput. Sci. Eng.*, vol. 6, no. 1, Mar. 2011, pp. 33-42.

[12] P.I.M. Johannesma, "The Pre-response Stimulus Ensemble of Neurons in the Cochlear Nucleus," *Proc. IPO Symp. Hearing Theory*, Eindhoven, The Netherlands, 1972, pp. 58-69.

[13] A.G. Katsiamis, E.M. Drakakis, and R.F. Lyon, "Practical Gammatone-Like Filters for Auditory Processing," *EURASIP J. Audio*, *Speech*, *Music Process*., vol. 2007, 2007, article no. 063685.

[14] R.F. Lyon, A.G. Katsiamis, and E.M. Drakakis, "History and Future of Auditory Filter Models," *Proc. IEEE Int. Symp. Circuits Syst.*, Paris, France, 2010, pp. 3809-3812.

[15] A.G. Katsiamis, E.M. Drakakis, and R.F. Lyon, "Introducing the Differentiated All-Pole and One-Zero Gammatone Filter Responses and Their Analogue VLSI Log-Domain Implementation," *Proc. 49th MWSCAS*, San Juan, Puerto Rico, Aug. 2006, pp. 561-565.

[16] W. Ngamkham et al., "Analog Complex Gammatone Filter for Cochlear Implant Channels," *Proc. IEEE Int. Symp. Circuits Syst.*, Paris, France, 2010, pp. 969-972.

[17] A.M.H.J. Aertsen and P.I.M. Johannesma, "Spectrotemporal Receptive Fields of Auditory Neurons in the Grassfrog — I: Characterization of Tonal and Natural Stimuli," *Bio. Cybern*., vol. 38, no. 4, 1980, pp. 223-234.

[18] M. Slaney, "An Efficient Implementation of the Patterson-Holdsworth Auditory Filter Bank," Apple Computer Corp., Technical Report #35, 1993.

[19] P. Mahalakshmi and M.R. Reddy, "Signal Analysis by Using FIR Filter Banks in Cochlear Implant Prostheses," *Proc. Int. Conf. Syst. Med. Biology*, IIT Kharagpur, India, Dec. 2010, pp. 253-258.

[20] K. Rajalakshmi and A. Kandaswamy, "VLSI Architecture of Digital Auditory Filter for Speech Processor of Cochlear Implant," *Int. J. Comput. Appl*., vol. 39, no. 7, Feb. 2012, pp. 19-22.

[21] T. Irino and M. Unoki, "An Analysis/Synthesis Auditory Filterbank Based on an IIR Gammachirp Filter," *Computational Models of Auditory Function*, S. Greenberg and M. Slaney, Eds. IOS Press, 2001, pp. 317-332.

[22] V. Hohmann, "Frequency Analysis and Synthesis Using a Gammatone Filterbank," *Acta Acustica United Acustica*, vol. 88, no. 3, 2002, pp. 433-442.

[23] M. Unoki and M. Akagi, "A Method for Signal Extraction from Noise-Added Signals," *Electron. Commun. Japan (Part 3)*, vol. 80, no. 11, Nov. 1997, pp. 1-11.

[24] K.K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*, Wiley India Pvt. Ltd., 2007.

[25] M. Ayinala, M. Brown, and K.K. Parhi, "Pipelined Parallel FFT Architectures via Folding Transformation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, June 2012, pp. 1068-1081.

[26] S.-R. Kuang, J.-P. Wang, and C.-Y. Guo, "Modified Booth Multipliers with a Regular Partial Product Array," *IEEE Trans. Circuits Syst. II*, *Exp. Briefs*, vol. 56, no. 5, May 2009, pp. 404-408.

[27] A.E. Willner et al., "Optical Signal Processing Using Tunable Delay Elements Based on Slow Light," *IEEE J. Sel. Topics Quantum Electron*., vol. 14, no. 3, 2008, pp. 691-705.

[28] K. Rajalakshmi, S. Gondi, and A. Kandaswamy, "A Fractional Delay FIR Filter Based on Lagrange Interpolation of Farrow Structure," *Int. J. Elect. Electron. Eng*., vol. 1, no. 4, 2012, pp. 103-107.

Rajalakshmi Karuppuswamy is an assistant professor for the senior grade in the Department of Electronics and Communication Engineering at PSG College of Technology, Coimbatore, India. Her teaching experience spans 16 years. She received her MEng in applied electronics from PSG College of Technology in 2001 and her BEng from the Electrical and Electronics Engineering Department of the Coimbatore Institute of Technology, Coimbatore, India, in 1994. She is pursuing her PhD at Anna University, Chennai, India. Her research interests include VLSI digital signal processing and nanoelectronics.

Kandaswamy Arumugam is a professor in and the head of the Department of Biomedical Engineering at PSG College of Technology, Coimbatore, India. He received his BEng in electrical engineering with honors in 1969 from PSG College of Technology. He obtained his MEng from Madras University, Chennai, India, in 1974 and his PhD from Bharathiar University, Coimbatore, India, in 1992. He was the dean of Electrical Sciences during 2005 and a professor and the head of the ECE Department from 1986 to 2002. He has 38 years of teaching experience. He has authored more than 80 papers that have been published in international journals, national journals, or conference proceedings. His fields of interest are signal processing, image processing, applications in medicine, and wireless communication systems. He is a member of the Institute of Electrical and Electronics Engineers (USA), a fellow of the Institution of Engineers (India), and a life member of the Biomedical Engineer Society of India, the Indian Society of Technical Education, the Computer Society of India, and the System Society of India.

Swathi Priya M. received her BEng in electronics and communication engineering and MEng in VLSI design from Anna University, Chennai, India, in 2010 and 2012 respectively. She was an undergraduate and post-graduate student in the Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, India. She is currently working as a design engineer in the FPGA-based product development industry. Her research interests are VLSI digital signal processing and biomedical applications.