FPGA-Based Implementation of Comb Filters Using Sequential Multiply-Accumulate Operations for Use in Binaural Hearing Aids

Shankarayya G. Kambalimath
Dept. of Electronics & Comm. Engg.
Basaveshwar Engg. College, Bagalkot, Karnataka, India
<kambalimathsg@rediffmail.com>

Prem C. Pandey
Dept. of Electrical Engg.
IIT Bombay, Mumbai, India
<pcpandey@ee.iitb.ac.in>

Pandurangarao N. Kulkarni
Dept. of Electronics & Comm. Engg.
Basaveshwar Engg. College, Bagalkot, Karnataka, India
<pnk_bewoor@yahoo.com>

Shivaling S. Mahant-Shetti
Karnataka Microelectronics Design Center
Manipal, Karnataka, India
<mahant@karmic.co.in>

Sangamesh G. Hiremath
G.M. Institute of Technology
Davangere, Karnataka, India
<dr_sgh@yahoo.co.in>

Abstract— Sensorineural hearing loss, caused by damage to the cochlea in the inner ear or to the auditory nerve, is associated with reduced dynamic range and increased spectral and temporal masking and results in degraded speech perception. For listeners using binaural hearing aids, spectral splitting of the speech signal can be used to reduce the effects of increased intraspeech spectral masking thereby improving speech perception. It has been reported that a pair of linear-phase comb filters, with auditory critical bandwidth based complementary magnitude responses selected for perceptual balance of loudness, resulted in a significant improvement in speech perception without affecting localization of broadband sound sources. The paper presents an investigation for FPGA-based implementation of these comb filters for use in binaural hearing aids. It is shown that a processing architecture employing sequential multiply-accumulate operations results in an efficient implementation.

Keywords— binaural hearing aid; comb filters; FPGA based implementation; sensorineural hearing loss

I. Introduction

Hearing losses are classified on the basis of location of the defect in the auditory system as conductive, sensorineural, and central losses [1]. The problems associated with either the outer or middle ear, which prevent sound from getting to the inner ear cause conductive loss. Sensorineural loss is mainly caused by the loss of hair cells in the cochlea and/or due to degeneration of the auditory nerve. Central loss is associated with the inability of the brain in decoding the neural firings into meaningful linguistic information. Elevated hearing thresholds, decreased dynamic range and loudness recruitment (abnormal loudness growth), and increased spectral and temporal masking are the characteristics of sensorineural loss and result in degraded speech perception.

Hearing aids generally provide frequency-selective amplification to compensate for the elevated hearing thresholds. Automatic volume control and dynamic range compression (with settable compression ratios, attack time, and release time) can be used to overcome the problems associated with the loudness recruitment and reduced dynamic range. Increased temporal masking adversely affects the detection of acoustic events. Widening of auditory filters results in increased spectral masking leading to poor discrimination of spectral contrasts. Increased masking makes speech perception very difficult in the presence of noise. It also results in poor speech perception due to increased intraspeech masking. The adverse effects of increased temporal masking can be addressed by enhancing the consonant-to-vowel ratio (CVR) [2], [3]. Schemes based on spectral contrast enhancement [4], [5] and multiband frequency compression [6], [7] have been used for reducing the effects of increased spectral masking. These schemes introduce processing related artifacts and have high computational requirements. Earlier studies have shown that, for persons having moderate bilateral sensorineural loss, who can use binaural aids, the effects of increased intraspeech masking can be reduced by using comb filters with complementary magnitude responses [8] – [10].

In the auditory system, the masking primarily takes place at the peripheral level, and integration of binaural information takes place at higher levels. In binaural presentation using complementary comb filters, the spectral components likely to mask or get masked by each other are presented to different ears for reducing the adverse effects of increased intraspeech spectral masking thereby improving the speech perception. Kulkarni et al. [10] reported that comb filters based on

978-1-4799-5364-6/14/$31.00 ©2014 IEEE

auditory critical bandwidths [11], with magnitude responses designed for perceptual balance of loudness and linear phase responses, resulted in a significant improvement in speech perception. Modified rhyme test (MRT) [12], [13] conducted on eleven subjects with moderate bilateral sensorineural loss showed 14 – 31% improvement in the consonant recognition scores and a decrease of 0.26 s in mean response time, without adversely affecting localization of source direction for broadband environmental sound and speech.

Audio signal processing techniques can be implemented using ASICs (application specific integrated circuits), FPGAs (field programmable gate arrays), or DSP (digital signal processing) chips. Because of power and space constraints, hearing aids are generally designed using ASICs, which involves significant nonrecurring cost related to chip fabrication. This cost can be reduced by using FPGA for fast prototyping and then taking the design to ASIC stage. The FPGA structure facilitates parallel and compound data processing operations every clock cycle, whereas the DSP-chip based processing involves sequential instruction fetch-and execute cycles [14]. Several investigations [14] – [17] have reported FPGAs to be better suited than DSP chips for applications involving highly parallel computations. After FPGA-based prototyping and testing, RTL-to-GDS, an automated ASIC digital design flow, can be used for transforming synthesized VHDL or Verliog code into GDSII mask layout file [18] – [20].

Figure 1 shows block diagram of a binaural hearing aid using comb filters for dichotic presentation to reduce the adverse effects of increased intraspeech spectral masking. The filters used in [10] were realized using Matlab-based offline processing with floating-point operations. They were designed as 513-coefficeint linear-phase FIR filters for the sampling frequency of 10 kHz, with the magnitude responses meeting the requirements of narrow transition widths, low pass-band ripple, large stop-band attenuation, and very small deviation in the sum of the magnitude responses on a linear scale. The filters were designed by iterative application of frequency sampling method of filter design [21], [22]. With the same set of filter coefficients scaled to 15-bit sign magnitude integers, an investigation for an efficient FPGA-based implementation, using different processing architectures, of these comb filters for use in binaural hearing aids is presented.

II. Implementation

The comb filters are implemented using “Altera DE2-70” board, consisting of FPGA “Altera Cyclone II EP2C70F896C6” and audio stereo codec “Wolfson WM8731”, as shown in Figure 2. Implementation of the comb filters using different architectures is investigated to compare the filter characteristics and the resource requirements in order to evaluate the feasibility for use in binuaral hearing aids.

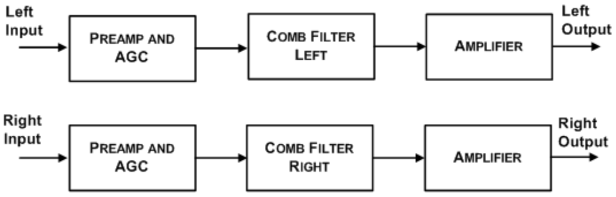

Fig. 1. A binaural hearing aid using comb filters.

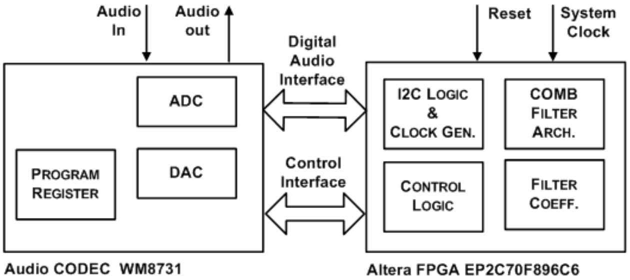

Fig. 2. FPGA and Audio codec interfacing on FPGA board used for comb filter implementation.

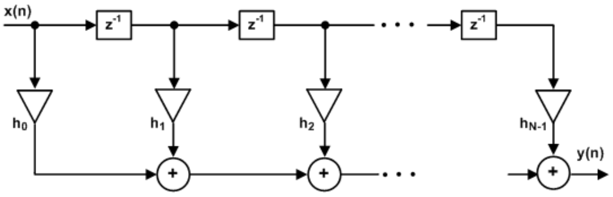

Fig. 3. Comb filter realization as direct-form FIR filter structure.

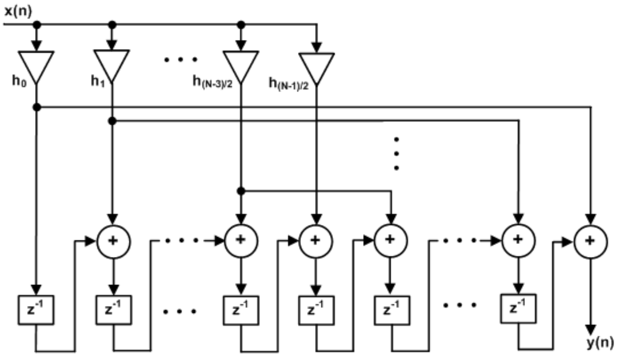

Fig. 4. Comb filter realization as transposed-form linear phase FIR filter structure.

Digital filters are realized using delays, coefficient multipliers, and adders connected as specific filter structures. Different filter structures offer different trade-offs between the number of processing blocks, tolerance to coefficient quantization errors, and dynamic range requirements for intermediate values. Figure 3 shows direct-form FIR filter realization, with input $x(n)$,filter coefficients h_m ($h_0,h_1,.., h_{N\text{-}1}$), and output $y(n)$ given as

$$y(n) = h_0x(n) + h_1x(n-1) + h_2x(n-2) + ... + h_{N\text{-}2}x(n-N-2) + h_{N-1}x(n-N-1)$$

A transposed-form structure as shown in Figure 4 (for odd N) can be used for linear-phase FIR filters having symmetric impulse response, with the output as the following:

$$y(n) = h_0[x(n) + x(n-(N-1))] + h_1[x(n-1) + x(n-(N-2))] + ...$$
$$+ h_{(N-3)/2}[x(n-(N-3)/2 + x(n-(N+1)/2)] + h({}_{N\text{-}1)/2}x(n-(N-1)/2)$$

Exploiting the symmetry in coefficients, it reduces the number of multipliers to half as compared to that in the direct-form realization.

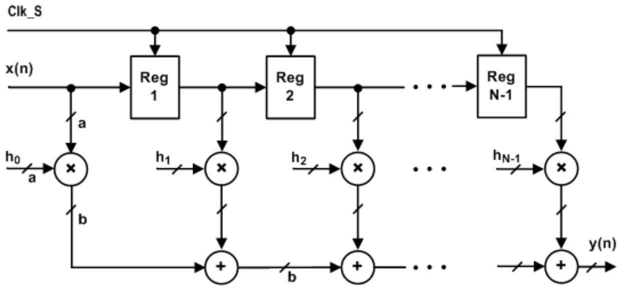

Fig. 5. FPGA-based implementation of direct-form FIR filter structure using parallel multiply-accumulate operations.

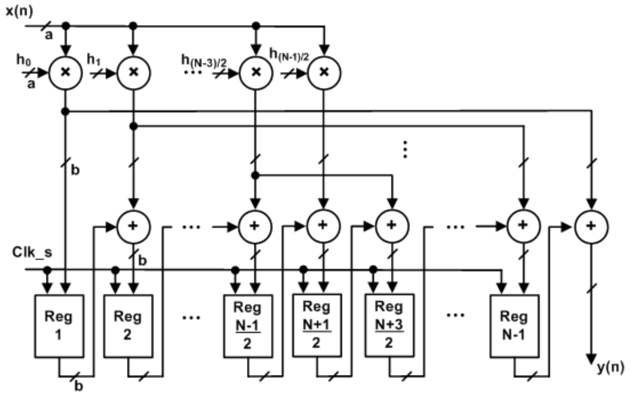

Fig. 6. FPGA-based implementation of transposed-form linear phase filter structure using parallel multiply-accumulate operation.

FPGA-based implementation of the filter involves a representation of the processing architecture using registers for delay operations and multiplier and adder blocks for multiply-accumulate operations. Coefficients are stored as constant using logic cells. Assuming no resource constraint in the form of a limit on the available number of multipliers and adders, FPGA-based implementation of an FIR filter can be carried out using parallel sets of multiply-accumulate operations. This type of implementation for the direct-form realization is shown in Figure 5. It uses $N-1$ registers for delays, N coefficient inputs (one for each coefficient or tap weight h_m, realized using gates), N multipliers, and $N-1$ adders. The tap weights and signal samples are a-bit integers. At the edge of each sampling clock, with frequency equal to the sampling frequency f_s, the content of each of the delay register gets transferred to the next one. Each multiplier output is available as b-bit integer and the multiplier outputs are added together using two-input adders. In this implementation, the multipliers and adders perform N multiplication and $N-1$ addition operations in parallel. It results in an implementation of the transposed-form linear phase FIR realization, using parallel multiply-accumulate operations is shown in Figure 6. The number of multipliers in this implementation is half of that used in the previous one.

We need to conserve hardware resources involved in the implementation of the comb filters, because a hearing aid needs to have several other processing blocks, particularly the frequency-dependent gain and dynamic range compression. A parallel processing based implementation of the comb filters using ALTERA FIR design compiler 9.1 along with standard libraries has been reported in [23]. As that implementation has been found to be very resource-intensive, an RTL code based implementation of the processing architectures in Figure 5 and Figure 6 is carried out for reducing the resource requirement.

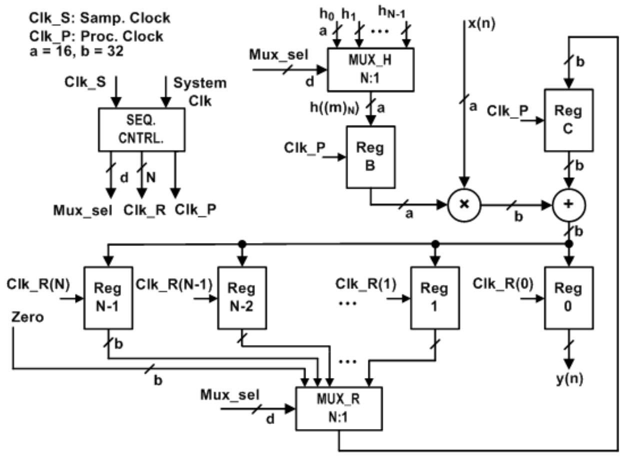

Fig. 7. FPGA-based implementation of direct-form FIR filter structure using sequential multiply-accumulate operations.

The filter implementation using parallel multiply-accumulate operations in the processing architecture of Figure 5 and Figure 6 is relatively straightforward. However, it does not use the processing blocks optimally. The delay, multiplication, and adding operations are carried out once every sampling clock and the processing blocks remain idle for most of the duration of the sampling interval ($1/f_s$). We can use a single multiplier-adder block iteratively with a processing clock CLK_P, with frequency $f_P > N\ f_s$, for carrying out all the multiply-accumulate operations in a sequential fashion by using a multiplexer for data routing. This implementation, based on “sequential multiply-accumulate operations” is shown in Figure 7. It saves on the resources needed for multipliers and adders, but needs extra resources for implementing the multiplexers, control logic, and clock generator, as shown in the figure.

The processing architecture as shown in Figure 7 consists of a multiplier, an adder, N registers (Reg-I, $0 < I < N-1$) for intermediate results, 2 registers (Reg-B, Reg-C) as buffers, a multiplexer (MUX-H) for tap weights, and a multiplexer (MUX-R) for intermediate results. It also has a sequence controller (SEQ. CNTRL.) which controls the operation sequence by generating the processing cycle clock CLK_P and register load clocks CLK_R(I) ($0 \leq I \leq N-1$) and multiplexer controls (MUX_sel) from the sampling clock CLK_S and the System Clock as inputs. Within each sampling interval, N processing clocks are generated, with each of the register load clocks generated in successive processing clock cycles. Active edges of CLK_R are out of phase with that of CLK_P. The MUX_sel is changed sequentially in phase with the processing clock. In each processing cycle, the input sample $x(n)$ is

TABLE I. REGISTER OUTPUTS IN SEQUENTIAL ARCHITECTURE OF FIG. 7

Proc. Cycle j	$R_0(n,j)$	$R_1(n,j)$	$R_2(n,j)$	...	$R_{N\text{-}2}(n,j)$	$R_{N\text{-}1}(n,j)$
1	$h_0x(n)+R_1(n-1,N)$	$R_1(n-1,N)$	$R_2(n-1,N)$	...	$R_{N\text{-}2}(n-1,N)$	$R_{N\text{-}1}(n-1,N)$
2	$R_0(n,1)$	$h_1x(n)+R_2(n-1,N)$	$R_2(n-1,N)$	...	$R_{N\text{-}2}(n-1,N)$	$R_{N\text{-}1}(n-1,N)$
3	$R_0(n,1)$	$R_1(n,2)$	$h_2x(n)+R_3(n-1,N)$	...	$R_{N\text{-}2}(n-1,N)$	$R_{N\text{-}1}(n-1,N)$
...	...	...	...	...	...	...
...	...	...	...	...	...	...
$N-1$	$R_0(n,1)$	$R_1(n,2)$	$R_2(n,3)$	...	$h_{N-2}x(n)+R_{N\text{-}1}(n-1,N)$	$R_{N\text{-}1}(n-1,N)$
N	$R_0(n,1)$	$R_1(n,2)$	$R_2(n,3)$	...	$R_{N\text{-}2}(n,N\text{-}1)$	$h_{N\text{-}1}x(n)+0$

TABLE II. RESOURCE REQUIREMENTS OF DIFFERENT FILTER ARCHITECTURES.

Filter Architecture	Resources used for 257-coefficient filter				Resources used for 513-coefficient filter			
	Total logic elements	Total combina-tional functions	Dedicated logic registers	Embedded 9-bit multipliers	Total logic elements	Total combinational functions	Dedicated logic registers	Embedded 9-bit multipliers
Earlier work [23]					53%	47%	34%	----
Direct-form, Parallel multiply-accumulate	18%	17%	12%	----	32%	29%	24%	----
Transposed-form linear phase, Parallel multiply-accumulate	18%	17%	12%	----	32%	29%	24%	----
Direct-form, Sequential multiply-accumulate	15%	9%	12%	<1%	30%	18%	24%	1%

multiplied with the selected coefficient and added with the correspondingly selected register content. The operation sequence with respect to the processing cycle j and the sample number n is summarized in Table I. After the first processing cycle of a sampling interval, the R0 output is taken as the output $y(n)$, given as the following:

$$
\begin{aligned}
y(n) &= R_0(n,N)\\
&= h_0x(n) + R_1(n-1,N)\\
&= h_0x(n) + h_1x(n-1) + R_2(n-2,N)\\
&= h_0x(n) + h_1x(n-1) + h_2x(n-2) + R_3(n-3,N)\\
&\quad\ldots\\
&\quad\ldots\\
&= h_0x(n) + h_1x(n-1) + h_2x(n-2) + \ldots + R_{N\text{-}2}(n-N-2)\\
&= h_0x(n)+h_1x(n-1)+h_2x(n-2) + \ldots + h_{\mathrm{N}\text{-}2}x(n-N-2) + R_{\mathrm{N}\text{-}1}(n-N-1)\\
&= h_0x(n) + h_1x(n-1) + h_2x(n-2) + \ldots + h_{\mathrm{N}\text{-}2}x(n-N-2) + h_{\mathrm{N}\text{-}1}x(n-N-1)
\end{aligned}
$$

and thus the implementation serves as N-tap FIR filter.

FPGA-based implementation of the comb filters using RTL code was carried out for the three processing architectures, as described above: (a) direct-form filter structure with parallel multiply-accumulate operations as shown in Figure 5 (b) transposed-form linear phase filter structure with parallel multiply-accumulate operations as shown in Figure 6, and (c) direct-form filter structure with sequential multiply-accumulate operations as shown in Figure 7. For compatibility with earlier investigations and considering the end-use in a hearing aid, a sampling frequency of 10 kHz along with 16-bit quantization was used. The I2C logic for interfacing to the audio codec was implemented as described in [23]. Implementations were carried out for comb filters designed with 257 and 513 tap weights (i.e. N = 257 and 513) and with the floating point coefficients scaled to 15-bit signed integer coefficients and 32-bit adder and register (i.e. a = 16, b = 32). In the implementation using sequential operations the processing clock CLK-P is set as 6.125 MHz and as 3.0625 MHz for N = 513 and 257, respectively.

III. RESULTS

All the implementations worked satisfactorily for sampling frequency of 10 kHz. Binaural presentation of the processed test stimuli through headphones did not show any perceptual distortion for the processed sounds indicating nearly perfect perceptual fusion of the binaural sounds. Magnitude responses of the comb filters implemented using the three architectures as described in the previous section were obtained. The responses of the comb filters with 257 and 513 coefficients are given in Figures 8 and 9, respectively. The comb filter with MATLAB based implementation reported in [10] for 513 coefficients have low pass-band ripple (<1 dB), large stop-band attenuation (>30 dB), complementary magnitude responses with narrow transition widths (<55 Hz), and cross-over gains of −5 to −6 dB. All the filter responses shown in Figure 8 and Figure 9 have pass-band ripple below 2 dB and cross-over gains of −4 to −8 dB. Stop-band attenuation is greater than 25 dB for 513 coefficient filters and greater than 18 dB for 257-coefficient filter. Deviation in the sum of magnitude responses on a linear scale is below 0.098 for all responses. Thus the properties of the 513-coefficient FPGA-

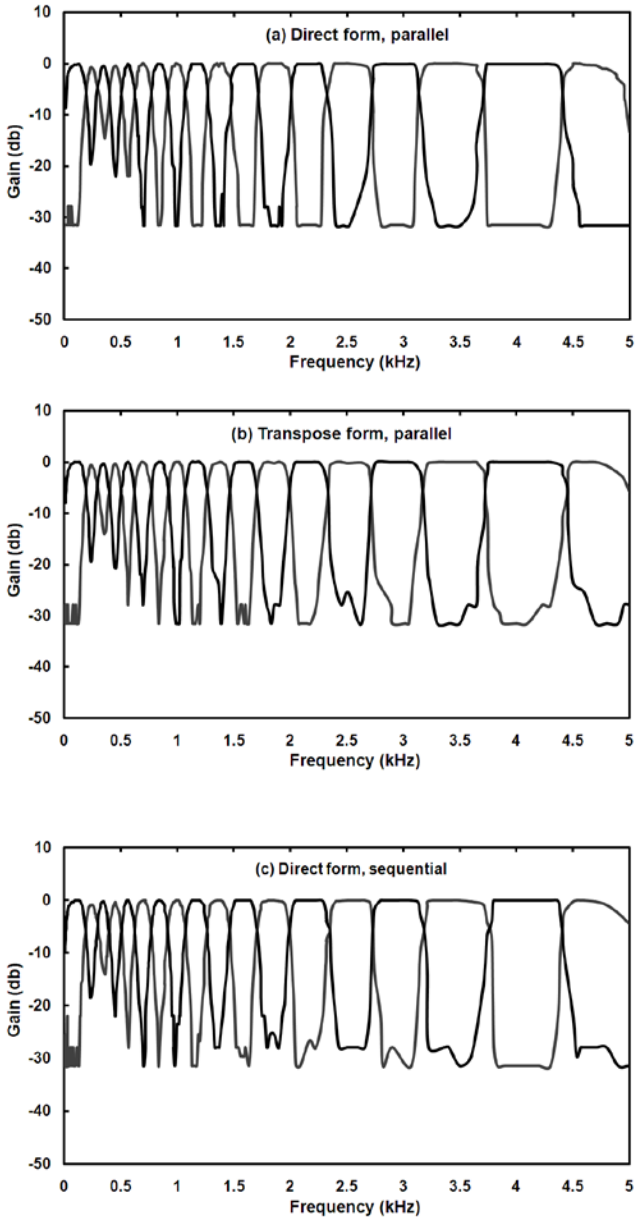

Fig. 8 Magnitude responses of the 257-coefficient comb filters (dark & light for L & R filters) using the three processing architectures: (a) Direct-form parallel, (b) Transpose-form parallel, (c) Direct-form sequential.

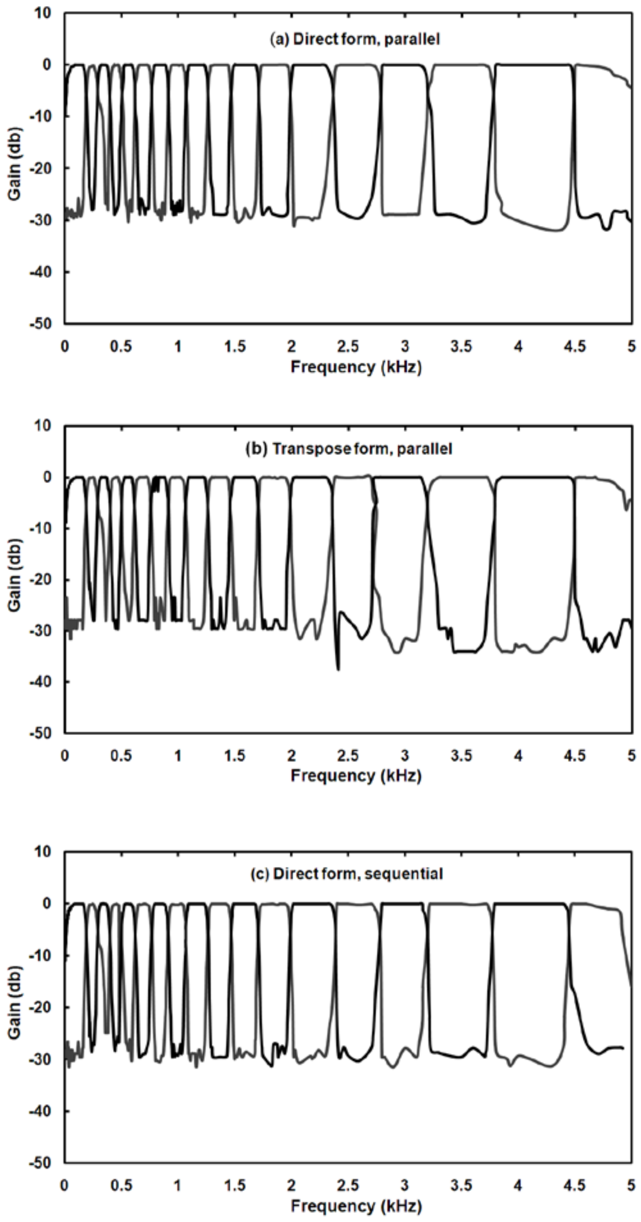

Fig. 9 Magnitude responses of the 513-coefficient comb filters (dark & light for L & R filters) using the three processing architectures: (a) Direct-form parallel, (b) Transpose-form parallel, (c) Direct-form sequential.

based comb filters closely match with those of the filters in [10].

A comparison of resource requirements of the three filter implementations using RTL code presented here and the one in [23] is given in Table II. It is seen that use of the RTL code based implementation has significantly reduced the resource requirement. Use of transposed-form linear phase architecture did not result in any saving over the direct-form architecture. Use of sequential multiply-accumulate operations has significantly reduced the resource requirement. However, the saving is not in proportion to the number of coefficients, which is expected because of additional resources needed for data routing. For 513-coefficient comb filter, the implementation takes a fraction of the resources available on FPGA EP2C70F896C6 (30% of total logic elements, 18% of total combinational functions and 24% of dedicated logic registers, 1% embedded multipliers), with a scope for implementation of the other processing blocks of the hearing aid.

IV. CONCLUSIONS

An earlier investigation [10] has shown that for listeners with bilateral sensorineural hearing loss, pair of comb filters based on auditory critical bandwidths, with magnitude responses designed for linear phase responses and perceptual balance of loudness may be useful in improving speech perception. These comb filters with 513 and 257 coefficients were implemented using three different architectures on an FPGA and tested. Implementation using a 16-bit codec, 15-bit signed coefficients, and 32-bit registers resulted in satisfactory filter responses and implementations used only a fraction of resources available on the chip. The filter architecture using sequential multiply-accumulate operations was found to be more efficient in resource utilization, and has shown the scope for implementing other processing blocks of a hearing aid on the same chip.

Implementation of a prototype hearing aid with dynamic range compression, frequency-selective response, and comb filters can be taken up for developing a hearing aid. The frequency selective response can be integrated in the comb filter design itself, reducing the resource requirement and the delay in the signal processing path. After successful testing of the FPGA-based prototype on patients in audiology clinics, FPGA design can be converted to an ASIC for developing wearable hearing aid.

REFERENCES

[1] H. Dillon, Hearing Aids. NewYork: Thieme Medical, 2001.

[2] E. Kennedy, H. Levitt, A. C. Neuman, and M. Weiss, “Consonant-vowel intensity ratios for maximizing consonant recognition by hearing hearingimpaired listeners,” J. Acoust. Soc. Am. vol. 103, pp. 1098 – 1114, 1998.

[3] S. Gordon-Salant, “Recognition of natural and time/intensity altered CVs by young and elderly subjects with normal hearing,” J. Acoust. Soc. Am. vol. 80, pp. 1599 – 1607, 1986.

[4] T. Baer, B. C. J. Moore, and S. Gatehouse, “Spectral contrast enhancement of speech in noise for listeners with sensorineural hearing impairment: effects on intelligibility, quality, and response times,” J. Rehabil. Res. Dev. vol. 30, pp. 49 – 72, 1993.

[5] I. Cohen, “Speech spectral modeling and spectral enhancement based on autoregressive conditional heteroscedasticity models,” Signal Processing. vol. 86, pp. 698 – 709, 2006.

[6] T. Arai, K. Yasu, and N. Hodoshima, “Effective speech processing for various impaired listeners,” in Proc. 18th Int. Congress Acoust. (ICA ‘04), Kyoto, Japan, 2004, pp. 1389 – 1392.

[7] P. N. Kulkarni, P. C. Pandey, and D. S. Jangamashetti, “Multi-band frequency compression for improving speech perception by listeners with moderate sensorineural hearing loss,” Speech Commun., vol. 54, pp. 341–350, 2012.

[8] P. E. Lyregaard, “Frequency selectivity and speech intelligibility in noise,” Scand. Audiol. Suppl. vol. 15, pp. 113 – 122, 1982.

[9] T. Lunner, S. Arlinger, and J. Hellgren, “8-channel digital filter bank for hearing aid use: preliminary results in monaural, diotic, and dichotic modes,” Scand. Audiol. Suppl., vol. 38, pp. 75 – 81, 1993.

[10] P. N. Kulkarni, P. C. Pandey, and D. S. Jangamashetti, “Binaural dichotic presentation to reduce the effects of spectral masking in moderate bilateral sensorineural hearing loss,” Int. J. Audiology, vol. 51, pp. 334 – 344, 2012.

[11] E. Zwicker, “Subdivision of audible frequency range into critical bands (Frequenzgruppen),” J. Acoust. Soc. Am. vol. 33, pp. 248, 1961.

[12] A. S. House, C. E. Williams, M. H. L. Hecker, and K. D. Kryter, “Articulation testing methods: consonantal differentiation with closed response set,” J. Acoust. Soc. Am. vol. 37, pp. 158 – 166, 1965.

[13] ANSI. “Methods for measuring the intelligibility of speech over communication systems,” Revised Standards Report ANSI S3.2-1989, American National Standards Institute, New York, 1989.

[14] D. Halupka, A. S. Rabi, P. Aarabi, and A. Sheikholeslami, "Low-power dual-microphone speech enhancement using field programmable gate arrays," IEEE Trans. Signal Process., vol. 55, pp. 3526 – 3535, 2007.

[15] R. Woods, J. McAllister, G. Lightbody, and Y. Yi, FPGA-based Implementation of Signal Processing Systems. West Sussex, UK: John Wiley, 2008.

[16] N. Kehtarnavaz and S. Mahotra, Digital Signal Processing Laboratory: LabVIEW-based FPGA Implementation. Boca Raton, Florida: BrownWalker, 2010.

[17] N. Kehtarnavaz and S. Mahotra, "FPGA implementation made easy for applied digital signal processing courses," in Proc. IEEE ICASSP, 2011, pp. 2892 – 2895.

[18] T. L. Brandon, B. F. Cockburn, and D. G. Elliott, "HDL2GDS: a fully automated ASIC digital design flow," in Proc. 2005 Canadian Conf. Electrical Computer Engineering., pp. 1535 – 1538.

[19] J. Lu and B. Taskin, "From RTL to GDSII: An ASIC design course development using Synopsys® University Program," in Proc. 2011IEEE International Conference on Microelectronic Systems Education (MSE), pp. 72 – 75.

[20] D. Hill, and A. B. Kahng, "Guest editors' introduction: RTL to GDSII - from foilware to standard practice," IEEE Design & Test of Computers, vol. 21, no. 1, pp. 9 –12, Jan. – Feb. 2004.

[21] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, Discrete-time Signal Processing. Englewood Cliffs, New Jersey: Prentice-Hall, 1999.

[22] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications. New York: Macmillan, 1992.

[23] S. G. Kambalimath, P. C. Pandey, P. N. Kulkarni, S. S. Mahant-Shetti, and S. G. Hiremath, "FPGA based implementation of comb filters for use in binaural hearing aids for reducing intraspeech spectral masking," in Proc. IEEE INDICON, 2013, pp.1–5.