Multichannel Filter Banks and Their Implementation Using Computers with a Parallel Structure

D. I. Kaplun, D. M. Klionskiy, A. S. Voznesenskiy, and V. V. Gulvanskiy
Computer Science Department
Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, Russian Federation

Abstract— The paper is devoted to the development of a WOLA-algorithm (weighted overlap add algorithm) for processing vector (multichannel) signals. The algorithm is considered as a generalization of one-dimensional WOLA with certain modifications. WOLA is expounded as an algorithm for real-time signal processing and the main advantages of WOLA are provided. We also discuss software-hardware implementation of WOLA using CPU (Central Processing Unit) and CUDA (Compute Unified Device Architecture). Finally we demonstrate the possibility of reducing hardware costs for multichannel signal processing using FPGA (field programmable gate arrays) and distributed arithmetic.

1. INTRODUCTION

In the present paper we are going to consider multichannel filter banks and their implementation for wideband monitoring tasks. The term "monitoring" implies a systematic or continuous data acquisition from some object or system [1]. Monitoring is often performed in a wide frequency band (wideband monitoring) and includes tracking the main system parameters and deviation search in these parameters. Any deviation is usually caused by abnormal functioning of an object and requires immediate action in order to prevent failures.

Wideband monitoring is frequently used in the following areas [1–5]: *hydroacoustics* (monitoring of water areas (coastal and marine waters), underwater and surface objects, study of navigation canals and near-shore waters, analysis of oceanic seismicity, etc.); *radio monitoring* (time-frequency processing of radiations, noise suppression, signal classification, signal parameter estimation, signal demodulation, direction-finding, etc.); *vibration analysis* (vibration measurements, vibration parameter estimation in the time and frequency domains, resonance frequency estimation; analysis of vibrations is important for spacecraft, aircraft, engines, turbines, machinery, etc.); *geophysics* (monitoring of seismic and geomagnetic activity in various regions of our planet).

2. MULTICHANNEL DFT-MODULATED FILTER BANK AND WOLA-ALGORITHM FOR PROCESSING VECTOR SIGNALS

Wideband monitoring tasks are often handled using digital *FIR filters* [6–8] designed by the window or Chebyshev method [6–8]. Digital filters are also needed for *multichannel signal processing* [8]. One of the approaches to multichannel signal processing can be implemented by means of *digital filter banks* with the help of high-performance software-hardware tools.

The present paper is devoted to the development of a WOLA [8] modification for processing multichannel signals.

Digital filter banks and their implementations have been studied for many years and DFT-modulated filter banks (discrete Fourier transform modulated filter banks) [6–8] have long been applied to various tasks related to multirate signal processing and wideband monitoring. Among the most significant achievements are the development of the polyphase form of a filter bank and also the one-dimensional WOLA-algorithm [8].

DFT-modulated filter banks have several implementations. One of the simplest is a filter bank with complex modulation [8]. However, this implementation is not efficient since linear convolution in each channel is calculated for a large sampling rate (this leads to high computational costs). Therefore it is necessary to reduce the sampling rate, for instance, by choosing the polyphase structure [8].

One of the ways of implementing a polyphase filter bank is the WOLA algorithm, which performs *block-by-block* analysis [8]. Like a filter bank with complex modulation, the output signal is determined as

$$X_k(m) = \sum_{n=-\infty}^{\infty} h\left(mM - n\right) x(n) W_K^{-kn}, \tag{1}$$

where k is the number of a filter bank channel, m is the block number in WOLA, $h(n)$ is the impulse response of a LF-prototype (low-frequency prototype) filter, M is the decimating factor of the input signal,

$$W_K^{-kn} = e^{-j[(2\pi k/K)n]}, \tag{2}$$

where $j = \sqrt{-1}$.

According to (1) it is possible to introduce a filter with the impulse response $h(mM - n)$ as a moving analysis window for extracting the sequence $y_m(n)$, which is then subjected to short-time Fourier transform (STFT). This is the key idea behind the algorithm. In the case of critical decimation, when $M = K$, WOLA is identical to a polyphase filter bank. The main difference is that WOLA is oriented to block-by-block analysis (rather than parallel processing) and the signal in question has to be split into sections of the length N_h and overlap equal to M samples.

The main factor of the filter bank implementation that affects computational complexity is the LF-prototype filter. The magnitude response of the filter coincides with the desired one for a single channel. LF-prototype order depends on a required channel bandwidth, rectangularity shape factor, and gain flatness in the passband of one single channel.

Thus, we have to create efficient FIR-filters using the minimal computational complexity criterion for reducing hardware and software costs.

The algorithm outlined above is intended for processing *one-dimensional signals* (the term "one-dimensional signal" is used as a synonym for "single-channel signal") presented in the form of a vector. In practice the original signal can consist of several one-dimensional signals (multichannel signal).

The WOLA-algorithm for processing vector signals can be considered as a generalization of the corresponding WOLA-algorithm for processing one-dimensional signals. However, the suggested modification will make it possible to perform efficient processing of a vector signal and then to perform efficient hardware-software implementation using high-performance software and hardware tools.

After signal computation at the output of each channel sub-band processing is carried out (the signal in each channel corresponds to some frequency band) including spectral analysis, time-frequency analysis, statistical analysis in the time domain, demodulation, etc..

Consider an input vector (multichannel) signal

$$x(n) = [x_0(n) \ \ldots \ x_i(n) \ \ldots \ x_{S-1}(n)]^T$$

as a set of one-dimensional (single-channel) signals

$$x_i(n), \quad i = 0, \ldots, S-1; \quad n = 0, \ldots, N-1.$$

The original vector signal $x(n)$ can be represented by a rectangular matrix of the dimension $S \times N$, where N is the signal length of $x_i(n)$; S is the number of one-dimensional signals.

In the following we are going to use the following notations:

1) $\mathbf{x}(n) = [\mathbf{x}_0(n), \mathbf{x}_1(n), \ldots, \mathbf{x}_{S-1}(n)]$ is a multichannel signal in the matrix form, where the sub-matrices $\mathbf{x}_0(n)$, $\mathbf{x}_1(n)$, $\ldots$, $\mathbf{x}_{S-1}(n)$ can be represented as

$$\mathbf{x}_0 = \begin{bmatrix} x_0(0) \\ x_0(1) \\ \vdots \\ x_0(N-1) \end{bmatrix}, \ \mathbf{x}_1 = \begin{bmatrix} x_1(0) \\ x_1(1) \\ \vdots \\ x_1(N-1) \end{bmatrix}, \ \ldots \ \mathbf{x}_{S-1} = \begin{bmatrix} x_{S-1}(0) \\ x_{S-1}(1) \\ \vdots \\ x_{S-1}(N-1) \end{bmatrix}. \tag{3}$$

2) S is the size of a multichannel signal (number of one-dimensional signals).

3) M is the decimation factor, $M = 1, \ldots, K$, where K is the number of filter bank channels for each polyphase implementation corresponding to $x_i(n)$.

4) $g_i(m)$ and $e_i(m)$ is the impulse response of the i-th polyphase filter in the direct and inverse sections, respectively.

In the case of critical decimation when $M = K$ computation of output signals of a polyphase filter bank is equivalent to WOLA application, but the WOLA algorithm is oriented to *block-by-block analysis*.

The input of the WOLA-algorithm is a vector signal $\mathbf{x}(n) = [\mathbf{x}_0(n), \mathbf{x}_1(n), \ldots \mathbf{x}_{S-1}(n)]$, where $n = 0, \ldots, N-1$. Furthermore, we need to know $h(n)$, which is the impulse response of a LF-prototype filter (vector of size $1 \times N_h$).

Next, weighting has to be applied:

$$\tilde{x}_{mi}(n) = h\,(mM - n)\,x_i(n),\ \ i = 0, \ldots, S-1;\ \ n = 0, \ldots, N-1, \tag{4}$$

where m is the block number (N_h is the block length). The total number of blocks P of the length N_h with the overlap $(N_h - M)$ is determined as

$$P = 1 + \left\lfloor \frac{N - N_h}{M} \right\rfloor, \tag{5}$$

where $\lfloor \cdot \rfloor$ denotes rounding towards minus infinity. As a result of (4) we arrive at the matrix of weighted samples:

$$\tilde{\mathbf{x}}(n) = [\tilde{\mathbf{x}}_0(n),\ \tilde{\mathbf{x}}_1(n),\ \ldots,\ \tilde{\mathbf{x}}_{S-1}(n)], \tag{6}$$

$$\tilde{\mathbf{x}}_0 = \begin{bmatrix} \tilde{x}_0(0) \\ \tilde{x}_0(1) \\ \vdots \\ \tilde{x}_0(N-1) \end{bmatrix},\ \tilde{\mathbf{x}}_1 = \begin{bmatrix} \tilde{x}_1(0) \\ \tilde{x}_1(1) \\ \vdots \\ \tilde{x}_1(N-1) \end{bmatrix},\ \ldots,\ \tilde{\mathbf{x}}_{S-1} = \begin{bmatrix} \tilde{x}_{S-1}(0) \\ \tilde{x}_{S-1}(1) \\ \vdots \\ \tilde{x}_{S-1}(N-1) \end{bmatrix}. \tag{7}$$

After that the blocks of the length N_h are split into non-overlapping segments of the length K and we perform summation of these segments. The total number of segments Q of the length K each within the block of the length N_h is determined as $Q = \lceil N_h/K \rceil$, where $\lceil \cdot \rceil$ denotes rounding towards infinity. As a result we obtain the following expressions:

$$z_{0,m}(r) = \sum_{l=-\infty}^{\infty} \tilde{x}_{0,m}\,(r + lK),\ \ z_{1,m}(r) = \sum_{l=-\infty}^{\infty} \tilde{x}_{1,m}\,(r + lK),\ \ldots,\ z_{S-1,m}(r) = \sum_{l=-\infty}^{\infty} \tilde{x}_{S-1,m}\,(r + lK), \tag{8}$$

where $m = 0, \ldots, P-1$, $r = 0, \ldots, K-1$. The matrix $\mathbf{Z}$ can be written as

$$\mathbf{Z} = \begin{bmatrix} z_{00}(0) & z_{00}(1) & \ldots & z_{00}(K-1) \\ \ldots & \ldots & \ldots & \ldots \\ z_{0,P-1}(0) & z_{0,P-1}(1) & \ldots & z_{0,P-1}(K-1) \\ \ldots & \ldots & \ldots & \ldots \\ z_{S-1,0}(0) & z_{S-1,0}(1) & \ldots & z_{S-1,0}(K-1) \\ \ldots & \ldots & \ldots & \ldots \\ z_{S-1,P-1}(0) & z_{S-1,P-1}(1) & \ldots & z_{S-1,P-1}(K-1) \end{bmatrix}. \tag{9}$$

This matrix consists of sub-matrices and each one has P rows corresponding to blocks of the length N_h of each one-dimensional signal $x_i(n)$, $i = 0, \ldots, S-1$. Thus, $\mathbf{Z}$ has the size $P \cdot S \times K$.

Next, DFT is applied to the matrix $\mathbf{Z}$. This operation can be implemented on the basis of vector DFT algorithms intended for calculating DFT of multichannel data:

$$\mathbf{Y} = \mathrm{VDFT}\,\{\mathbf{Z}\}, \tag{10}$$

where $\mathbf{Y}$ is the resulting matrix after DFT computation, VDFT is the operator of vector DFT.

There are two main approaches to vector DFT computation. One of them suggests application of one-dimensional DFT to each row of $\mathbf{Z}$. Overall, it is necessary to compute $P \cdot S$ one-dimensional DFTs of the length K each. The second approach is based on calculating one-dimensional DFT of the length $P \cdot S \cdot K$:

$$y = \mathrm{DFT}\,(z_{00}(0)\ \ z_{00}(1)\ \ \ldots\ \ z_{00}(K-1)|\ \ \ldots|z_{S-1,P-1}(0)\ \ z_{S-1,P-1}(1)\ \ \ldots\ \ z_{S-1,P-1}(K-1)), \tag{11}$$

where DFT is the operator of one-dimensional DFT. In order to compute one-dimensional DFT, the matrix $\mathbf{Z}$ has to be converted into a vector (a sequence of matrix rows). There are several reasons for this transformation:

- an algorithm of vector DFT computation affects the total size of local memory of a DFT-processor. By reducing multichannel DFT to one-dimensional DFT it is possible to minimize computational memory and optimize the structure of a DFT-processor;

- computation of one-dimensional DFT of the length $P \cdot S \cdot K$ allows us to implement the computational procedure in a DFT-processor more efficiently in comparison with computing $P \cdot S$ different DFTs of the length K.

After computing DFT (11) of the vector $\mathbf{y}$ it is necessary to transform the result back into a matrix. The dimension of $\mathbf{Y}$ will coincide with the that of $\mathbf{Z}$. Matrix $\mathbf{Y}$ will consist of several sub-matrices:

$$\mathbf{Y}(r) = [\mathbf{Y}_0(r),\, \mathbf{Y}_1(r),\, \ldots,\, \mathbf{Y}_{S-1}(r)]^T, \tag{12}$$

$$\mathbf{Y}_i(r) = \begin{bmatrix} y_{i,0}(0) & y_{i,0}(1) & \ldots & y_{i,0}(K-1) \\ & \ldots & & \\ y_{i,P-1}(0) & y_{i,P-1}(1) & \ldots & y_{i,P-1}(K-1) \end{bmatrix}, \; i = 0, \; , S-1; \; r = 0, \ldots, K-1. \tag{13}$$

Finally, we have to form the matrix $\tilde{\mathbf{Y}}(r)$ by multiplying all the rows of $\mathbf{Y}(r)$ by the factor W_K^{-krM}:

$$\tilde{\mathbf{Y}}_i(r)\Big|_{k-\text{th line}} = \mathbf{Y}_i(r)|_{k-\text{th line}}\, W_K^{-krM}, \; r = 0, \ldots, K-1; \; k = 0, \ldots, P-1. \tag{14}$$

To summarize, multichannel WOLA-algorithm can be presented as a sequence of steps:

1) weighting the multichannel signal $\mathbf{x}(n)$ in order to obtain $\tilde{\mathbf{x}}(n)$ according to (4)–(7).
2) splitting the multichannel signal (6) into non-overlapping segments of the length K and summation of segments of each one-dimensional signal according to (8);
3) application of vector DFT (10) to the matrix $\mathbf{Z}$ using one-dimensional DFT (11) of the length $P \cdot S \cdot K$ and computation of the matrix $\mathbf{Y}(r)$ according to (12)–(13);
4) computation of the matrix $\tilde{\mathbf{Y}}(r)$ according to (14).

3. SOFTWARE-HARDWARE IMPLEMENTATION OF THE WOLA FOR VECTOR SIGNAL PROCESSING

The crucial factor of computational complexity for software-hardware implementation of a multichannel filter bank is a low-frequency prototype filter (LF-prototype filter), which is responsible for forming the required magnitude response for one filter bank channel. The order of a LF-prototype filter can be found depending on the parameters of a wideband monitoring system.

The LF-prototype filter has been designed by the window technique. The parameters of the LF-prototype will define the parameters of the whole filter bank. First, it is necessary to provide small flatness of the magnitude response (no more than 1 dB) in the passband, large suppression (at least 80 dB) in the stopband and a high shape factor (the LF-prototype must have high selectivity). Below we have given the exact parameters of our LF-prototype: frequency band for wideband monitoring: [0; 1] MHz; number of filter bank channels: $K = 320$; channel bandwidth: $\Delta f = 3125$ Hz; signal suppression at the edge of the passband of a LF-prototype filter: 1 dB; signal suppression at the edge of the stopband of a LF-prototype filter: 120 dB; magnitude response flatness: 0.01 dB; rectangularity shape factor: 1.25. As a result we have designed a LF-prototype filter of the order 6245.

A filter bank performs parallel processing of an input signal and therefore in order to improve the efficiency of its hardware implementation it is reasonable to employ computers with a parallel structure. Among such computers are field programmable gate arrays (FPGA) and processing devices based on the Compute Unified Device Architecture technology (CUDA technology) [9].

CUDA technology allows us to develop software by minimizing temporal resources and satisfying high standards of information processing in real time. The main idea behind CUDA is application of a set of parallel graphical processors (Graphics Processing Unit — GPU) for handling non-graphical tasks. GPU is a specialized computational device, which has the following properties: GPU is a co-processor for CPU; GPU has its own memory; GPU allows parallel performance of a large number of data flows (in this context, a flow means several parts of a program run in a parallel way).

The experiment was made on a personal computer with the following configuration: processor Intel Core i7-3630QM (IvyBridge) 2.4 GHz; RAM DDR3 16 Gb, operating system Windows 7 64 bits; video card NVidia GeForce GT650M (384 core CPU, core frequency 850 MHz, video card memory 2 Gb).

In Table 1 there are estimates of WOLA performance time using CUDA and CPU. When we were applying CUDA we also took into consideration the data transfer time between RAM and video card memory.

Table 1: Comparison of CPU and CUDA performance.

Data size (samples/Mbytes)	**CPU and CUDA performance**		
	CUDA without data transfer between RAM and video card memory	**CUDA with data transfer between RAM and video card memory**	**CPU**
3000000 / 11 Mbytes	0.293 sec.	0.508 sec.	46.931 sec.
8000000 / 30 Mbytes	0.320 sec.	0.895 sec.	122.656 sec.
16000000 / 61 Mbytes	0.346 sec.	1.496 sec.	246.834 sec.
30000000 / 114 Mbytes	0.437 sec.	2.593 sec.	459.354 sec.

Analysis of the data in Table 1 allows us to draw the following conclusions:

- application of CUDA leads to significant computational time reduction in comparison with CPU;
- increase of data size results in temporal cost rise and therefore the greater a signal set is the more reasonable it is to apply CUDA.

It is necessary to mention that the time for data transfer between RAM and video card memory can sometimes reach half of the total processing time, which a sort of a disadvantage of the CUDA technology.

Figure 1 illustrates a general structure of a digital filter bank with FPGA (field programmable gate array) implementation.

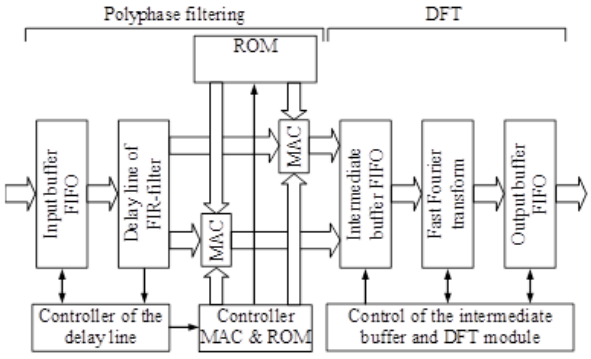

Figure 1: General structure of a digital filter bank with FPGA implementation.

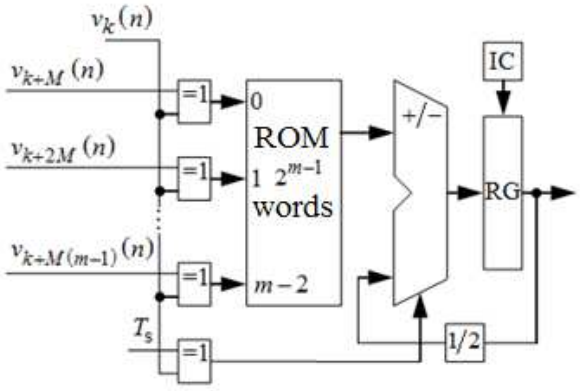

Figure 2: General structure of a digital filter bank with FPGA implementation.

By analyzing this figure we can conclude that one of the crucial factors of hardware implementation of a digital filter bank is the order of a LF-prototype filter, which defines the magnitude response of a FIR-filter of a single channel. Hardware resources of computers used for digital filter bank implementation are limited (the number of multipliers is the most critical) and the main task consists in synthesizing efficient FIR-filters (the criterion is minimum computational complexity), which will make it possible to reduce software-hardware costs and to accelerate signal analysis. In the case of using FPGA such an approach allows us to reduce the number of devices involved (primarily, the number of FPGA multipliers) and increase the number of those on a chip.

One more practical approach to digital filter bank implementation using FPGA is application of distributed arithmetic (DA) [10]. The main difference of this approach from direct implementation based on MAC cores is the fact that there are no multipliers in the DA.

Computation of the k-th polyphase component of a multichannel filter bank corresponds to computing the scalar product of $\mathbf{v}_{k+mi}(n)$ (output of a delay chain) and $\mathbf{g}_{k+mi}$ (coefficients of the

polyphase component):

$$Y_k(n) = \sum_{i=0}^{m-1} \mathbf{v}_{k+mi}(n)\mathbf{g}_{k+mi}. \tag{15}$$

Consider the scalar product of two vectors: $Y = \sum\limits_{k=0}^{L-1} \mathbf{A}_k\mathbf{x}_k$ and let us assume that that the input data $\mathbf{x}_k$ are the numbers in the complement code normalized to one: $x_k = -b_{k0} + \sum\limits_{n=1}^{N-1} b_{kn}2^{-n}$, where b_{kn} are bits for numeric representation of x_k. Having made a few transformations we can obtain the following expression:

$$Y = -\sum_{k=0}^{L-1} \mathbf{A}_k b_{k0} + \sum_{n=1}^{N-1} \left[\sum_{k=0}^{L-1} A_k b_{kn}\right] 2^{-n}. \tag{16}$$

The sum $\sum\limits_{k=0}^{L-1} A_k b_{kn}$ can take 2^L different values, which can be computed in advance and saved in the memory. Then the vector $\mathbf{b}_n = [b_{0n}\ b_{1n}\ \ldots\ b_{(L-1)n}]$ is the address where the value $\sum\limits_{k=0}^{L-1} A_k b_{kn}$· is stored. Thus, we need the memory size of 2^L cells of the type "adder-subtractor" and a register for storing intermediate results. The memory size can be reduced twice if we interpret the input data as those represented in the offset binary code.

Each polyphase component of a prototype-filter is a low-order FIR-filter, which can be implemented efficiently using the distributed arithmetic. The polyphase component implementation using the distributed arithmetic is shown in Fig. 2. Data $v_k(n)$ from a block of registers are sent to the input in the sequential code beginning from a low-order bit. The signal T_s follows the sign bit and after that the register RG will store the result. The control device for this circuit can be implemented using a counter. The block IC (initial condition) stores the initial value loaded in the register RG prior to the beginning of the circuit operation for representing filter coefficients in an offset binary code for reducing the size of ROM [10]. When filtering is performed, data from the register RG shifted to the right by one bit (this is equivalent to dividing by 2) are added to the contents of ROM. The number of polyphase components is determined by the number of sub-channels of one channel of a multi-channel filter bank M and the order of each polyphase component is determined by the parameter m.

Filter banks are efficient for preprocessing when the original signal is divided into a given number of parallel channels with the required parameters. Next, we can select any of the channels and apply specialized processing algorithms that are needed for solving wideband monitoring tasks. Thus, application of a filter bank makes it possible to obtain a number of parallel devices and each device improves the performance and efficiency of software-hardware wideband monitoring packages.

4. CONCLUSION

In the present paper we have provided the WOLA algorithm for processing vector (multichannel) signals in real time. We have provided the results of software-hardware implementation using CUDA and we have shown that CUDA performs much better for long data sets (the processing time has been reduced). Hardware resources for multichannel signal processing implementation can be reduced using the distributed arithmetic.

ACKNOWLEDGMENT

The paper is supported by the grant of the Russian Foundation for Basic Research ("My first grant") No. 14-07-31250/15 and Contract No. 02.G25.31.0058 dated 12.02.2013 (Board of Education of Russia).

REFERENCES

1. Porcino, D. and W. Hirt, "Ultra-wideband radio technology: Potential and challenges ahead," *IEEE Communications Magazine*, Vol. 41, No. 7, 66–74, 2003.
2. Li, L. and Y. H. Yen, "Wideband monitoring and measuring system for optical coatings," *Applied Optics*, Vol. 28, No. 14, 2889–2894, Jul. 15, 1989.

3. Allen, B., T. Brown, K. Schwieger, E. Zimmermann, W. Q. Malik, D. J. Edwards, L. Ouvry, and I. Oppermann, "Ultra wideband: Applications, technology and future perspectives," *Proc. Int. Workshop Convergent Tech.*, Oulu, Finland, Jun. 2005.
4. Xu, Y., Y. Lu, H. Zhang, and Y. Wang, "An overview of ultra-wideband technique application for medial engineering," *2007 IEEE/ICME International Conference on Complex Medical Engineering (CME)*, Beijing, May 2007.
5. Hu, X.-Q., Y.-M. Chen, and J.-F. Tang, "Apparatus for wideband monitoring of optical coatings and its uses," *Applied Optics*, Vol. 28, No. 14, 2886–2888, 1989.
6. Antoniou, A., *Digital Filters: Analysis, Design, and Applications*, 689, McGraw-Hill, USA, 1993.
7. Mitra, S. K., *Digital Signal Processing: A Computer-Based Approach*, 940, McGraw-Hill, USA, 1998.
8. Crochiere, R. E. and L. R. Rabiner, *Multirate Digital Signal Processing*, 411, Prentice Hall, USA, 1983.
9. Kirk, D. B. and W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, 280, Morgan Kaufmann, USA, 2010.
10. White, S. A., "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Magazine*, Vol. 6, 4–19, 1989.