World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:2, 2008

FPGA Based Parallel Architecture for the Computation of Third-Order Cross Moments

Syed Manzoor Qasim, Shuja Abbasi, Saleh Alshebeili, Bandar Almashary and Ateeq Ahmad Khan

***Abstract*—**Higher-order Statistics (HOS), also known as cumulants, cross moments and their frequency domain counterparts, known as poly spectra have emerged as a powerful signal processing tool for the synthesis and analysis of signals and systems. Algorithms used for the computation of cross moments are computationally intensive and require high computational speed for real-time applications. For efficiency and high speed, it is often advantageous to realize computation intensive algorithms in hardware. A promising solution that combines high flexibility together with the speed of a traditional hardware is Field Programmable Gate Array (FPGA). In this paper, we present FPGA-based parallel architecture for the computation of third-order cross moments. The proposed design is coded in Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) and functionally verified by implementing it on Xilinx Spartan-3 XC3S2000FG900-4 FPGA. Implementation results are presented and it shows that the proposed design can operate at a maximum frequency of 86.618 MHz.

***Keywords*—**Cross moments, Cumulants, FPGA, Hardware Implementation.

I. Introduction

HIGHER-ORDER STATISTICS (HOS) (greater than two) also known as cumulants, cross moments and their associated Fourier transforms known as higher-order spectra or poly spectra, are commonly used as powerful signal processing tool in diverse application domains such as digital communications, sonar, radar, speech, biomedical, geophysical, plasma physics, image processing, signal reconstruction, array processing, harmonic retrieval, time-delay estimation, adaptive filtering and blind equalization [1]–[6] etc.

HOS, unlike the second-order statistics are well known for their robustness to additive Gaussian noise and their ability to preserve phase information. However, the computational complexity involved in the estimation of HOS far exceeds that of conventional second-order statistics because HOS are multidimensional functions [6].

Traditionally, Digital Signal Processing (DSP) algorithms are coded in C language or MATLAB, and implemented on a general purpose (programmable) DSP chips for low-rate applications. For moderate rates, special purpose DSP chips are used. However, for higher rates the software implementation is transformed either manually or compiled automatically into a high-level hardware description language such as VHDL or Verilog and implemented in an Application Specific Integrated Circuit (ASIC) or Field Programmable Gate Array (FPGA). General purpose DSP chips often lack the performance necessary for moderate sampling rates, and ASIC approaches are limited in flexibility and may not be cost effective for many applications. Recently, FPGAs have become an attractive alternative for realization of computation intensive algorithms.

Recent advancement in FPGA technology has resulted in enormous possibilities for the implementation of sophisticated algorithms of high complexity, in a variety of important applications, by using low cost, high performance and high speed reconfigurable hardware. FPGAs have become one of the prevailing technologies for fast prototyping and implementation of digital systems. Being dynamically reconfigurable, FPGAs provide additional interesting features to implement complex algorithms in hardware with performance that exceeds both general-purpose and DSP implementations.

In hardware design community, it is well known fact that by assigning computation intensive tasks to hardware (FPGA) and exploiting the parallelism in algorithms yields a significant speedup in computation time. This paper presents an FPGA based architecture for high speed computation of third-order cross moments. This paper exploits the inherent parallelism of FPGA technology as well as the algorithm which is based on the idea of formulating the computation of cross moments as a series of matrix multiplication operations [7].

The remainder of the paper is organized as follows. Section II describes the related work done in this area. The architecture of Spartan-3 FPGA is described briefly in Section III. Algorithm for the computation of third-order cross moments is discussed in Section V. In Section VI, we discuss the architecture for the computation of third-order cross

Manuscript received May 6, 2008; revised June 6, 2008. This work was supported by the Research Centre, College of Engineering, King Saud University under research grant no. 426/7.

Syed Manzoor Qasim is with the Electronics Group, Department of Electrical Engineering, College of Engineering, P.O.Box-800, King Saud University, Riyadh 11421, Kingdom of Saudi Arabia (phone: +966-1-4676774; fax: +966-1-4676757; E-mail: smanzoor@ ksu.edu.sa).

Saleh Alshebeili is with the Communication Group, Department of Electrical Engineering, College of Engineering, King Saud University, Kingdom of Saudi Arabia (E-mail: dsaleh@ksu.edu.sa).

Shuja Abbasi, Bandar Almashary and Ateeq Ahmad Khan, are with the Electronics Group, Department of Electrical Engineering, College of Engineering, King Saud University, Kingdom of Saudi Arabia (E-mail: {abbasi,bmashary,akhan}@ksu.edu.sa).

moments. FPGA implementation results are summarized in Section VI. Finally, conclusions are drawn in section VII.

II. Related Work

During the last few years substantial amount of work has been generated towards the design and development of application specific systems to accelerate the computation of higher-order statistics using Very Large Scale Integration (VLSI) architectures and other emerging technologies [5]–[10]. However, to the best of our knowledge, none of the previous work has used FPGA for the computation of third-order cross moments.

Ahmed *et al*. [5] presented a computationally efficient VLSI architecture for the computation of third-order cumulants. Their architecture is based on the systolic array and implemented with 1.0 μm CMOS technology. Their design operates at a speed of 5.2 MHz. Another efficient approach for computing third-order cumulants based on matrix multiplications was presented by Turaigi *et al*. [6]. They used fast systolic array system to compute third-order cumulants.

Alshebeili [7] presented a novel approach based on matrix multiplication for the computation of higher order cross moments. The computation of cross moments was formulated as a series of matrix multiplication operations to take advantage of well established systolic array techniques for the computation of matrix multiplication.

Turaigi *et al*. [8] presented a concurrent systolic array system for the computation of higher-order moments. The system was used for the computation of second, third and fourth-order moments simultaneously. It was implemented in CMOS VLSI technology with an operating speed of 3.9 MHz. Aloqeely *et al*. [9] used a new approach based on matrix multiplication for the estimation of third-order cumulants using linear systolic array.

Stellakis *et al*. [10] used adaptive sliding window time-and-order recursive algorithm for the computation of higher-order moments up to the fourth order. The algorithm was once again mapped to a systolic array.

III. Target FPGA Architecture

The Spartan-3 FPGA consists of an array of Configurable Logic Blocks (CLBs), which are the basic elements that can be programmed to perform various logic functions. Each CLB is coupled with a programmable interconnect switch matrix that connects the CLB to adjacent and nearby CLBs [11]–[12].

Each CLB contains four logic slices, where each logic slice usually consists of two four-input Look Up Tables (LUTs), two configurable flip-flops, some muxes, and other control logic. In addition to the CLBs and the switch matrices, the Spartan-3 FPGA have a number of higher–level logic blocks such as block RAMs (BRAMs), 18-bit multipliers, digital clock managers (DCMs) and even CPUs [12].

IV. Design Flow

An FPGA design flow is the process of turning an FPGA design into a correctly timed bitstream file used to program the FPGA. In order to realize any algorithm on an FPGA it must be programmed (configured) first. To achieve this, a design methodology is adopted. Usually, design entry is done using hardware description language (HDL) such as VHDL and Verilog. In this paper, the design entry is done using VHDL. The objective is to make the system description independent of the physical hardware such that it can be used on other FPGAs and even on Application Specific Integrated Circuits (ASICs). Once a design has been completed it is simulated to verify the correct operation. A netlist is generated from the design and is mapped onto the FPGA using synthesis, place and route and optimizing tools. Mapping produces a bit-stream file that is used to program the FPGA [12]. The steps followed are summarized in Fig. 1.

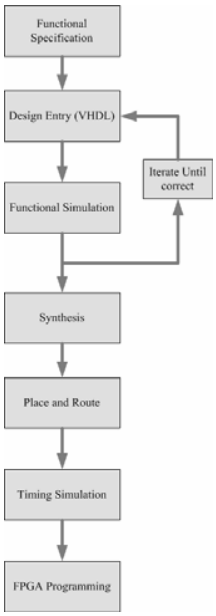

Fig. 1 Design Flow

V. Problem Formulation

In this section, we discuss the basic definition and formulation of third-order cross moments. The third-order cross moment function m_3 of a stationary random process $x(n)$ with samples $x_0(n)$, $x_1(n)$ and $x_2(n)$ is defined as [1]:

$$m_3(\tau_1, \tau_2) = E\{x_0(n)\, x_1(n+\tau_1)\, x_2(n+\tau_2)\} \quad (1)$$

where, $E\{\cdot\}$ denotes statistical expectation [1]. If $x(n)$ is a zero-mean stationary process, then the third-order cross moments are identical to the third-order cross cumulants and are computed from the given formula [2].

$$m_3(\tau_1, \tau_2) = \frac{1}{N}\sum_{n=l_1}^{l_2} x_0(n)\, x_1(n+\tau_1) x_2(n+\tau_2) \quad (2)$$

where, N is the length of each data record, $l_1 = \max\{0, -\tau_1, -\tau_2\}$, and $l_2 = \min\{N-1, N-\tau_1-1, N-\tau_2-1\}$.

In this paper, third-order cross moments are computed based on the idea of converting (2) into matrix multiplication [7]. Let M_i be a square matrix whose elements are samples of third-order cross moments defined in (2). M_i is given by (3) where $i = -q, -q + 1, \ldots, q,$ and q is the maximum lag of third-order cross moment function.

$$M_i = \begin{bmatrix} Nm_3(-q,-q,i) & Nm_3(-q,-q+1,i) & \cdots & Nm_3(-q,q,i) \\ Nm_3(-q+1,-q,i) & Nm_3(-q+1,-q+1,i) & \cdots & Nm_3(-q+1,q,i) \\ \vdots & \vdots & \ddots & \vdots \\ Nm_3(q,-q,i) & Nm_3(q,-q+1,i) & \cdots & Nm_3(q,q,i) \end{bmatrix} \quad (3)$$

By substituting (2) into (3), it can be seen that M_i can be written as

$$M_i = XY_iZ \quad (4)$$

where, X is a $(2q + 1)$ x N rectangular matrix which is given by (5).

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & \cdots & 0 & x_1(0) & \cdots & x_1(N-q-1) \\ 0 & \cdots & \cdots & x_1(0) & x_1(1) & \cdots & x_1(N-q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & x_1(0) & \cdots & \cdots & \cdots & \cdots & x_1(N-2) \\ x_1(0) & x_1(1) & \cdots & \cdots & \cdots & \cdots & x_1(N-1) \\ x_1(1) & x_1(2) & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1(q) & x_1(q+1) & \cdots & x_1(2q-1) & x_1(2q) & \cdots & 0 \end{bmatrix} \quad (5)$$

$$\mathbf{Z} = \begin{bmatrix} 0 & 0 & \cdots & x_2(0) & x_2(1) & \cdots & x_2(q) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & x_2(0) & \cdots & \cdots & \cdots & \cdots & x_2(2q-1) \\ x_2(0) & x_2(1) & \cdots & \cdots & \cdots & \cdots & x_2(2q) \\ x_2(1) & x_2(2) & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2(N-q-1) & x_2(N-q) & \cdots & x_2(N-1) & 0 & \cdots & 0 \end{bmatrix} \quad (6)$$

Z is an N x $(2q + 1)$ rectangular matrix which is given by (6) and Y_i is a diagonal square matrix whose elements are $x_0(n) \cdot x_3(n + i)$, where n = 0, 1, …, $N - 1$. X and Z are Hankel matrices. Thus the computation of third-order cross moment function is reduced to the computation of $(2q + 1)$ different matrices whose elements are obtained by multiplying three matrices as given in (4).

VI. Architectural Design

Third-order cross moments are evaluated by computing the entries for matrix M_i for different values of i. The entries for matrix M_i can be calculated by performing the matrix multiplication XY_iZ. The system block diagram for the computation of third-order cross moments is shown in Fig. 2.

It consists of two arrays MM_1 and MM_2. The first array MM_1 performs the multiplication of Hankel matrix X by diagonal matrix Y_i and feeds the results to array MM_2. The second array MM_2, on the other hand multiplies XY_i by Z**.**

We use the 2D systolic array based architecture as shown in Figs. 3 and 4 for the matrix multiplication. Systolic arrays speedup computationally intensive algorithms with inherent parallelization, by exploiting data parallelism. The major features of systolic array are: (1) simple and regular design; (2) concurrent design; and (3) nearest neighbor communication. FPGAs can be used efficiently to implement fine grain systolic arrays since they inherently possess the same regular structure.

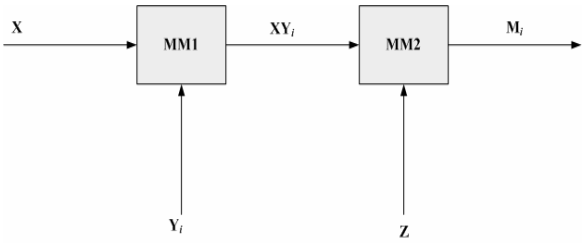

Fig. 2. Block Diagram

Fixed point numbers are used for the elements of the matrices. Matrix Y_i is a diagonal square matrix with entries $x_0(0) \cdot x_3(i)$, $x_0(1) \cdot x_3(1 + i)$, $x_0(2) \cdot x_3(2 + i)$, $\ldots x_0(n) \cdot x_3(N\text{-}1 + i)$, where $x_0(0) \cdot x_3(i)$ is the non zero element in the first row, $x_0(1) \cdot x_3(1 + i)$ is the non zero element in the second row, and so on.

Multiplying the Hankel matrix X by the diagonal square matrix Y_i is equivalent to multiplying the first diagonal element by the entries of first row of X, the second diagonal element by the entries of the second row of X and so on.

Figs. 3 and 4 shows the systolic architecture for array MM_1 and MM_2 for $N_1 = 3$ and $N_2 = 3$ respectively. MM_1 and MM_2 consist of nine identical Processing Elements PE_1 and PE_2 respectively. Each processing element PE_1 consists of multiplier whereas PE_2 contains Multiply-Accumulate (MAC) unit and each MAC unit consists of a multiplier, adder, and a register for storage. 2's complement method is used for negative numbers.

The function of each PE_1 in MM_1 array is to multiply the diagonal element of Y_i [Y_{11}, Y_{22}, Y_{33}] by one element of matrix X during each clock period. First column PE_1s are responsible for producing first column of the product XY_i referred to as W in the Fig. 3, second column generates the second column and so on. The entries are stored in an output buffer to be used later by next array MM_2.

Similarly, array MM_2 as shown in Fig. 4 performs the final multiplication of (XY_i) with Z i.e., WZ using the same technique as discussed for array MM_1. The elements of matrix M_i represent the samples of third-order cross moments. For N = 3, M_i is represented in matrix form as (7).

$$M_i = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (7)$$

$$P = N^2 + N^2 = 2N^2 \quad (8)$$

The total number of PEs, P required for the computation of third-order cross moments depends on N. The algorithm formulated as product of three matrices becomes computationally intensive [7] and the complexity further increases with the parameter q and number of samples N. The simulation results of the proposed system after FPGA implementation are shown in Fig. 5.

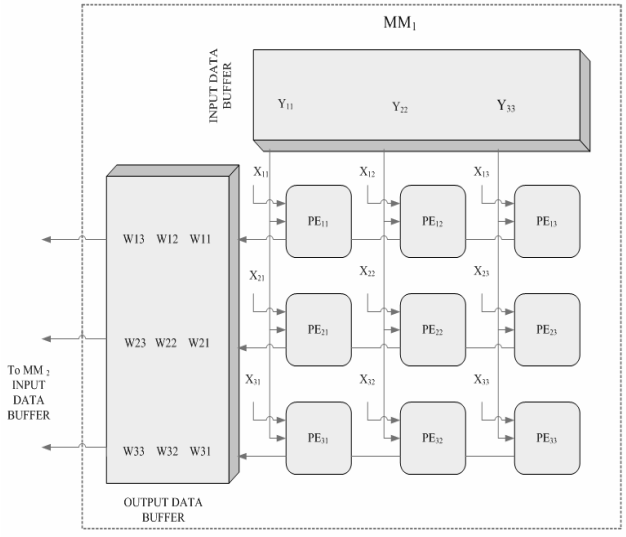

Fig. 3 Systolic Architecture of Matrix Multiplier MM_1

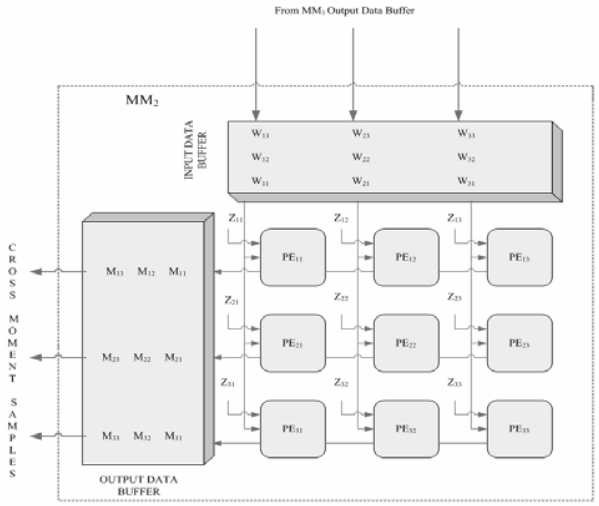

Fig. 4 Systolic Architecture of Matrix Multiplier MM_2

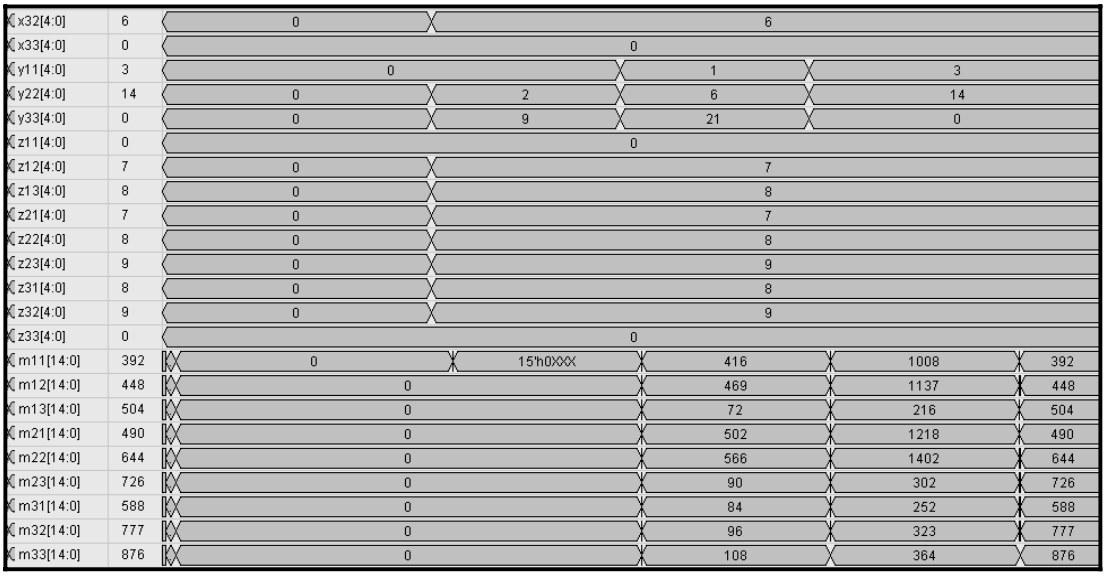

Fig. 5 Simulation Results

VII. FPGA Implementation Results

The algorithm is coded in VHDL and realized in FPGA using Xilinx ISE 9.1i to synthesize and place-and-route the design. Xilinx ISE simulator is used to verify the design in simulation before it is implemented on Xilinx Spartan-3 FPGA. Because of their exceptionally low cost and inherent reconfigurability, Spartan-3 FPGAs are ideally suited for signal processing applications such as computation of third-order cross moments. The design achieves a top frequency of 86.618 MHz.

The proposed architecture shows significant improvement in speed as compared to existing VLSI architectures [5], [8]. The implementation results generated by Xilinx ISE 9.1i is listed in Table I.

VIII. Conclusions

In this paper, an FPGA based architecture for the computation of third-order cross moments based on novel matrix multiplication algorithm is presented. The algorithm is implemented on Xilinx Spartan-3 FPGA, a low cost FPGA that are now used in applications that were once relegated strictly to ASIC domain. The maximum operating speed of the design as reported by the synthesis tools is 86.618 MHz. The use of FPGA technology has proven to be an attractive alternative for efficient and fast computation of third-order cross moments under real-time constraints.

TABLE I
IMPLEMENTATION RESULTS ON XILINX SPARTAN-3 FPGA DEVICE

XC3S2000 FG900- 4	
Resources	Utilization
Number of Slices	144 out of 20480
Number of 4 Input LUTs	270 out of 40960
Number of Multipliers (18X18s)	36 out of 40
Minimum Period (ns)	11.545
Maximum Frequency (MHz)	86.618
Power Consumption (mW)	100

References

[1] C. L. Nikias and A. P. Petropulu, *Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[2] S. A. Alshebeili, "Estimation of higher-order moments via discrete orthogonal laguerre functions," in *Proc. of 3rd IEEE Int. Conf. on Signal Processing*, vol. 1, Oct. 1996, pp. 11–14.

[3] P. Paajarvi and J. P. Leblanc, "Online adaptive blind deconvolution based on third-order moments," *IEEE Signal Processing Letters*, vol. 12, No.12, Dec. 2005, pp. 863 − 866.

[4] L. Wenkai, "Blind channel estimation using zero-lag slice of third-order moment," *IEEE Signal Processing Letters*, vol.12, No.10, Oct. 2005, pp. 725 − 727.

[5] R. E. Ahmed, M. A. Al-Turaigi, and S. A. Alshebeili, "VLSI Architecture for computing third-order cumulants," *International Journal of Electronics*, vol. 77, No. 1, 1994, pp. 95-104.

[6] M. A. Al-Turaigi and S. A. Alshebeili, "A high-speed systolic array for computing third-order cumulants," *Canadian Journal of Electrical and Computer Engineering*, vol. 22, no. 1, 1997, pp.19-23.

[7] S. A. Alshebeili, "Computation of higher-order cross moments based on matrix multiplication," *Journal of the Franklin Institute*, 338, 2001, pp. 811-816.

[8] M. A. Al-Turaigi, R. E. Ahmed, and S. A. Alshebeili, "A concurrent system for the computation of higher-order moments," *Journal of Circuits, Systems and Signal Processing*, vol. 18, no. 2, 1999, pp. 111 − 130.

[9] M. A. Aloqeely, M. A. Al-Turaigi, and S. A. Alshebeili, "A new approach for the design of linear systolic arrays for computing third-order cumulants," *Integration: the VLSI Journal*, vol. 24, 1997, pp. 1−17.

[10] H. M. Stellakis and E.S. Manolakos, "Adaptive computation of higher order moments and its systolic realization," *International Journal of Adaptive Control and Signal Processing*, vol. 10, 1996, pp. 283–302.

[11] T. Tuan, S. Kao, A.Rahman, S. Das, and S.Trimberger, " A 90 nm low-power FPGA for battery-powered applications," in *Proc. of 14th ACM/SIGDA Int. Symp. on FPGAs,* Feb. 2006, pp. 3-11.

[12] S. M. Qasim and S. A. Abbasi, "A Novel FPGA-based approach for digital waveform generation using orthogonal functions," *Journal of Circuits, Systems and Computers,* vol.16, no. 6, 2007, pp. 895-909.

Syed Manzoor Qasim received B.Tech and M.Tech Degrees in Electronics Engineering from Zakir Hussain College of Engineering and Technology, Aligarh Muslim University, India in 2000 and 2002 respectively. Later in 2002, he joined the Electrical Engineering Department, King Saud University as a Researcher. His areas of interest include Digital VLSI System Design and Reconfigurable computing using FPGAs. He is a member of the Institution of Electronics and Telecommunication Engineers, India.

Shuja Abbasi (M'89–SM) is a Professor in the Electrical Engineering Department, King Saud University. He obtained the B.Sc and M.Sc Degrees in Electrical Engineering from Aligarh Muslim University, India in 1970 and 1972 respectively. He received the Ph.D Degree in Microelectronics from University of Southampton, England in 1980. He is a senior member of IEEE and Fellow of Institution of Electronics and Telecommunication Engineers, India.

Saleh Alshebeili (S'89–M'91–SM'96) is a Professor in the Electrical Engineering Department, King Saud University. He received the B.Sc and M.Sc Degrees in Electrical Engineering from King Saud University in 1984 and 1986, respectively. He received the Ph.D. Degree in Electrical Engineering from University of Toronto, Ontario, Canada, in 1992. His research interests are in the area of Signal Processing with applications to Communications, Speech, and Image Processing. He is a senior member of IEEE.

Bandar Almashary is an Associate Professor in the Electrical Engineering Department, King Saud University. He received the BS and MS Degrees in Electrical Engineering from King Saud University, Saudi Arabia. He received the Ph.D Degree in Optoelectronics from University of Pittsburgh, USA in 1996. He is a member of IEEE.