J Sign Process Syst (2013) 70:209–218
DOI 10.1007/s11265-012-0705-3

A Hardware-Efficient Algorithm for Real-Time Computation of Zadoff–Chu Sequences

Mohammad M. Mansour

Received: 11 May 2012 / Revised: 5 July 2012 / Accepted: 18 September 2012 / Published online: 5 October 2012

Abstract A hardware-efficient algorithm and architecture for computing Zadoff–Chu (ZC) sequence elements on-line using the CORDIC algorithm are proposed. Zadoff–Chu sequences possess good correlation properties that are essential in a variety of engineering applications, such as establishing timing synchronization between a mobile terminal and a base station in the emerging 3GPP long-term evolution (LTE) physical layer standard for cellular communications. The proposed algorithm computes ZC-sequence elements both in time domain and frequency domain using a simple duality relationship. Algorithm transforms are employed to compute the elements recursively and eliminate the need for multipliers with non-constant terms. A reconfigurable hardware architecture was implemented and applied in a searcher block for detecting the physical random access channel (PRACH) in LTE. The PRACH provides a mechanism for a mobile to establish initial access along with uplink synchronization by transmitting a preamble that is constructed from ZC sequences. The proposed architecture is capable of generating these preambles on the fly with high accuracy, eliminating the need for storing a large number of long complex-valued ZC sequence elements. Simulation results demonstrate that the proposed architecture is capable of achieving detection error rates for LTE PRACH that are close to ideal rates achieved using floating point precision. (The work has been presented in part in Mansour (2009).)

M. M. Mansour (✉)
American University of Beirut, Beirut, Lebanon
e-mail: mmansour@aub.edu.lb

1 Introduction

In cellular communications, the physical random access channel (PRACH) is a common uplink channel used by mobile users within a cell to establish initial access to a base station, along with uplink synchronization to compensate for round-trip delays to the base station. In the emerging Third Generation Partnership Project (3GPP) long-term evolution (LTE) physical layer standard [2], the mechanism is based on mobile users transmitting a randomly chosen preamble over a dedicated time-frequency resource on the PRACH. A pool of known preambles is allocated to a base station within a cell. A PRACH processor in the base station attempts to detect a transmitted preamble by performing matched filtering across the pool of preambles allocated to the base station. The matched filtering is performed as a cross-correlation of the PRACH signal with each of the known preambles dedicated to the base station. The cross correlations provide a metric to be compared to a threshold, from which the presence of a preamble can be detected and the mobile's timing offset relative to the base station can be estimated.

An important requirement is that the system must be capable of supporting a large number of users per cell with quasi-instantaneous access to the radio resources, and sustain a good detection probability, while maintaining a low false alarm rate. Hence, preambles must be constructed using sequences that possess good periodic correlation properties. One candidate sequence is the well-known Zadoff–Chu sequence [3], which belongs to a class of sequences called constant amplitude zero auto-correlation (CAZAC) sequences. These sequences are currently employed in LTE PHY layer [2] to construct PRACH preambles. Zadoff–Chu

sequences are complex exponential codes whose discrete auto-correlations are zero for all non-zero lags, with no restrictions on code lengths [3].

However, a major disadvantage is that ZC sequences are difficult to generate in real-time due to the nature of their construction. Current implementations typically resort to pre-computing these sequences offline, quantizing them to the required precision, and storing them in memory. For example, in LTE with 3-sector cells, and where a pool of 64 preambles of length 839 are allocated to each sector (Format-0 preambles) [2], a memory storage of 2.5 M-bits is needed to store these complex-valued sequences, assuming 8-bit quantization. Other techniques for computing trigonometric functions use low-precision arithmetic components to approximate high precision results, and then to correct the approximation error occasionally using high-precision arithmetic components [4]. Many of these techniques are based on Volder's CORDIC algorithm [5], and since complexity rather than precision is the primary interest for us in this work, the basic CORDIC algorithm for computing trigonometric functions is utilized.

The LTE standard must also support preambles of length 139, so the total memory needed is around 3 M-bits. In practical implementations, the searcher block is part of a large and complex System-on-Chip (SoC) modem designed to serve multiple users in a cell-site. In order to sustain the stringent requirements in LTE, such modem SoCs are primarily designed to achieve high-throughput performance. However, it is well-known that power consumption has become a barrier to performance scaling, mainly due to power consumption in on-chip memory. Moreover, modem SoCs are known to be memory-dominant and not logic-dominant chips. Hence global optimization of on-chip memory resources is a crucial step in high-speed modem designs. Through algorithmic and architectural optimizations, it is possible to eliminate large and power-consuming memory blocks used to store static data on-chip, and replace them with smaller and more power-efficient functional units to generate the required data in real-time.

In this paper, a hardware-efficient algorithm for computing ZC sequences with high accuracy in real-time is proposed, which is the main contribution of this work. The algorithm is based on the CORDIC (COordinate Rotation DIgital Computer) algorithm for computing complex exponentials using only shift and add operations [5]. Using algorithm transforms, multipliers with non-constant terms are eliminated. Independently, by exploiting a duality relationship between ZC sequences in time domain and frequency domain, the algorithm can easily compute ZC sequences in both domains. An important consequence is that the correlation operation for detection can be computed more efficiently by passing to the frequency domain. Correlation in frequency domain corresponds to multiplying the FFT of the received noisy preamble with a preamble generated using ZC sequences in frequency domain, and then taking the inverse FFT.

The remainder of the paper is organized as follows. Section 2 introduces ZC sequences and analyzes their correlation properties. In Section 3, an optimized algorithm for computing ZC sequences is proposed, and a corresponding hardware architecture is developed. To demonstrate the efficiency and accuracy of the proposed algorithm, Section 4 presents an application of ZC sequences in constructing PRACH preambles in LTE, and their corresponding detection procedure. In Section 5, an LTE searcher block using the proposed architecture is implemented, and hardware simulation results are presented. Finally, Section 6 provides some concluding remarks.

2 Zadoff–Chu Sequences

A Zadoff–Chu (ZC) sequence $z_\gamma[n]$ is defined as [3]

$$z_\gamma[n] = \begin{cases} \exp\left(-j\dfrac{2\pi\gamma}{N}\dfrac{n(n+2q)}{2}\right), & N \text{ even}; \\ \exp\left(-j\dfrac{2\pi\gamma}{N}\dfrac{n(n+1+2q)}{2}\right), & N \text{ odd}, \end{cases} \tag{1}$$

for $n = 0, 1, \cdots, N-1$, where $j = \sqrt{-1}$, N is the length (or period) of the sequence, q is an arbitrary integer, and γ is a positive integer (called the *index*) relatively prime to N. The number of available ZC sequences of length N (or the number of possible values of γ) equals the number of integers that are relatively prime to N, which is given by Euler's totient function $\phi(N)$. When N is an odd prime, the maximum number of distinct ZC sequences of length N exists, which is given by $\phi(N) = N-1$. Hence typically N is chosen to be an odd prime, which we assume in the remainder of the paper.

In some applications, however, the required length of a ZC sequence is a composite integer M, and hence prime-length ZC sequences cannot be directly used. Instead, such composite-length ZC sequences are derived from prime-length ZC sequences. Two methods are typically employed: 1) truncation, or 2) cyclic extension. In truncation, a ZC sequence of length N, where N is the *smallest* prime number $\geq M$, is truncated to M. In

cyclic extension, a ZC sequence of length N, where N is the *largest* prime $\leq M$, is cyclically extended to M.

2.1 Correlation Properties of ZC Sequences

Zadoff–Chu sequences possess good correlation properties which are essential in several engineering applications such as establishing timing synchronization between a mobile terminal and a base station, performing channel estimation, and reducing peak-to-average power ratio. The periodic auto-correlation function of an arbitrary sequence $z[n]$ of length N is defined as $R_{zz}[\tau] = \sum_{n=0}^{N-1} z[n] \cdot z^*[(n+\tau) \bmod N]$, where τ is an integer, and * is complex conjugation. It is easy to show that for a ZC sequence $z_\gamma[n]$, $R_{z_\gamma z_\gamma}[0] = N$ and $R_{z_\gamma z_\gamma}[\tau] = 0$ if $\tau \bmod N \neq 0$. Since the out-of-phase value of $R_{z_\gamma z_\gamma}[\tau]$ is zero, ZC sequences are called perfect sequences [6, 7].

In addition, the periodic cross-correlation function of two sequences x and y of length N is defined as $R_{xy}[\tau] = \sum_{n=0}^{N-1} x[n] \cdot y^*[(n+\tau) \bmod N]$. Let $\Gamma = \{z_\gamma[n] \mid 1 \leq \gamma \leq N-1\}$ denote the set of ZC sequences of length N. In [6] it was shown that $|R_{z_{\gamma_i} z_{\gamma_j}}[\tau]| = \sqrt{N}$ for any integer τ and any distinct pair of sequences $z_{\gamma_i}, z_{\gamma_j} \in \Gamma$.

The correlation properties of a ZC sequence remain invariant under cyclic shifts, addition of a constant to the phases in the exponentials in Eq. 1, conjugation of the entire sequence, or addition of a linear phase shift of the form $\exp\left(j\frac{2\pi qn}{N}\right)$ for any integer q [3]. Hence, q in Eq. 1 is typically set to 0 without any loss of generality, which we assume in the remainder of this paper. Finally, ZC sequences obviously have constant amplitude (and hence called CAZAC sequences), which is inline with the basic characteristics of 3G transmission schemes. For composite-length ZC sequences derived from prime-length sequences, the CAZAC property however is degraded to some extent [8].

2.2 ZC Sequences in Frequency Domain

In [6] it was observed that there exists a duality between ZC sequences in time domain and ZC sequences in frequency domain. Let Z_γ be the N-point discrete Fourier transform (DFT) of the time-domain ZC sequence z_γ:

$$Z_\gamma[k] = \sum_{n=0}^{N-1} z_\gamma[n] \exp\left(-j\frac{2\pi nk}{N}\right), \quad k = 0, \cdots, N-1. \tag{2}$$

Note that both z_γ and Z_γ are periodic sequences of period N. The following theorem shows that there exists a simple relationship between $z_\gamma[n]$ and $Z_\gamma[k]$ [6]. A similar result is derived independently in the recent references [9, 10].

Theorem 1 $Z_\gamma[k] = Z_\gamma[0] \cdot z_\gamma^*[\gamma' k]$, $k = 0, 1, \cdots, N-1$, *where* γ' *is defined such that* $\gamma'\gamma \bmod N = 1$.

A slightly modified proof of the theorem than the one in [6] is included in the Appendix. Hence, a ZC sequence can be directly generated in the frequency domain without the need for a DFT operation.

3 An Optimized Algorithm for Computing ZC Sequences

In this section, we present an optimized algorithm for computing ZC sequences both in time domain (TD) and frequency domain (FD) using the CORDIC algorithm [5]. From Theorem 1 it is obvious that both TD-ZC sequence $z_\gamma[n]$ and FD-ZC sequence $Z_\gamma[n]$ are related as $Z_\gamma[n] = Z_\gamma[0] \cdot z_\gamma^*[\gamma' n]$. Hence, in frequency domain we effectively compute the same ZC sequence $z_\gamma[n]$ but with its elements conjugated and scaled by the constant $Z_\gamma[0]$, and reordered according to the map $n \mapsto \gamma' n \bmod N$, where $\gamma\gamma' \bmod N = 1$.

3.1 The CORDIC Algorithm

From Eq. 1, we can evaluate a ZC sequence using trigonometric functions as $z_\gamma[n] = \cos\left(\frac{2\pi\gamma}{N}\frac{n(n+1)}{2}\right) - j\sin\left(\frac{2\pi\gamma}{N}\frac{n(n+1)}{2}\right)$. The CORDIC algorithm is a hardware-efficient iterative algorithm for evaluating trigonometric and other transcendental functions using only shift and add operations. The algorithm, credited to Volder [5], is derived from the general Givens rotation transform, and can perform the rotation of a two-dimensional vector (x, y) in linear, circular and hyperbolic coordinates. CORDIC has a wide range of applications in signal processing and matrix operations, such as matrix decomposition [11, 12], three dimensional rotations [13], discrete cosine transform [14], speech processing, antenna array processing, computer graphics [15], among others.

In this work, we consider CORDIC in rotation mode and restrict the rotation angle to be within the range $|\theta| \leq \pi/2$. Other rotation angles outside this range can be easily converted to be within this range. The circular

CORDIC rotation with accuracy of B fractional bits is expressed as [5]:

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i} \tag{3}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i} \tag{4}$$

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i}) \tag{5}$$

for $i = 1, \cdots, B$, where x_i, y_i are the vector coordinates and z_i is the residual angle relative to the x-axis at the ith iteration. The decision variable d_i is updated based on the sign of the residual angle z_{i+1} in Eq. 5. If the $z_{i+1} < 0$, then $d_{i+1} = -1$ which corresponds to adding the micro-rotation angle $\tan^{-1}(2^{-(i+1)})$ at iteration $i+1$. Otherwise, $d_{i+1} = +1$ and the micro-rotation angle is subtracted.

If the initial conditions for the three recursions (Eqs. 3–5) are chosen as $x_1 = K$, $y_1 = 0$, $z_1 = \theta$, with $|\theta| \le \pi/2$ and $K = \prod_{i=1}^{B} \sqrt{(1 + 2^{-2i})}$ [5], then the final outputs after B iterations are the cosine and sine functions $x_{B+1} = \cos(\theta)$ and $y_{B+1} = \sin(\theta)$. The scaling constant K is fixed and can be pre-computed off-line for a given B. The arc-tangent values for $2^{-i}, i = 1, \cdots, B$, are stored in a look-up table.

3.2 Optimized ZC-Algorithm

The CORDIC algorithm is employed to evaluate the sine and cosine functions of the exponential in Eq. 1. The argument of these trigonometric functions however needs to be computed first. For $z_\gamma[n]$ in time domain the argument is $\frac{2\pi}{N}\frac{\gamma n(n+1)}{2}$, and for $z^*_\gamma[\gamma' n]$ in frequency domain the argument is $\frac{2\pi}{N}\frac{\gamma\gamma' n(\gamma' n+1)}{2}$. To avoid the use of multipliers with non-constant terms in evaluating the arguments for a given n, we compute the arguments recursively as we traverse the elements of the ZC sequence. That is, we compute $\arg(z_\gamma[n])$ for n using $\arg(z_\gamma[n-1])$ at $n-1$, and we compute $\arg(z^*_\gamma[\gamma' n])$ for n using $\arg(z^*_\gamma[\gamma'(n-1)])$ for $n-1$. To handle both cases, define

$$\theta_m[n] = \frac{2\pi}{N}\frac{\gamma mn(mn+1)}{2},$$

$$n = 0, 1, \cdots, N-1;\ m = 1 \text{ or } \gamma';\quad |\theta_m[n]| < \pi/2,$$

to be the argument of a ZC sequence element. Hence $\arg(z_\gamma[n]) = \theta_1[n]$ with $m = 1$ for the time-domain sequence, and $\arg(z^*_\gamma[\gamma' n]) = \theta_{\gamma'}[n]$ with $m = \gamma'$ for the frequency-domain sequence.

Next we consider evaluating $\theta_m[n]$ recursively for $n = 0, \cdots, N-1$, by applying a sequence of transformations. Let $\alpha_m[n] = \gamma\frac{mn(mn+1)}{2} \bmod N$, for $m = 1$ or γ', and hence $\theta_m[n] = \frac{2\pi}{N}\alpha_m[n]$. Obviously, $\alpha_m[n]$ is quadratic in n. The following lemma allows us to express $\alpha_m[n]$ recursively as a first order difference equation that is linear in n:

Lemma 1

$$\alpha_m[n] = \begin{cases} 0, & n = 0; \\ \left(\alpha_m[n-1] + \gamma m^2 n - \gamma\dfrac{m(m-1)}{2}\right) \bmod N, & n \ge 1. \end{cases} \tag{6}$$

The proof follows by mathematical induction, and hence it has been omitted. The terms γm^2 and $\gamma\frac{m(m-1)}{2}$ in Eq. 6 are constants for a given γ, and hence can be computed off-line. The multiplier with n needed to compute the product $(\gamma m^2)\cdot n$ can be eliminated by defining $\beta_m[n] = \left(\gamma m^2 n - \gamma\frac{m(m-1)}{2}\right) \bmod N$. Again, by mathematical induction, it is easy to proof that $\beta_m[n]$ can be computed recursively as

$$\beta_m[n] = \begin{cases} -\gamma\dfrac{m(m-1)}{2} \bmod N, & n = 0; \\ \left(\beta_m[n-1] + \gamma m^2\right) \bmod N, & n \ge 1. \end{cases} \tag{7}$$

Substituting Eq. 7 in Eq. 6, $\alpha_m[n]$ becomes

$$\alpha_m[n] = \begin{cases} 0, & n = 0; \\ (\alpha_m[n-1] + \beta_m[n]) \bmod N, & n \ge 1, \end{cases} \tag{8}$$

Note that the use of the modulo operator in computing $\beta_m[n]$ in Eq. 7 in addition to Eq. 6 simplifies the implementation of the modulo operator into a subtractor (without the need of an integer divider). In Eq. 7, we have $0 \le \beta_m[n-1] + \gamma m^2 < 2N$, hence

$$(\alpha_m[n-1] + \beta_m[n]) \bmod N = \begin{cases} \alpha_m[n-1]+\beta_m[n], & \text{if } \alpha_m[n-1]+\beta_m[n] < N; \\ \alpha_m[n-1]+\beta_m[n] - N, & \text{otherwise.} \end{cases}$$

A similar argument applies to the modulo operator in Eq. 6. Therefore,

$$\theta_m[n] = \frac{2\pi}{N}\alpha_m[n]. \tag{9}$$

This $\theta_m[n]$ must be translated into the range $[-\pi/2, \pi/2]$, or equivalently $\alpha_m[n]$ translated into the range $[-N/4, N/4]$ with appropriate sign adjustments, before applying it to the CORDIC (Eqs. 3–5).

The algorithm for computing the elements of a ZC sequence recursively in time domain ($m = 1$) and frequency domain ($m = \gamma'$) using the previous scheme is shown in the pseudo-code in **Algorithm 1**. Note that the modulo operations in the algorithm can be easily implemented using only one subtraction operation (with N) for every n.

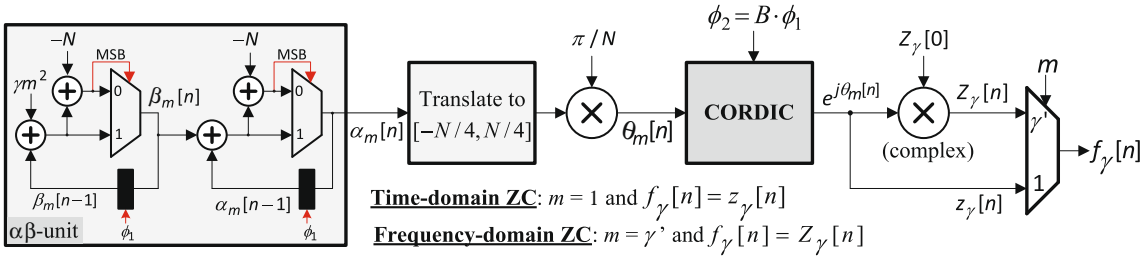

Figure 1 Optimized architecture for ZC sequence computation.

Figure 1 shows the optimized architecture for computing ZC sequences based on **Algorithm 1**. The $\alpha\beta$ unit computes the argument $\alpha_m[n]$ as described earlier. It generates a value every $1/\phi_1$ clock cycles. The CORDIC block implements B iterations of (Eqs. 3–5) either in parallel by hardware replication, or sequentially by reusing the hardware for one iteration B times. In the later case, the clock frequency ϕ_2 should set to $B\phi_1$ in order to keep up with the speed of the $\alpha\beta$ unit.

Algorithm 1 Optimized Zadoff-Chu sequence computation algorithm using CORDIC.

procedure $f_\gamma = \text{ZADOFF-CHU}(N, \gamma, Z_\gamma[0], B, K; m)$
  $\alpha_m[0] \leftarrow 0, \beta_m[0] \leftarrow -\gamma\frac{m(m-1)}{2} \bmod N$
  for $n = 1$ to $N-1$ **do**
    $\beta_m[n] \leftarrow (\beta_m[n-1] + \gamma m^2) \bmod N$ ▷ **Note that γm^2 is a constant.**
    $\alpha_m[n] \leftarrow (\alpha_m[n-1] + \beta_m[n]) \bmod N$
    if $|\alpha_m[n]| > N/2$ **then** ▷ **Translate $\alpha_m[n]$ to $[-N/2, N/2]$**
      $\alpha_m[n] \leftarrow \alpha_m[n] - \mathbf{sgn}(\alpha_m[n]) \cdot N$
    end if

    $s \leftarrow 1$ ▷ **Variable for sign adjustment**
    $r \leftarrow 2\alpha_m[n]$
    if $|\alpha_m[n]| > N/4$ **then** ▷ **Translate $2\pi\alpha_m[n]/N$ to $[-\pi/2, \pi/2]$**
      $r \leftarrow r - \mathbf{sgn}(\alpha_m[n]) \cdot N$
      $s \leftarrow -1$ ▷ **Results from CORDIC must be negated**
    end if
    $\theta_m[n] \leftarrow \frac{\pi}{N} \cdot r$ ▷ $\theta_m[n] \in [-\pi/2, \pi/2]$

    $x[1] \leftarrow K,\ y[1] \leftarrow 0,\ z[1] \leftarrow \theta_m[n]$
    for $i = 1$ **to** B **do** ▷ **CORDIC iterations**
      $x[i+1] \leftarrow x[i] - \mathbf{sgn}(z[i]) \cdot (y[i] \gg i)$ ▷ **Computes** $\cos(\theta_m[n])$
      $y[i+1] \leftarrow y[i] + \mathbf{sgn}(z[i]) \cdot (x[i] \gg i)$ ▷ **Computes** $\sin(\theta_m[n])$
      $z[i+1] \leftarrow z[i] - \mathbf{sgn}(z[i]) \cdot \arctan(2^{-i})$ ▷ **Computes residual angle**
    end for

    if $m = 1$ **then**
      $f_\gamma[n] \leftarrow s \cdot (x[B+1] + jy[B+1])$ ▷ **Time domain:** $\cos(\theta_m[n]) + j\sin(\theta_m[n])$
    else
      $f_\gamma[n] \leftarrow s \cdot (x[B+1] - jy[B+1]) \times Z_\gamma[0]$ ▷ **Freq. domain:** $(\cos(\theta_m[n]) - j\sin(\theta_m[n])) \times Z_\gamma[0]$
    end if
  end for
end procedure

4 Application to Random Access Channel in LTE

4.1 PRACH Preamble

Prior to transmission of data, a mobile terminal or user equipment (UE) needs to establish connectivity to the network through a process called cell-search. As a result, the UE obtains the identity of the cell and estimates the frame timing of the identified cell. The UE can then request a connection setup by undertaking a *random access* process to establish uplink synchronization and obtain a unique identity. The random access procedure consists of transmission of a random-access preamble by the UE, allowing the base station (eNodeB) to estimate the transmission timing of the terminal. The eNodeB transmits, in response, a timing advance command to adjust the terminal transmit timing based on the measurement in the previous step. This mechanism is handled by the physical layer (PHY) in LTE through the physical random-access channel (PRACH) [8].

The PRACH preamble consists of three parts: a cyclic prefix, a sequence, and a guard interval, of durations T_{CP}, T_{SEQ}, T_{G}, respectively (see Fig. 2). There are various preamble formats in LTE consisting of different T_{CP}, T_{SEQ}, and T_{G} lengths to support different cell size requirements. Figure 2 illustrates preamble format-0 for small-medium cells. The sequence part of the preamble is constructed using ZC sequences with zero correlation zone, generated from one or several root ZC sequences. The network configures the set of preamble sequences the UE is allowed to use, and the preamble transmitted by the UE is randomly chosen from this pool. Extensions to perform detection in the presence of longer propagation delays has been recently considered in [16].

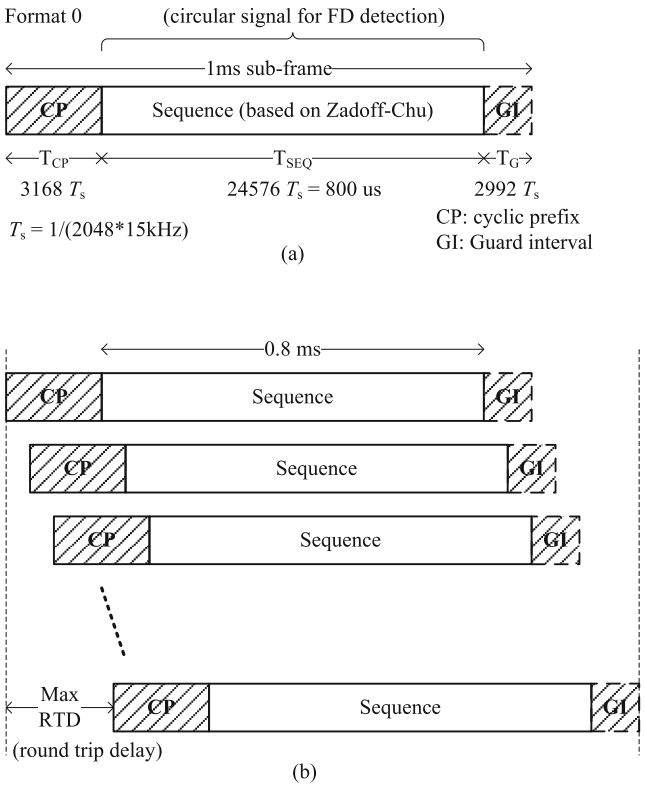

Figure 2 **a** Random access preamble format. **b** Circular PRACH signal (format 0). The staggered timing offsets in the figure correspond to various round-trip delays between the mobile and the base station.

There are 64 preambles available in each cell. This set of preamble sequences is found from ZC sequences with pre-configured roots and all their cyclic shifts [2]. Let Γ denote the set of distinct ZC roots needed to generate these 64 preambles, and $z_\gamma[n]$ be a root ZC sequence with index $\gamma \in \Gamma$ and length $N = 839$. From $z_\gamma[n]$, random access preambles $p_{\gamma,\nu}[n]$ with zero correlation zones of some length $N_{\mathrm{CS}} - 1$ are defined by cyclically shifting $z_\gamma[n]$ by multiples of N_{CS} according to $p_{\gamma,\nu}[n] = z_\gamma[(n + \nu N_{\mathrm{CS}}) \bmod N]$, $\gamma \in \Gamma,\ \nu = 0, 1, \cdots, \lfloor N/N_{\mathrm{CS}} \rfloor - 1$. Different values of N_{CS} are defined for various cell sizes and Doppler shifts. The second consecutive root γ is picked from Γ and the procedure is repeated until all 64 preambles are generated.

4.2 Preamble Detection and Delay Estimation

At the receiver side, the eNodeB performs a circular cross-correlation between the extracted preamble and the pool of preambles allocated to an eNodeB, as illustrated in Fig. 3. Since multiple preambles are generated from a single ZC root sequence, the cross-correlation is performed directly with the ZC root sequence as

$$y_{r,\gamma}[n] = \sum_{m=0}^{N-1} x_r[m]\, z_\gamma^*[(n-m) \bmod N],$$
$$n = 0, 1, \cdots, N-1,\ r = 1, 2, \cdots, N_r,\ \gamma \in \Gamma, \qquad (10)$$

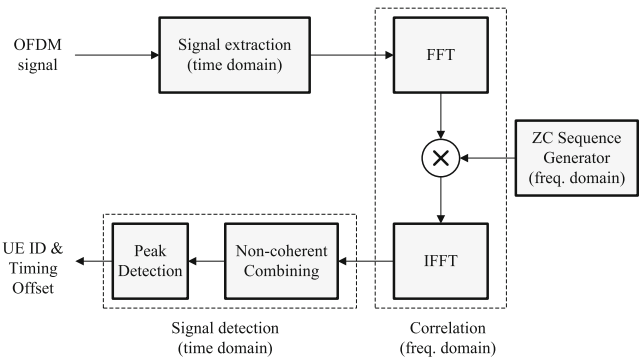

Figure 3 Block diagram for an eNodeB LTE searcher.

where $x_r[m]$ is the received signal from the rth antenna (in a system with N_r antennas), and $z_\gamma[m]$ is the γth root ZC sequence. In vector form, the cross-correlations are expressed as $\underline{y_{r,\gamma}} = \left[y_{r,\gamma}[0], y_{r,\gamma}[1], \cdots, y_{r,\gamma}[N-1]\right]^T$, where $r = 1, 2, \cdots, N_r$. These cross-correlations can be computed efficiently in the frequency domain as $\underline{y_{r,\gamma}} = \text{IFFT}\left[X_r[l] \cdot W_\gamma[l]\right]$, where $X_r[l]$ is the FFT of $\underline{x_r}[m]$, $Z_\gamma[l]$ is the FFT of $z_\gamma[m]$, and $W_\gamma[l] = Z^*_\gamma[N-1-l]$ flips $Z^*{}_\gamma$ to account for the circular shifting of z_γ in Eq. 10. An inverse FFT is performed on the product to provide a vector of cross correlations in time domain. The cross-correlations from the N_r antennas are then non-coherently combined to yield the following vector of correlations: $\underline{y_\gamma} = \left[y_\gamma[0], y_\gamma[1], \cdots, y_\gamma[N-1]\right]^T$, where $\gamma \in \Gamma$.

Let γ' and n' denote respectively the ZC root and the sample number such that $y_{\gamma'}[n']$ is maximum. Let i' be the index of γ' in Γ, where $0 \le i' < |\Gamma|$. A preamble is detected by comparing with a pre-determined threshold that minimizes the probability of a false alarm rate. The ID of the detected preamble is $I = \lfloor N/N_{\text{CS}} \rfloor \times i' + \lfloor n'/N_{\text{CS}} \rfloor$, and the delay is $D = (n' \bmod N_{\text{CS}}) \times T_s$, where T_s is the sampling period.

4.3 Other Applications

ZC sequences are also applied in LTE to perform downlink symbol timing and carrier frequency synchronization. The synchronization procedure is based on reference signals constructed using ZC sequences that are broadcast by the eNodeB [2, 8]. The UE performs cross-correlations against these reference signals similar to the preamble detection procedure discussed earlier in order to compute the time and frequency offsets. Since the common operation among all of these applications that use ZC sequence is cross-correlation followed by max-selection and comparison to a threshold, the accuracy depends on the bit-precision with which ZC sequence elements are represented as well as the value of the threshold used. As demonstrated in the following section for the case of preamble detection, the cross-correlation operation is quite resilient to quantization errors in representing ZC sequences.

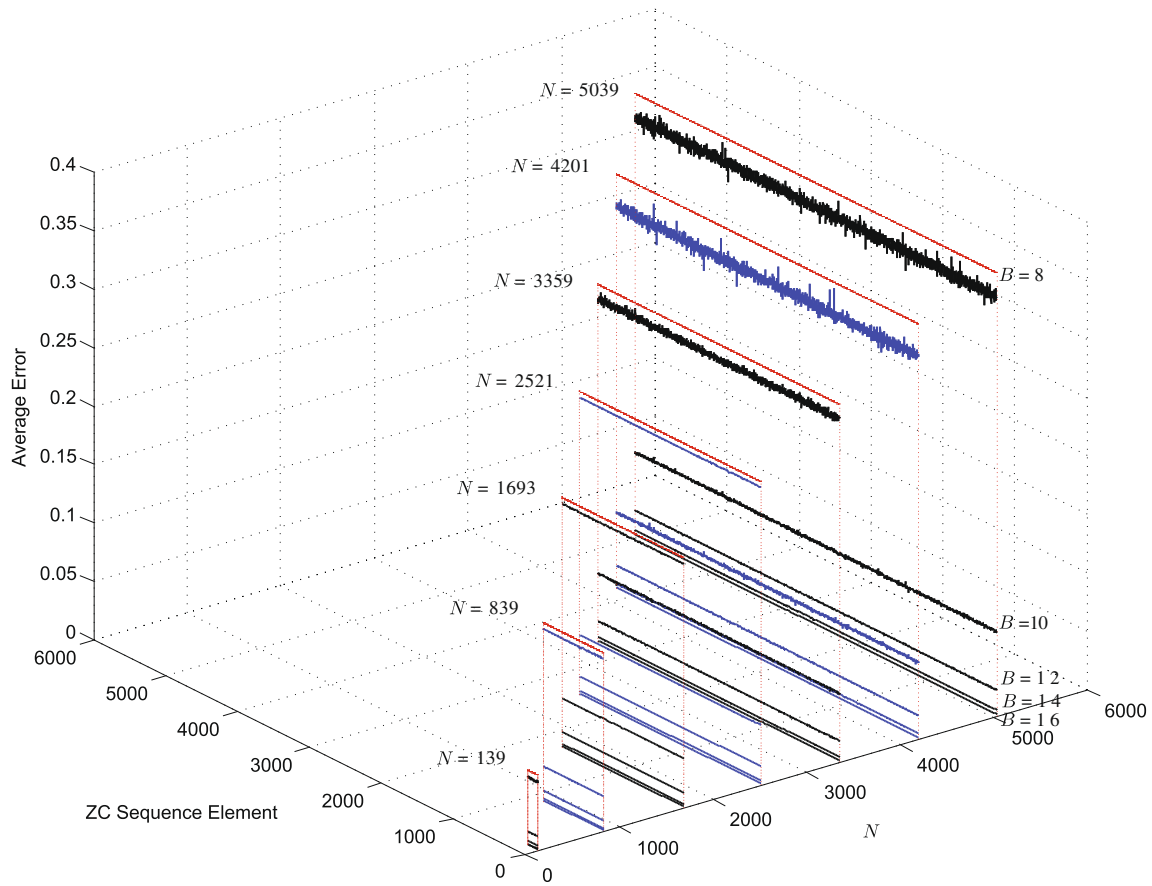

Figure 4 Average error between ideal ZC sequence elements and ZC sequence elements generated using the proposed architecture as a function of N and ZC element, for various values of B.

5 Simulation Results

The searcher block employing the proposed architecture for computing ZC sequences using the CORDIC algorithm was implemented in VHDL and then synthesized on an FPGA. A corresponding bit-accurate C model using fixed-point arithmetic was developed for verification. The architecture employs a parallel reconfigurable CORDIC engine for the required input bit precision B. The internal datapath bit-width of the overall ZC architecture is programmable to facilitate fixed-point analysis. Extra guard bits are added internally in the datapath to avoid error propagation depending on the number of CORDIC iterations. The programmable parameters are the input precision B and the number of iterations I. For a given pair (B, I), a parallel CORDIC engine is automatically synthesized using I hardware copies with internal bit-precision of $B + \lceil \log_2(I) \rceil$ to implement the iterations (Eqs. 3–5) in parallel, together with the required look-up tables (LUTs) and scaling constants $K = \prod_{i=1}^{B} \sqrt{(1 + 2^{-2i})}$. The architecture computes ZC elements both in time domain and frequency domain.

Figure 4 plots the average error between ZC elements computed with floating point precision and ZC elements generated using the architecture as a function of length N and ZC element, for various values of B, with $I = B$. The error is computed as the pointwise absolute difference in magnitude between the floating point and fixed point implementation, averaged out across all the elements in the ZC sequence. LTE Format-0 preamble sequences of length $N = 839$ is also considered. First, for a given N and B, the average error is uniform across all ZC sequence elements. As N increases, the average error increases for a given B because more precision is needed to represent $\theta_m[n]$. As B increases, the average error drops. For the ZC application considered, an input bit-precision of $B = 8$ results in an acceptable error. The plots demonstrate the high accuracy of the proposed

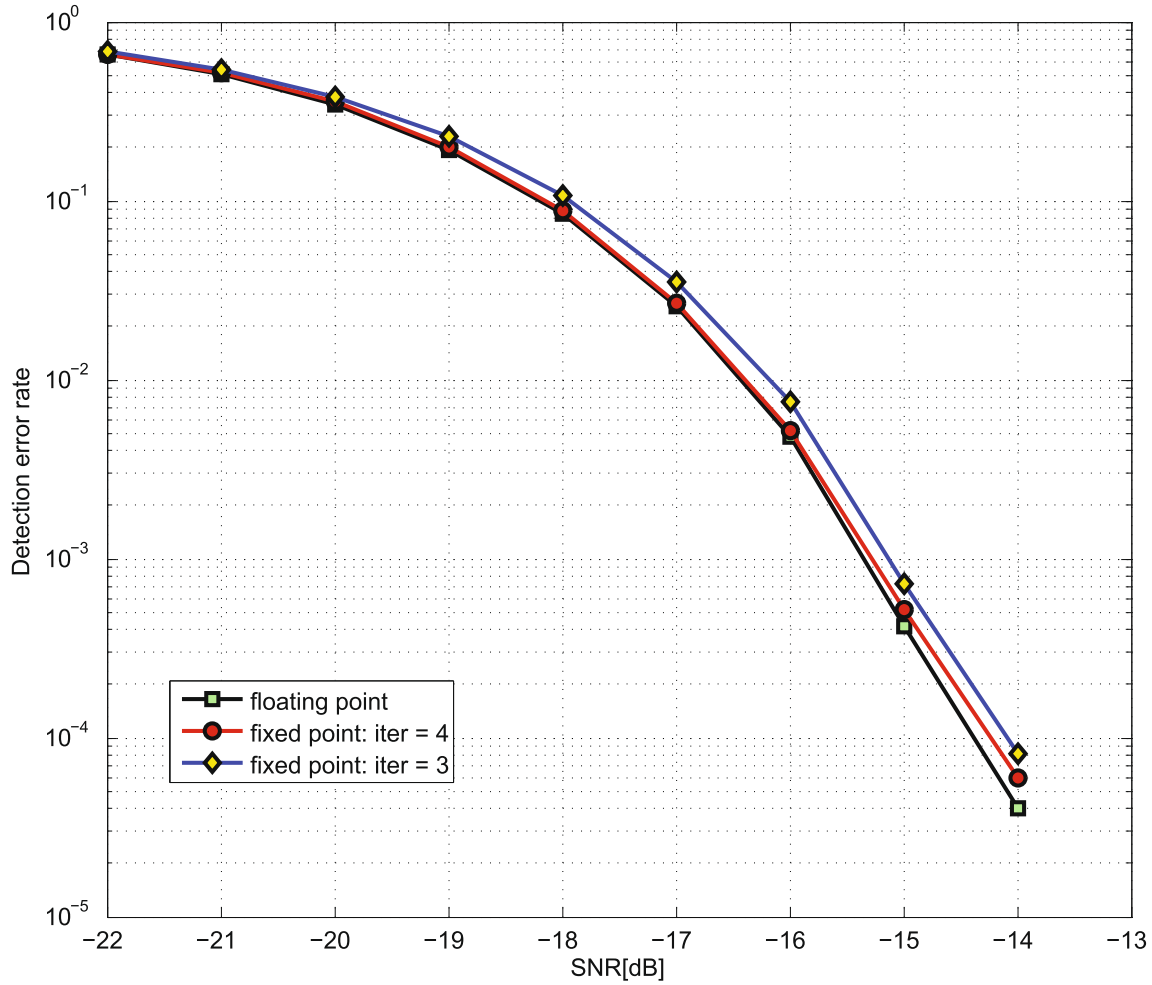

Figure 5 Detection error rate versus SNR in dB.

algorithm and corresponding architecture in computing ZC sequences.

The processing time required for an N-element ZC sequence is NBT seconds using the serial CORDIC, and NT using the parallel CORDIC implementation, where T is the clock period.

The LTE eNodeB searcher block shown in Fig. 3 was implemented in VHDL using the proposed ZC algorithm for computing ZC sequence elements in the frequency domain. The block was then synthesized part of a larger system on a Xilinx Virtex-5 FPGA. Format-0 preambles with $N_{CS} = 15$ corrupted with random noise and random shifts were fed into the searcher block. A pool of 64 preambles with corresponding logical root indices were allocated to the searcher. By performing correlations in the frequency domain between the noisy preambles and the corresponding 64 preambles generated from ZC sequences using the proposed architecture and then comparing to a threshold, the searcher is capable of detecting a random access attempt by a UE and estimating its round trip delay. The input datapath bit-width of the architecture was set to $B = 8$ bits. The CORDIC engine within the ZC block was configured to run I iterations. Figure 5 plots the detection error rate versus signal-to-noise ratio in dB for ideal detection using floating point precision, and detection using the proposed architecture with the CORDIC engine running 3 iterations and 4 iterations, respectively, in computing ZC sequence elements. Using only 4 iterations, the detection error rate of the proposed architecture almost matches the ideal rate.

Table 1 summarizes the resources utilized by the searcher block when synthesized on a Virtex-5 FPGA, for $B = 8$, $I = 3, \cdots, 8$. In all cases, 5 DSP multipliers were utilized as well. As shown in the table, the resources increase moderately as the number of CORDIC pipeline stages (iterations I) is increased. The efficiency of the CORDIC optimizations performed in Section 3 is reflected in synthesis which shows a maximum percentage resource utilization of only 3 %. The synthesized searcher block can run up to a clock frequency of 500 MHz.

Table 1 Searcher block resource utilization on a Xilinx Virtex-5 FPGA, for $B = 8$ and $I = 3, \cdots, 8$.

Iterations I	Slice registers	Slice LUTs	Bit slices	Adders
3	411 (1 %)	755 (2 %)	896	23
4	465 (1 %)	807 (2 %)	952	26
5	519 (1 %)	859 (2 %)	1006	29
6	573 (1 %)	911 (2 %)	1060	32
7	627 (1 %)	963 (2 %)	1114	35
8	681 (2 %)	1015 (3 %)	1169	38

Note: 248 (1 %) out of 12480 memory slice LUTs are used as RAM

Finally, a significant advantage of the proposed architecture is the memory savings attained compared to the case where the ZC sequences are pre-computed and stored. In a practical setting with a 3-sector cell, where each sector is allocated 64 preambles of length 839, the total memory required to store these complex-valued sequences on-chip assuming 8-bit representation is around 2.5 M-bits. Being able to compute these sequences on the fly with high accuracy using a low-complexity architecture eliminates the need for such memory, saving on valuable chip area and reducing power consumption.

6 Conclusion

A hardware-efficient algorithm for computing ZC sequences with high-accuracy in real-time has been presented. The algorithm has been optimized to eliminate the need for memory as well as multipliers with non-constant terms. A corresponding hardware architecture has been developed and employed in an LTE searcher block for PRACH preamble detection. Simulation results have demonstrated the efficiency and accuracy of the proposed algorithm. The proposed work is valuable in other applications that use ZC sequences such as channel estimation and frequency/time tracking.

Appendix

Proof of Theorem 1 The DFT of the ZC sequence in Eq. 1 is given by $Z_\gamma[k] = \sum_{n=0}^{N-1} \exp\left(-j\frac{2\pi\gamma}{N}\frac{n(n+1)}{2}\right) \times \exp\left(-j\frac{2\pi kn}{N}\right)$. Since $\gamma'\gamma \bmod N = 1$, we can replace k by $\gamma\gamma' k$ in the last exponential and obtain $Z_\gamma[k] = \sum_{n=0}^{N-1} \exp\left(-j\frac{2\pi\gamma}{N}\frac{n(n+1)}{2}\right) \times \exp\left(-j\frac{2\pi\gamma\gamma' kn}{N}\right)$. Combining the arguments of the exponentials and completing the squares with respect to $g = (n + \gamma' k)$, we obtain:

$$
\begin{aligned}
Z_\gamma[k] &= \sum_{n=0}^{N-1} \exp\left(-j\frac{2\pi\gamma}{N}\frac{g^2 + g - (\gamma' k)^2 - (\gamma' k)}{2}\right) \\
&= \exp\left(j\frac{2\pi\gamma}{N}\frac{(\gamma' k)^2 + (\gamma' k)}{2}\right) \\
&\quad \times \sum_{n=0}^{N-1} \exp\left(-j\frac{2\pi\gamma}{N}\frac{g^2 + g}{2}\right) \\
&= z_\gamma^*[\gamma' k] \cdot \sum_{n=0}^{N-1} \exp\left(-j\frac{2\pi\gamma}{N}\frac{n^2 + n}{2}\right) \\
&= z_\gamma^*[\gamma' k] \cdot Z_\gamma[0]. \qquad \square
\end{aligned}
$$

References

1. Mansour, M.M. (2009). Optimized architecture for computing Zadoff–Chu sequences with application to LTE. In *Proc. IEEE Int. global communications conference, Honolulu, Hawaii, USA* (pp. 1–6).
2. 3GPP TS 36.211 V8.2.0 (2008). 3rd Generation Partnership Project; Technical specification group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 8).
3. Chu, D. (1972). Polyphase codes with good periodic correlation properties. *IEEE Transactions on Information Theory, IT-18*, 531–532.
4. Kantabutra, V. (1996). On hardware for computing exponential and trigonometric functions. *IEEE Transactions on Computers, 45*(3), 328–339.
5. Volder, J.E. (1959). The CORDIC trigonometric computing technique. *IRE Transactions on Electronic Computers, 8*, 330–334.
6. Sarwate, D.V. (1979). Bounds on crosscorrelation and autocorrelation of sequences. *IEEE Transactions on Information Theory, IT-25*, 720–724.
7. Li, C.-P. & Huang, W.-C. (2007). A constructive representation for the Fourier dual of the Zadoff–Chu sequences. *IEEE Transactions on Information Theory, 53*(11), 4221–4224.
8. Dahlman, E., et al. (2007). *3G evolution: HSPA and LTE for mobile broadband*. London, UK: Academic Press.
9. Beyme, S., & Leung, C. (2009). Efficient computation of DFT of Zadoff-chu sequences. *Electronics Letters, 45*(9), 461–463.
10. Popovic, B.M. (2010). Efficient DFT of Zadoff–Chu sequences. *Electronics Letters, 46*(7), 502–503.
11. Ercegovac, M., & Lang, T. (1990). Redundant and on-line CORDIC: application to matrix triangularization and SVD. *IEEE Transactions on Computers, 6*, 725–740.
12. Hu, X., Bass, S.C., Harber, R.G. (1993). An efficient implementation of singular value decomposition rotation transformations with CORDIC processor. *Journal of Parallel and Distributed Computing, 17*, 360–362.
13. Lang, T., & Antelo, E. (2001). High-throughput 3D rotations and normalizations. In *Proc. of Asilomar conf. signals, systems, computers* (Vol. 1, pp. 846–851).
14. Hu, Y.H., & Wu, Z. (1995). An efficient CORDIC array structure for the implementation of discrete cosine transform. *IEEE Transactions on Signal Processing, 43*, 331–336.
15. de Lange, A.A.J., et al. (1990). Real time applications of the floating point pipeline CORDIC processor in massive-parallel pipelined DSP algorithms. In *Proc. int. conf. acoustics, speech, sig. proc.* (Vol. 2, pp. 1013–1016).
16. Kim, S., Joo, K., Lim, Y. (2012). A delay-robust random access preamble detection algorithm for LTE system. In *Radio and wireless symposium (RWS), 2012 IEEE* (pp. 75–78).

Mohammad M. Mansour received his B.E. degree with distinction in 1996 and his M.E. degree in 1998 both in computer and communications engineering from the American University of Beirut (AUB), Beirut, Lebanon. In August 2002, Mohammad received his M.S. degree in mathematics from the University of Illinois at Urbana-Champaign (UIUC), Urbana, Illinois, USA. Mohammad also received his Ph.D. in electrical engineering in May 2003 from UIUC.

He is currently an associate professor of electrical and computer engineering with the ECE Department at AUB, Beirut, Lebanon. From December 2006 to August 2008, he was on research leave with QUALCOMM Flarion Technologies in Bridgewater, New Jersey, USA, where he worked on modem design and implementation for 3GPP-LTE, 3GPP-UMB, and peer-to-peer wireless networking PHY layer standards. From 1998 to 2003, he was a research assistant at the Coordinated Science Laboratory (CSL) at UIUC. During the summer of 2000, he worked at National Semiconductor Corp., San Francisco, CA, with the wireless research group. In 1997 he was a research assistant at the ECE department at AUB, and in 1996 he was a teaching assistant at the same department. His research interests are VLSI design and implementation for embedded signal processing and wireless communications systems, coding theory and its applications, digital signal processing systems and general purpose computing systems.

Prof. Mansour is a member of the Design and Implementation of Signal Processing Systems Technical Committee of the IEEE Signal Processing Society, and a Senior Member of the IEEE. He has been serving as an associate editor for IEEE Transactions on Circuits and Systems II since April 2008, associate editor for IEEE Transactions on VLSI Systems since January 2011, and associate editor for IEEE Signal Processing Letters since January 2012. He served as the technical co-chair of the IEEE Workshop on Signal Processing Systems (SiPS 2011), and as a member of the technical program committee of various international conferences. He is the recipient of the PHI Kappa PHI Honor Society Award twice in 2000 and 2001, and the recipient of the Hewlett Foundation Fellowship Award in March 2006. He joined the faculty at AUB in October 2003.