Journal of VLSI Signal Processing 30, 21–33, 2002

VLSI Implementation of the Multistage Detector for Next Generation Wideband CDMA Receivers

GANG XU, SRIDHAR RAJAGOPAL, JOSEPH R. CAVALLARO AND BEHNAAM AAZHANG
Department of Electrical and Computer Engineering, Rice University, 6100 Main St., Houston TX 77005, USA

Received September 6, 2000; Revised June 15, 2001

Abstract. The multistage detection algorithm has been proposed as an effective interference cancellation scheme for next generation Wideband Code Division Multiple Access (W-CDMA) base stations. In this paper, we propose a real-time VLSI implementation of this detection algorithm in the uplink system, where we have achieved both high performance in interference cancellation and computational efficiency. When interference cancellation converges, the difference of the detection vectors between two consecutive stages is mostly zero. Under the assumption of BPSK modulation, the differences between the bit estimates from consecutive stages are 0 and ± 2. Bypassing the zero terms saves computations. Multiplication by ± 2 can be easily implemented in hardware as arithmetic shifts. However, the convergence of the algorithm is dependent on the number of users, the interference and the signal to noise ratio and hence, the detection has a variable execution time. By using just two stages of the differencing detector, we achieve predictable execution time with performance equivalent to at least eight stages of the regular multistage detector. A VLSI implementation of the differencing multistage detector is built to demonstrate the computational savings and the real-time performance potential. The detector, handling up to eight users with 12-bit fixed point precision, was fabricated using a 1.2 μm CMOS technology and can process 190 Kbps/user for 8 users.

Keywords: CDMA, multiuser detection, multistage detector, interference cancellation, real-time implementation, fixed-point

1. Introduction

The fast growing cellular telephony industry provides higher capacities for larger number of subscribers each year, which in turn requires complex signal processing techniques and sophisticated multiple access methods to meet these demands. Direct Sequence Code Division Multiple Access (DS-CDMA) has been recognized as one of the best multiple access schemes for wireless communication systems [1]. The Wideband CDMA system discussed [2] in this paper is based on the short code DS-CDMA scheme. We place particular emphasis on the uplink (from mobiles to the base station) system, where all the subscribers share the common channel (shown in Fig. 1). In such an environment, the only way to distinguish these users is to use orthogonal or nearly orthogonal codes (spreading sequences) to modulate the transmitted bits.

Any single desired user in the CDMA uplink system experiences direct interference from the other users in the same cell and neighboring cells. This effect is called Multiple Access Interference (MAI), which is the major limitation in capacity for the current IS-95 CDMA standard. The other related problem is called the near-far problem. When a user is far from the base station, it is likely that his signal would be overshadowed by the users who are near the base station. In the IS-95 standard, perfect power control is utilized, which ensures that the received signal of any user within the cell is equal to any other. This requires a complicated control system on both base stations and mobile phones. Users at the far end of the cell usually consume

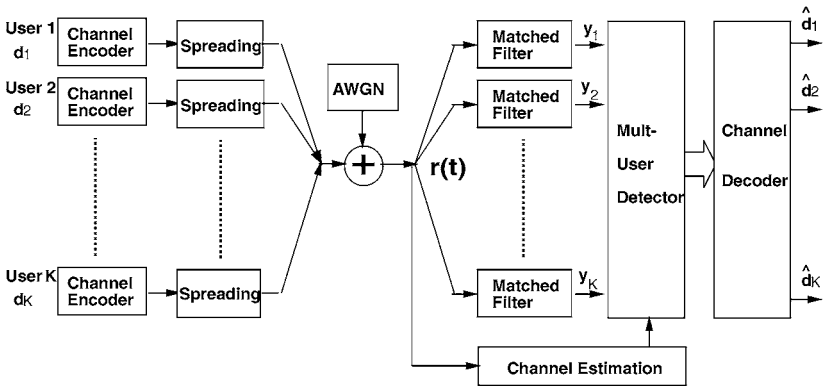


Figure 1. System diagram for a multiuser communication system (CDMA uplink physical layer): number of users K; original transmitted bits d_k; matched filter's output y_k; decoder hypothetical decisions $\hat{d_k}$.

extremely large amounts of power, which would inevitably shorten the battery life.

The assumption of simply considering all the other users as noise leads to the MAI and near-far problems [3]. One viable scheme is to use the cross-correlation information of all the users to do the linear or nonlinear multiuser detection [4] shown in Fig. 1, which requires a short code spreading scheme so that the cross-correlation information is determined. In a short code system, the spreading sequence is repetitive bit after bit (shown in Fig. 2) with different codes for each user. The channel estimation block in Fig. 1 is an essential part in a W-CDMA uplink system to estimate the delay and the amplitude information of each user. There are many advanced algorithms for channel estimation, such as maximum-likelihood estimation and subspace parameter tracking. Most of the proposed communication algorithms in W-CDMA systems consist of various matrix and vector level operations. Advanced computer arithmetic techniques, such as CORDIC, online arithmetic units [5], fast multiplier structures and so on, are especially valuable to the optimization and implementation of these algorithms. In this paper, we focus on the implementation of the multiuser detector block by using computer arithmetic techniques to reduce the complexity.

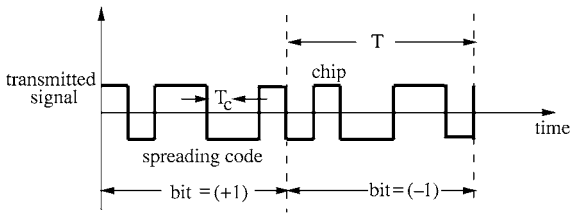


Figure 2. Spreading result with short codes: spreading code = {1, −1, 1, 1, −1, −1, 1}; chip duration T_c; bit duration T; spreading gain 7; Binary Phase Shift Keying (BPSK) modulation.

One group of multiuser detectors is based upon interference cancellation (IC), especially parallel interference cancellation (PIC). The concept is to cancel the interference generated by all users other than the desired user. Lower computation demand and hardware related structures are the major advantages of this strategy. One of the most effective PICs comes from the iterative multistage method, first proposed by Varanasi and Aazhang [6]. The inputs of one particular stage are the estimated bits of the previous stage. After interference cancellation, the new estimates, which should be closer to the transmitted bits, are fed into the next stage. Later researchers developed this multistage idea and introduced some other types of PICs [7]. However, almost all the existing multistage based algorithms neglect that fact that as the iterations progress, the solution becomes more invariant, i.e. more elements in the output vector turn out to be the same as the elements in the input vector. Ideally at the last iteration stage, the output and the input should be identical if the algorithm converges. Therefore in the last several stages, the multistage detector generates an output which is almost identical to its input. This is a substantial waste of computation power and increases the system delay.

Lin [8] developed a differential matched filter and presented an FPGA implementation [9], where they used the differential information in the FIR filter's coefficients to mitigate the complexity. This idea is important to our research on the complexity reduction for the multistage detector.

In this paper, we propose a differencing multistage detection algorithm. Unlike the conventional

multistage detector, the number of computations in each stage is not constant, but decreases dramatically stage after stage, which exactly reflects the characteristic of the iterative algorithm. Therefore the complexity is reduced, while in the meantime, the high performance of the interference cancellation of the multistage detector is preserved.

We have implemented both the conventional and the proposed differencing multistage detector in a single ASIC with a select function. This is because the differencing multistage detector uses the conventional detector as its first stage. Recent researchers also proposed various kinds of CDMA related matched filter, detector and decoder structures [9–11]. Compared to their approaches, our design focuses on the multiuser detector for the next generation W-CDMA and arithmetic level optimization. Our implementation was fabricated by MOSIS in 1.2 μm CMOS technology.

In the next section, we present the mathematical model of the multiuser communication system and our new differencing multistage detection algorithm. We will also analyze the convergence and fixed-point word length issues. An ASIC hardware implementation of this algorithm for real-time communication systems is shown in Section 3.

2. Differencing Multistage Detector

2.1. Multiuser Communication Model

We assume a multiuser binary phase shift keying (BPSK) modulated uplink DS-CDMA synchronous communication system. We could also extend this model to a general asynchronous system by adding the impact from adjacent bits, where an appropriate channel estimation block is required. The channel is a single path channel with additive white Gaussian noise (AWGN). Figure 1 shows the structure of a typical multiuser uplink communication system.

At the base station receiver, the continuous received signal $r(t)$ is given by:

$$r(t) = \sum_{k=1}^{K} \sqrt{\varepsilon_k}\, d_k(i) s_k(t - (i-1)T) + \eta(t), \quad (i-1)T \le t \le iT. \tag{1}$$

where K is the number of users. We obtain the estimates of the kth user's signal power $\sqrt{\varepsilon_k}$ from the channel estimation block. The source data bits are represented by $d_k(i)$. Here because we use BPSK modulation, $d_k(i) \in \{-1, +1\}$. The signature sequence s_k is the spreading code of the kth user, where T is the duration of one bit. A short repetitive Gold code sequence of period T is used to generate s_k in order to achieve better performance than random codes. Finally, $\eta(t)$ represents the Additive White Gaussian Noise (AWGN).

2.2. Matched Filters and Cross-Correlation Matrix

A matched filter bank is usually the first stage in the baseband signal detection. The technique of the matched filter bank in CDMA systems is to use one matched filter to detect one user's signal. There are no cross links among the filters. Each branch in the matched filter bank consists of the correlation operation of the received signal with one particular user's signature sequence. The ith output of the matched filter of the kth user is

$$y_k(i) = \frac{1}{T} \int_0^T r((i-1)T + \tau) s_k(\tau)\, d\tau \qquad k = 1, 2, \ldots, K. \tag{2}$$

Equation (2) can also be expressed in a simpler matrix notation on substituting (1),

$$\mathbf{y} = \mathbf{RAd} + \eta, \tag{3}$$

where vectors $\mathbf{y}$ and $\mathbf{d}$ are the output of the matched filter bank and the transmitted bits, respectively. There are K elements in each vector. In a synchronous system, the dimension of matrices $\mathbf{R}$ and $\mathbf{A}$ is $K \times K$. The elements in the cross-correlation matrix can be represented by:

$$r_{ij} = \begin{cases} \dfrac{1}{T} \displaystyle\int_0^T s_i(t) s_j(t)\, dt & i \ne j \\ 1 & i = j. \end{cases} \tag{4}$$

We can normalize the auto-correlation coefficients in (4) in our multistage detection algorithm because all the estimated bits are $+1$ or -1 within the multistage detector (we are interested only in the sign of these bits). The amplitude of each user would not affect the final hard decision. However, if we need to provide soft decision output for a later decoding block, we should also compute the real values of the auto-correlation coefficients.

The cross correlation matrix $\mathbf{R}$ can be split into three parts, as in (5):

$$\mathbf{R} = \mathbf{I} + \mathbf{L} + \mathbf{L}^{\mathbf{T}}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{L}$ is the lower triangular part of matrix $\mathbf{R}$. Since $\mathbf{R}$ is symmetric (shown in (4)), the upper triangular matrix should be the transpose of the lower triangular matrix.

The amplitude matrix of the signal $\mathbf{A}$ is represented as:

$$\mathbf{A} = \text{diag}\{\sqrt{\varepsilon_1}, \sqrt{\varepsilon_2}, \ldots, \sqrt{\varepsilon_K}\}. \tag{6}$$

$\mathbf{A}$ is a positive definite matrix with rank $\mathbf{K}$. Our differencing multistage detector is based on estimating the transmitted bits from (3) using a non-linear method.

2.3. *Derivation of the Differencing Multistage Detector*

The multistage detector is an interference cancellation scheme. In each stage of the multistage detector, PIC removes the component of other users from the received signal in parallel to obtain a better estimated signal for one particular user. Because we do not know the exact information bit for any user, we use the estimated (hard decision) bits $\hat{\mathbf{d}}$ in each stage. The output of the lth iteration is:

$$\begin{aligned} \mathbf{z}^{(l)} &= \mathbf{A}^{-1}\big[\mathbf{y} - (\mathbf{L} + \mathbf{L}^{\mathbf{T}})\mathbf{A}\hat{\mathbf{d}}^{(l-1)}\big] \\ \hat{\mathbf{d}}^{(0)} &= \text{sign}(\mathbf{A}^{-1}\mathbf{y}) \\ \hat{\mathbf{d}}^{(l-1)} &= \text{sign}\big(\mathbf{z}^{(l-1)}\big). \end{aligned} \tag{7}$$

After l iterations, we are more likely to observe $\hat{\mathbf{d}}^{(l)} = \hat{\mathbf{d}}^{(l-1)}$, which reflects the convergence of the iterative method. We observe that instead of dealing with each estimated bit vector $\hat{\mathbf{d}}^{(l)}$, as in (7), we can calculate the difference of the estimated bits in two consecutive stages, i.e. the input of each stage becomes $\hat{\mathbf{x}}^{(l)} = \hat{\mathbf{d}}^{(l)} - \hat{\mathbf{d}}^{(l-1)}$, which is called the differencing vector. If we denote $\mathbf{B} = \mathbf{A}^{-1}(\mathbf{L} + \mathbf{L}^{\mathbf{T}})\mathbf{A}$, Eq. (7) can be re-written as

$$\begin{aligned} \mathbf{z}^{(l)} &= \mathbf{y} - \mathbf{B}\hat{\mathbf{d}}^{(l-1)} = \mathbf{y} - \mathbf{B}\big(\hat{\mathbf{d}}^{(\mathbf{l-1})} - \hat{\mathbf{d}}^{(\mathbf{l-2})}\big) \\ &= \mathbf{z}^{(l-1)} - \mathbf{B}\hat{\mathbf{x}}^{(l-1)} \end{aligned} \tag{8}$$

$$\begin{aligned} \hat{\mathbf{d}}^{(l)} &= \text{sign}\big(\mathbf{z}^{(l)}\big) \\ \hat{\mathbf{x}}^{(l)} &= \hat{\mathbf{d}}^{(l)} - \hat{\mathbf{d}}^{(l-1)}. \end{aligned} \tag{9}$$

Using this differencing algorithm, computations can be saved by computing (8) instead of (7), as more elements in the vector $\hat{\mathbf{x}}^{(l)}$ tend to be zero after several iterations. Moreover, all the non-zero terms in $\hat{\mathbf{x}}^{(l)}$ (9) are equal to $+2$ or -2. The constant multiplication by ± 2 in (8) can be implemented by arithmetic shifts. Therefore, dedicated multipliers are not necessary for this algorithm. Finally, because our modification is a linear transformation that subtracts two consecutive stages, the bit error rate (BER) after each stage will not change, compared with the conventional multistage detector implementation. Therefore, the final BER is exactly the same as the conventional multistage detector. The complete algorithm is described below:

1. $\hat{\mathbf{d}}^{(0)} = \text{sign}(\mathbf{y})$
2. for $k = 1$ to K /* first stage conventional multistage detection */
3. $\quad z_k^{(1)} = y_k - \sum_{j=1}^{j=K} B_{kj}\hat{d}_j^{(0)}$
4. end
5. $\hat{\mathbf{d}}^{(1)} = \text{sign}(\mathbf{z}^{(1)})$
6. for $l = 1$ to L /* second and later stages: differencing multistage detection */
7. $\quad \hat{\mathbf{x}}^{(l)} = \hat{\mathbf{d}}^{(l)} - \hat{\mathbf{d}}^{(l-1)}$ /* differencing vector generation */
8. $\quad$ for $k = 1$ to K
9. $\quad\quad z_k^{(l+1)} = z_k^{(l)} - \sum_{j=1}^{j=K} B_{kj}\hat{x}_j^{(l)}$
10. $\quad$ end
11. $\quad \hat{\mathbf{d}}^{(l+1)} = \text{sign}(\mathbf{z}^{(l+1)})$ /* hard decision generation */
12. end

2.4. *Higher Modulation Schemes*

The differencing scheme can be easily extended to QPSK modulation, proposed in 3G systems as the real and imaginary components can be processed separately. To generalize this to other modulation schemes, the matrix transpose can be replaced with a Hermitian transpose using a generalized slicer in place of the sign function.

Let us first consider QPSK modulation. For QPSK, the transmitted bits are mapped into four different symbols: $1 + j$, $1 - j$, $-1 + j$, $-1 - j$. The output of the matched filter is now complex i.e. $\mathbf{y} \in \mathbb{C}^K$ and the matrix $(\mathbf{L} + \mathbf{L}')$ becomes $(\mathbf{L} + \mathbf{L}^*)$. Thus, (8) becomes

$$\begin{aligned} \mathbf{z}^{(l)} &= \mathbf{z}^{(l-1)} - \mathbf{B}\hat{\mathbf{x}}_{\mathbf{I}}^{(l-1)} - j\mathbf{B}\hat{\mathbf{x}}_{\mathbf{Q}}^{(l-1)} \\ \hat{\mathbf{d}}_{\mathbf{I}}^{(l)} &= \text{sign}\big(\Re\big(\mathbf{z}^{(l)}\big)\big) \\ \hat{\mathbf{d}}_{\mathbf{Q}}^{(l)} &= \text{sign}\big(\Im\big(\mathbf{z}^{(l)}\big)\big) \\ \hat{\mathbf{x}}_{\mathbf{I}}^{(l)} &= \hat{\mathbf{d}}_{\mathbf{I}}^{(l)} - \hat{\mathbf{d}}_{\mathbf{I}}^{(l-1)} \\ \hat{\mathbf{x}}_{\mathbf{Q}}^{(l)} &= \hat{\mathbf{d}}_{\mathbf{Q}}^{(l)} - \hat{\mathbf{d}}_{\mathbf{Q}}^{(l-1)} \end{aligned} \tag{10}$$

If we use two separate slicer functions to demodulate the real and imaginary part of QPSK modulated

WCDMA signal, we would have two independent hard decisions. The differencing method can be used in this case since the vector $\hat{\mathbf{x}}_I^{(l)}$ and $\hat{\mathbf{x}}_Q^{(l)}$ contain only 0 and ± 2's.

We can generalize the slicer function for more complex modulation schemes, such as MPSK and MQAM, the slicer function can be represented in the following equation:

$$\hat{\mathbf{d}}_i^{(l)} = \mathbf{S}_i\left(\hat{z}^{(l)}\right), \quad i = 0, 1, \ldots, m \tag{11}$$

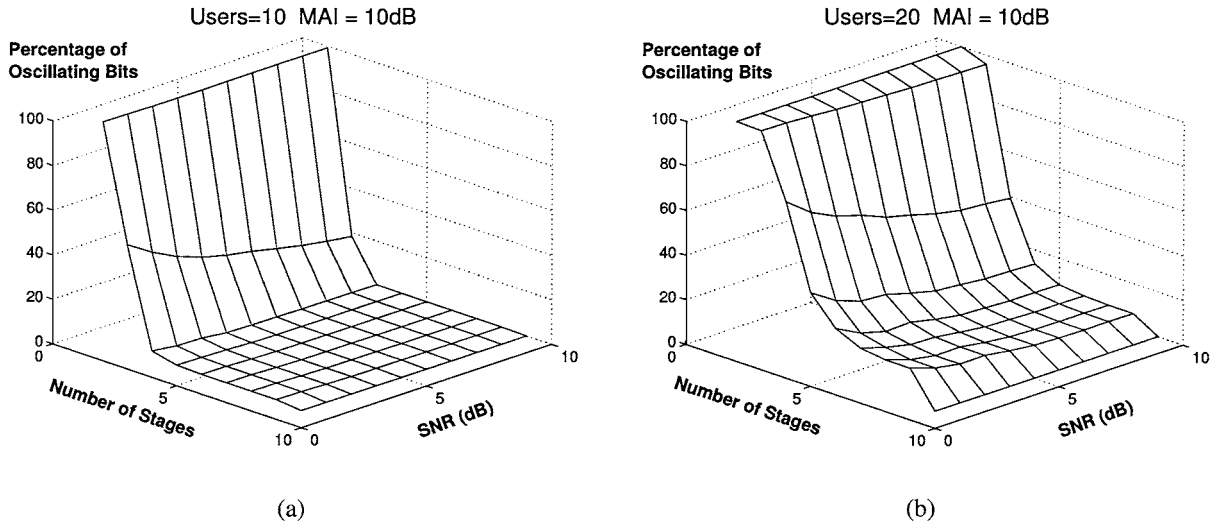
where $\hat{\mathbf{d}}_i^{(l)}$ is the ith bit in the hard decision symbol $\hat{\mathbf{d}}^{(l)}$ and m is the number of bits per symbol. The slicer function $\mathbf{S}_i$ maps the soft decision $\hat{\mathbf{z}}^{(l)}$ into the hard decisions $\hat{\mathbf{d}}_i$. By reconstructing the hard decisions using the estimated bits, we can still use (10) in the differencing computation mode. We are investigating this as future work.

2.5. *Numerical Results*

The differencing multistage detector is tested by Monte Carlo method with extensive simulations to estimate the convergence rate (shown in Fig. 3), given that iterations are forced to stop at the eighth stage. We assume a periodic Gold code sequence of length 31 as the spreading sequence. We observe that the differencing and conventional multistage detectors have the same convergence pattern and both of them work more effectively when SNR is high. Also, we observe that eight stages are sufficient for most cases, which guides the implementation of this algorithm.

The BER for the differencing multistage detector is exactly the same as the conventional multistage detector through the simulations. This is because we do not change the framework of the iterative method, nor the convergence rate. Equations (7) and (8) are essentially equivalent to each other. We define the multiple access interference (MAI) level as the ratio between the powers of the strongest and the weakest user. The BER plot versus SNR and MAI for a ten-user and twenty-user system is shown in Fig. 4. These figures show that the performance of the matched filter degrades dramatically when MAI increases or the number of users increases, which is due to the near-far and multiple access interference problem. In contrast, the performance of the differencing multistage detector, for moderate MAI and number of users, approaches the bound of a single user system, which is given by $P_e = Q(\sqrt{2E_b/N_0})$ where E_b/N_0 is the signal energy per bit to noise ratio.

The percentage of zeros, which in turn signifies the reduction in complexity, in the differencing vector is illustrated in Fig. 5(a). In this figure, we see that the percentage of zeros in the differencing vector increases as the iterations progress, which shows that the iterations converge progressively. After the fourth stage, the number of zeros approaches 98% in a 15-user communication system. This result explicitly indicates that if we use the conventional multistage detector, almost 98% of the computation resource is unnecessary in the fourth stage. Figure 5(b) gives us a clear view of how many



(a) (b)

Figure 3. Percentage of non-zero elements in vector $\hat{\mathbf{x}}$ in the multistage detection (a) $K = 10$, (b) $K = 20$.

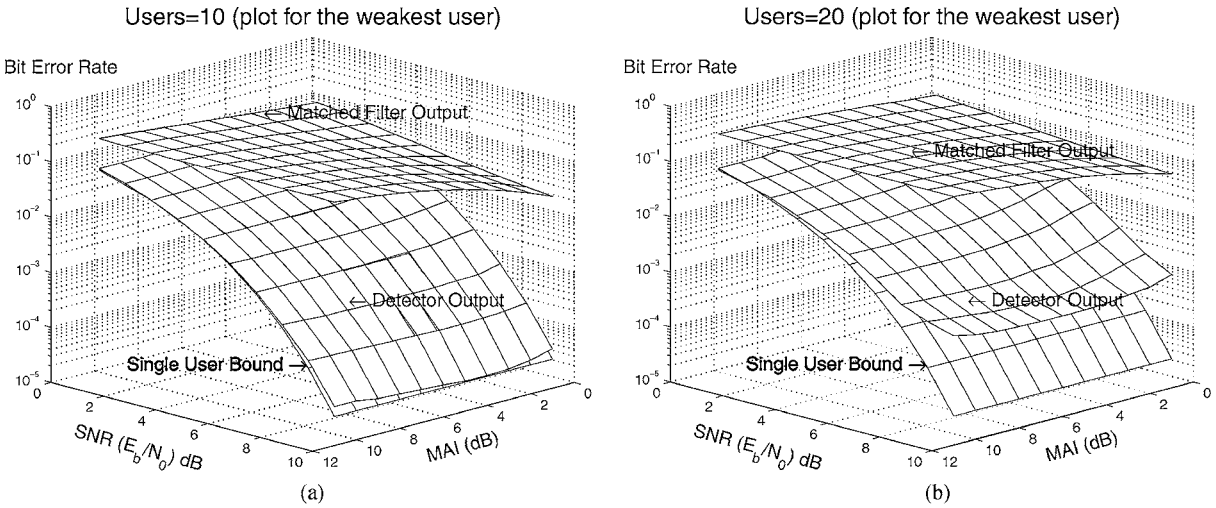


Figure 4. BER of the differencing multistage detector (a) $K = 10$, (b) $K = 20$.

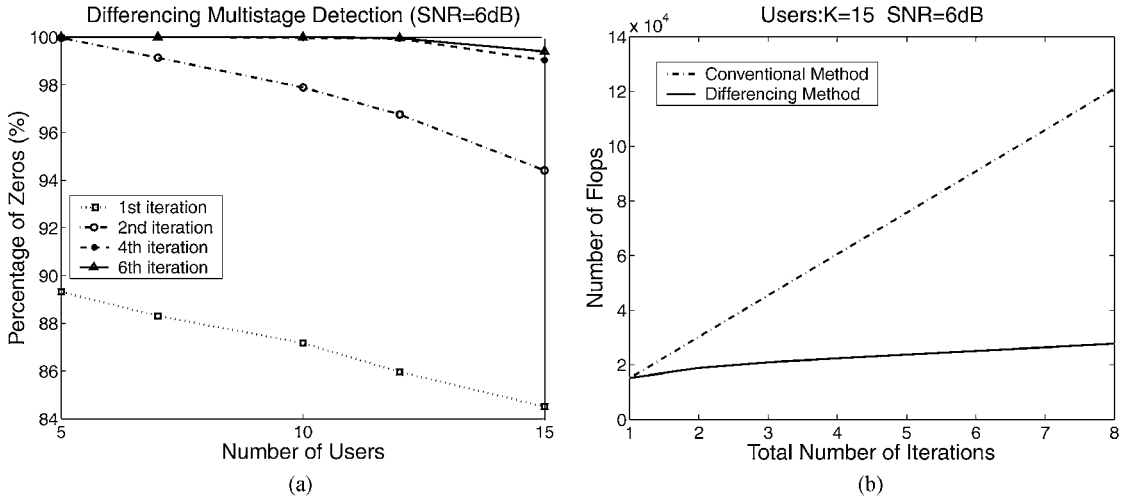


Figure 5. (a) Percentage of zeros in the differencing vector. (b) Flops comparison between differencing and non-differencing method.

computations are possible to save in a real system. The dotted line represents the accumulated number of floating point operations (flops) needed after each stage in the conventional multistage detector. As we explained earlier, the number of computations remains constant for each stage, which makes the total flops increase linearly. On the contrary, the number of computations in the differencing multistage detector decreases as the iteration proceeds. Thus we can achieve a $6\times$ speedup in an eight stage system according to Fig. 5(b). With more stages in the system to increase the BER, higher speedups are obtained relative to the conventional multistage detector.

3. Real-Time Implementation

The detector can be implemented in real-time by both DSPs and ASICs. Although high performance general purpose DSPs can meet the real-time requirements, they are not as cost-effective. In commercial communication systems, sophisticated algorithms tend to be implemented by dedicated ASICs. These hardware implementations are potentially cheaper and faster with lower power consumption [12–14]. In this section, we present a fixed-point implementation analysis and our ASIC implementation of the differencing multistage detector.

3.1. *Fixed-Point Implementation Analysis*

Converting an algorithm from floating point to fixed point requires two major procedures. First, we have to estimate the dynamic range of the input data and all the variables used in the algorithm. Also, we have to find an optimized wordlength to represent numbers and truncate the results. In this section, we present an analysis of the fixed-point implementation of the differencing multistage detector.

3.1.1. Range Estimation. The cross-correlation coefficients from the channel estimation block and the matched filter output from integrators are two major operands in the differencing multistage detector. Both are generated by high speed analog to digital (A/D) converters, which sample and digitize the analog input signals at the front end.

From the characteristics of the Gold code, we know that the maximum value of cross-correlation coefficients is the auto correlation of any particular spreading sequence, i.e., range R_r is

$$R_r = 2 \times (2^r - 1) \tag{12}$$

where the spreading gain is $2^r - 1$. Therefore $R_r = 62$ if we use a Gold code of length 31. The range of the user's amplitude depends on the dynamic range (or MAI) of the system. The relationship is the following,

$$R_a = 10^{\frac{\text{MAI}}{20}}. \tag{13}$$

The range estimation for the matched filter output is complicated because it is determined by SNR, MAI, and the number of users in the system. For sufficiently large number of interfering users with balanced interfering powers, the interference can be approximately modelled by a Gaussian distribution [15], as illustrated in Fig. 6. The distribution is also symmetric, based on the assumptions of BPSK modulation, binary distribution of the source bits, and the binary symmetric channel.

The range of such a distribution is estimated as

$$R_m = 2 \times (|\mu| + n\sigma), \tag{14}$$

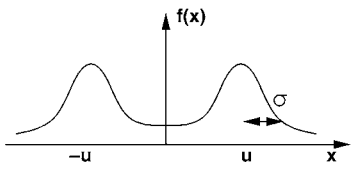


Figure 6. Dynamic range estimation.

where μ is the mean of one peak, σ is the standard deviation of that peak and n is an empirical constant. For the Gaussian distribution [14], $n = 3$ can guarantee 99.9% of all the samples fall in the range R_m.

3.1.2. Wordlength Analysis. From (12) and (13), we can conclude that the number of bits [14] needed to represent the result of matrix product **RA** in (3) is

$$\begin{aligned} L_{\min} &= \lceil \log_2(R_r \times R_a) \rceil \\ L_{\min} &= 1 + r + \frac{\text{MAI}}{20} \log_2 10. \end{aligned} \tag{15}$$

Here we assume a binary representation of the integers. If MAI = 10 dB and $r = 5$ (Gold code of length 31), $L_{\min} = 8$, which indicates that at least eight bits are needed to represent any cross-correlation coefficient.

For the matched filter output, the number of bits needed is nine in a perfect power control case, and ten in a MAI = 10 dB case for up to 20 users (shown in Fig. 7(a)). In Fig. 7, we can also observe that if the number of users is small, SNR will dominate the variation of the dynamic range. When more users are active in the system, MAI will determine the number of bits required.

For some applications, the optimized wordlength might not follow the relation in (15), but will usually be smaller than $L_{\min}$. The optimized wordlength is determined by simulation, in which the minimal mean square distortion is set corresponding to a particular performance requirement.

3.2. *Complexity Analysis*

Further investigations show that the differencing vector $\hat{\mathbf{x}}$ has over 80% zeros after the first iteration in general (shown in Fig. 5), which can be regarded as a sparse vector. When solving (8), instead of K^2 additions, we can deal with only the non-zero terms. In the second iteration for example, the total computations will shrink to approximately $0.2K^2$. The theoretical result shows that the total number of computations per user is linear in the number of users in the system.

Since we have mitigated all the multiplication operations to simple additions and shifts, dedicated multipliers are not necessary. However, advanced computer arithmetic techniques, such as full carry look-ahead adders, online arithmetic units [5], etc. are essential to achieve the real-time performance.

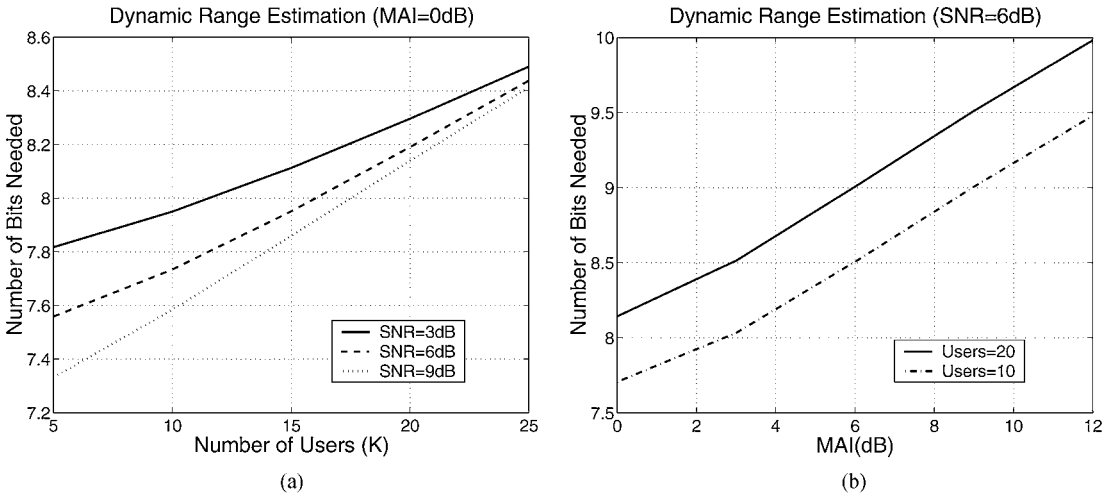


Figure 7. Dynamic range estimation by varying (a) SNR and K, (b) MAI and K.

3.3. Prototyping the Differencing Multistage Detection Algorithm

The structure of the first three stages of the differencing multistage detector is shown in Fig. 8. In the first stage, the PIC uses the previous estimates (from the matched filter output) to generate a new vector of estimated bits. We need a conventional multistage detector as the first stage, so that two initial vectors are obtained for the differencing method.

After the first stage, the differencing multistage detector starts to use the differencing vector $\hat{\mathbf{x}}^{(1)}$ as the input, which is generated by subtracting the input hard decision from the previous hard decision. In $\hat{\mathbf{x}}^{(1)}$, additional zeros should be observed. Furthermore, the inputs for the interference cancellation are not the matched filter output, but the previous stage's output $\mathbf{z}^{(1)}$. Later stages repeat the same structure of the differencing multistage detector stage II, and further computations can be saved.

Figure 9 is our architecture to implement one stage of the differencing multistage detector for synchronous users in hardware using a single custom chip. Using the select function, if we bypass the differencing vector and the arithmetic shift of the cross-correlation constants, it can also be used as a conventional multistage detector.

Soft decision inputs and outputs are generated in parallel for each user and all users are detected in a serial manner. The timing of inputs and outputs is controlled by a hand shaking mechanism. The input numbers are in two's complement format and they are stored in the data register bank. At the same time, the hard decisions are acquired from the sign bit of the soft decision and the differencing vector is generated by combinational

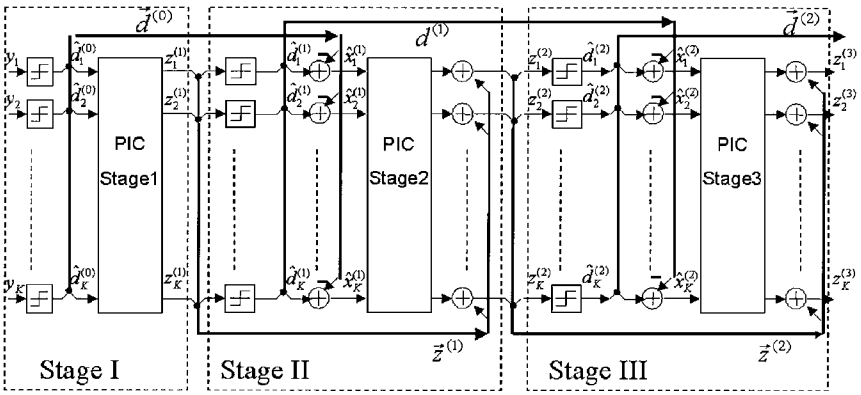


Figure 8. The structure of the differencing multistage detector.

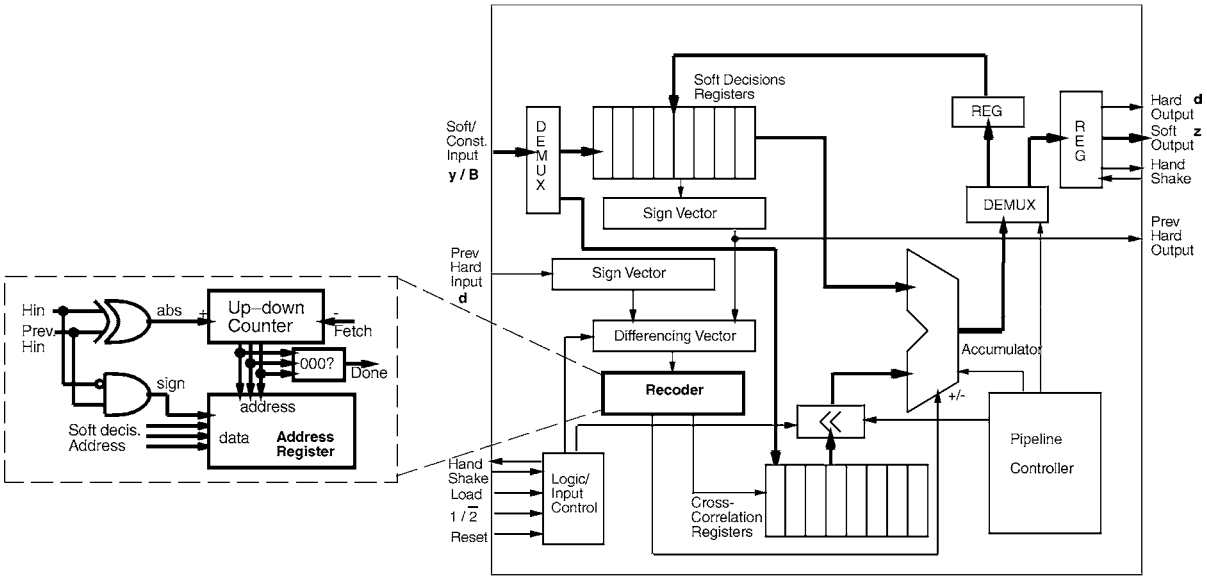


Figure 9. Single ALU implementation of differencing multistage detector.

logic. The recoder block (highlighted in Fig. 9) implements the key features of the differencing multistage detector by selecting all the non-zero elements and tagging their addresses. The timing for the accumulation is scheduled according to the positions of the non-zero elements. If an element is not zero, the recoder will pick out the corresponding cross-correlation data, and update all the soft decisions by subtracting or adding it, according to the sign of the differencing vector's element. Loading, shifting, accumulating and writing back are organized as a simple pipeline machine, managed by a two-phase clock. The pipeline will not stall because no data or control dependencies exist. Finally the soft and hard decisions are generated one by one with certain handshaking protocols to the next stage.

3.4. Chip Specifications

Table 1 summarizes our prototype chip specifications. To simplify the hardware design, we have focused on fixed-point implementation of a synchronous system and the design is based on an eight-user Gold code spreading system. However, it can be extended to a random code, asynchronous system with a variable number of users. We choose an eight-user system since all the control logic is primarily binary counters. Therefore, a number of users with a power of 2 would be most efficient. The input data bus is limited by the pin count of our prototype chip. In order to meet the fixed point word length requirement, as determined in the analysis in Section 3.1, we choose 10 bits as the input precision. The detector allows us to detect eight users in a MAI = 15 dB and SNR = 6 dB environment. The internal data bus is wider than the input or output bus to ensure that no overflow would occur during intermediate computations.

Figure 10(a) shows the actual chip die photo. The chip has five major blocks: recoder, 12-bit carry look-ahead adder, register banks for cross-correlation coefficients, soft decision registers and the address information of the non-zero elements. Some programmable logic arrays (PLAs) and temporary registers are necessary for control and pipeline management [16]. Figure 10(b) is the timing diagram from the real-time test of the differencing multistage detector chip. The complete process includes: loading cross-correlation matrix R, first PIC stage on the first chip, and second and later stages on the second chip.

Table 1. The chip specification.

Number of users	8 (synchronous)
Soft decision precision	10-bit fixed point
Cross correlation precision	8-bit fixed point
Width of internal bus	12 bits
Clock rate	Two-phase clock at 12.5 MHz
Throughput	190 Kbps/user at 12.5 MHz
Transistor count	6K
Die size	2.2×2.2 mm^2

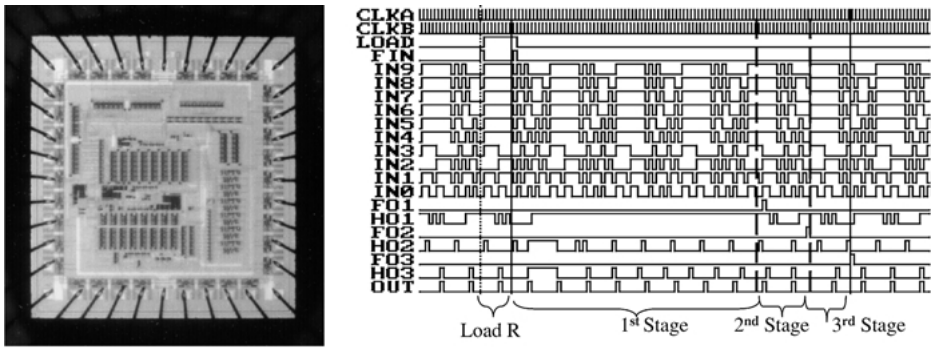


Figure 10. (a) ASIC chip layout: 1.2 μm, 2.2 $\times$ 2.2 mm^2 die. (b) Chip testing results show the second and third stage performance improvement.

3.5. Two Chip Multistage Detector with Predictable Delay

Our chip implements a single stage of the conventional/differencing multistage detector. A complete multistage detector is implemented by simply cascading two chips together with a proper feed back path and glue logic. The flow of data between the two chips is controlled by a simple hand shaking mechanism as we know that the next iteration for detection will rake time lesser than or equal to that of the previous stage. The first chip conducts the conventional multistage detection. As a complete matrix-vector operation (7), is performed in the conventional detector, the delay is constant. The second stage is configured as a differencing multistage detector, the output of which is fed back to its own input after the first differencing multistage detection iteration. Since the number of clock cycles required decrease for each iteration, multiple iterations of interference cancellation can be run on the second chip within the processing latency of the first chip. The throughput is determined by the clock rate

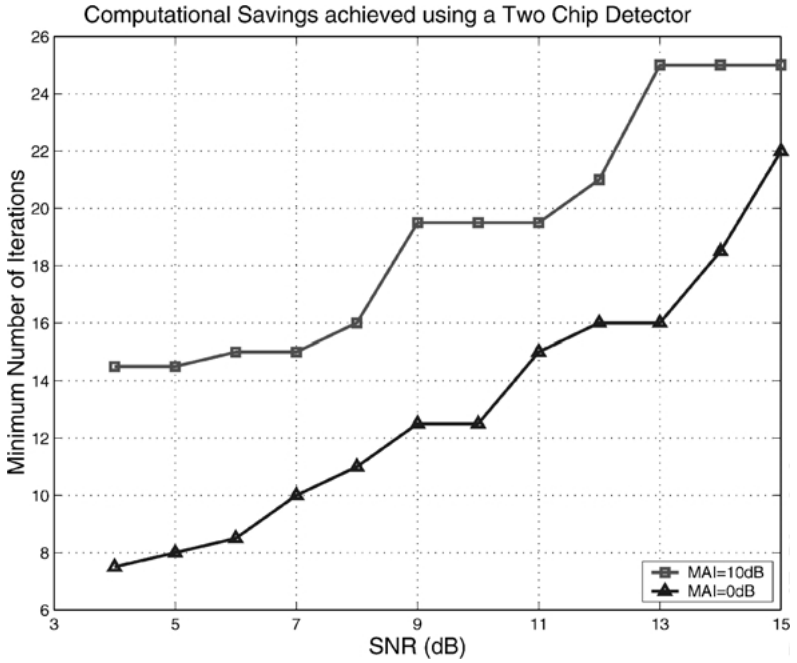


Figure 11. Computational savings achieved by a two chip differencing detector. The figure shows the number of iterations of a regular multistage detector possible with a two chip differencing detector.

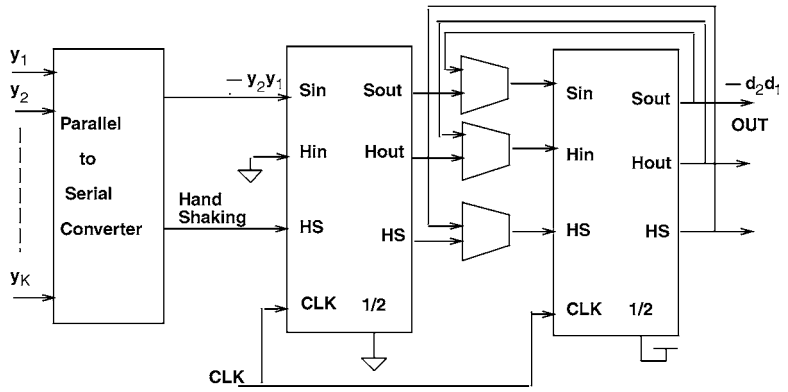


Figure 12. Cascade mode of the differencing multistage detector: where y_k is the matched filter output for user K; d is the final hard decision; "Hin" and "Hout" represent hard decision input and output respectively; "Sin" and "Sout" represent soft decision input and output respectively; "1/2" selects the first stage or later stages; "HS" is the hand-shaking port.

of both chips and the delay is simply two stages of conventional multistage detector.

Figure 11 shows the computational savings obtained by using the differencing technique over the conventional detection scheme. The figure shows the amount of iterations using the differencing method that are possible within a single iteration of the conventional method. For the worst operation case at SNR = 4 dB MAI = 0 dB, the two chip differencing system can execute at least seven iterations in the time taken for two iterations of a conventional detector. Figure 11 also shows when the SNR increases, the computational savings are higher and more iterations of the differencing scheme are possible. This is due to the reduction in noise, resulting in lower BER and faster convergence in the detection process. Also, it can be seen that higher MAI (10 dB) results in faster convergence and hence, more iterations can be performed for higher MAI. This is because MAI = 0 dB implies the equal power case (worst case) for all users. It should be noted that the Fig. 11 only conveys the computational savings due to the differencing scheme and 8 iterations are sufficient in most cases for convergence.

A cascade-mode two chip differencing multistage detector is shown in Fig. 12. Two ASICs are cascaded in a chain, driven by the same clock. From our hardware testing (shown in Fig. 10(b)), the two chip system delay with the differencing algorithm is less than 70 cycles. Working at a clock rate of 12.5 MHz, the system delay is about 6 μs, much less than that of the conventional multistage detector, which is around 48 μs for eight stages. Using our design, the system can reach a throughput up to 190 Kbps with proper buffering. This rate meets the 144 Kbps requirement of the W-CDMA communication proposals [2].

3.6. *Scalable ASIC Design*

Our hardware implementation shows the real-time performance in the communication system. We could estimate the size for a commercial base station detector chip in Table 2. If we design a chip which can handle 30 asynchronous users (upper limit for Gold code of length 31 system), it would require three full carry look-ahead adder as the ALU. The cross-correlation matrix has $30^2 = 900$ elements, each one of which has 8-bit precision (according to Section 3.1). We could expand the data bus width to 16 bits in order to accommodate higher MAI. Total number of register cells are $900 \times 8 + 30 \times 16 \approx 8$ Kb. If a conservative static

Table 2. Scalable ASIC design for the multistage detector.

	Prototyped chip specifications	Commercial detector requirements
Capacity	8 synchronous users	30 asynchronous users
Precision	12-bit	16-bit
Clock Rate	12.5 MHz	100 MHz
Internal registers	0.3 Kb	8 Kb
ALU	partial carry look-ahead adder	three full carry look-ahead adders
Transistors	6K	100K

register cell consists of approximately 10 transistors, the total number of transistors would be around 100 K.

4. Conclusion

In this paper, we have focused on the real-time implementation issues for the multistage detection algorithm in Wideband CDMA receivers. We developed a novel differencing multistage detection algorithm, by exploiting the convergence property of the iterative algorithm to greatly reduce the complexity of the multistage detector. The new differencing multistage detector computes the difference of vectors between two consecutive stages and saves computations when the difference becomes zero. We designed an ASIC chip to implement the differencing multistage detector. The chip was fabricated by 1.2 μm CMOS technology with a die size of 2.2×2.2 mm^2. Two cascaded chips can perform at least eight stages of multistage detection with a throughput of 190 Kbps/user and around 6 μs delay in an eight-user system. The architecture is scalable for a larger design.

Acknowledgments

We are extremely grateful to the reviewers for their valuable suggestions. We would like to acknowledge the help of Praful Kaul in the VLSI implementation of the differencing multistage detector. We would also like to thank Srikrishna Bhashyam and Ashutosh Sabharwal for discussions and ideas. This work was supported by Nokia Inc., Texas Instruments, Texas Advanced Technology Program under grants 1997-003604-044 and 1999-003604-080, and NSF under grants NCR-9506681 and ANI-9979465.

References

1. J.G. Proakis, *Digital Communications*, New York: McGraw-Hill, 1989.
2. "Third Generation Partnership Project," http://www.3gpp.org
3. S. Moshavi, "Multi-User Detection for DS-CDMA Communications," *IEEE Communications Magazine*, Oct. 1996, pp. 124–136.
4. S. Verdú, *Multiuser Detection*, Cambridge University Press, Cambridge, 1998.
5. M.D. Ercegovac, "On-Line Arithmetic: An Overview," in *Real Time Signal Processing VII, SPIE*, vol. 495, 1984, pp. 86–93.
6. M.K. Varanasi and B. Aazhang, "Multistage Detection in Asynchronous Code-Division Multiple-Access Communications," *IEEE Transactions on Communications*, vol. 38, no. 4, 1990, pp. 509–519.
7. D. Divsalar, M.K. Simon, and D. Raphaehi, "Improved Parallel Interference Cancellation for CDMA," *IEEE Transactions on Communications*, vol. 46, no. 2, 1998, pp. 258–268.
8. W. Lin, "Differentially Matched Filter for a Spread Spectrum System," United States Patent 5,663,983, Sept. 1997.
9. K. Liu, W. Lin, and C. Wang, "Pipelined Digital Differential Matched Filter FPGA Implementation and VLSI Design," in *IEEE Custom Integrated Circuits Conference*, San Diego, CA, May 1996, pp. 75–78.
10. J.K. Hinderling, T. Rueth, K. Easton, D. Eagleson, D. Kindred, R. Kerr, and J. Levin, "CDMA Mobile Station Modem ASIC," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 3, 1993, pp. 253–260.
11. I. Kang and A.N. Willson, "Low-Power Viterbi Decoder for CDMA Mobile Terminals," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, 1998, pp. 473–482.
12. C. Sengupta, S. Das, J.R. Cavallaro, and B. Aazhang, "Fixed Point Error Analysis of Multiuser Detection and Synchronization Algorithms for CDMA Communication Systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seattle, WA, May 1998, vol. 6, pp. 3249–3252.
13. S. Das, C. Sengupta, J.R. Cavallaro, and B. Aazhang, "Hardware Design Issues for a Mobile Unit for Next Generation CDMA Systems," in *Proceedings of SPIE—Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, San Diego, CA, July 1998, vol. 3461, pp. 476–487.
14. S. Kim, K. Kum, and W. Sung, "Fixed-Point Optimization Utility for C and C++ Based Digital Signal Processing Programs," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 45, Nov. 1998, pp. 1455–1464.
15. H.V. Poor and S. Verdú, "Probability of Error in MMSE Multiuser Detection," *IEEE Transactions on Information Theory*, vol. 43, no. 3, May 1997, pp. 858–871.
16. G. Xu and J.R. Cavallaro, "Real-Time Implementation of Multistage Algorithm for Next Generation Wideband CDMA Systems," in *Advanced Signal Processing Algorithms, Architectures, and Implementations IX, SPIE*, Denver, CO, July 1999, vol. 3807, pp. 62–73.

Gang Xu received his bachelor's degree in electrical engineering from Tsinghua University, Beijing, China in 1996 and master's degree in computer engineering from Rice University, Houston, TX in 1999.

From 1996 to 1997, he worked in Zhenzhong Electronic Corp., Beijing, China, as a hardware engineer in charge of palmtop computer design. In 1999, he joined Nokia Research Center in Irving, TX.

Since then, he has involved in the rapid prototyping of the OFDM radio using FPGAs and digital signal processors. His current research interests include signal processing architectures and rapid prototyping of advanced wireless algorithms.
gang.gary.xu@nokia.com

Sridhar Rajagopal received his B.E. (Honors with Distinction) in Electronics Engineering from VJTI, Bombay University, India in 1998 and his M.S. in Electrical and Computer Engineering from Rice University in 2000. He is currently a doctoral candidate at Rice University. His research interests are in wireless communications, VLSI signal processing, computer arithmetic and computer architecture. Web: http://www.ece.rice.edu/∼sridhar/
sridhar@rice.edu

Joseph R. Cavallaro was born in Philadelphia, Pennsylvania on August 22, 1959. He received the B.S. degree from the University of Pennsylvania, Philadelphia, PA, in 1981, the M.S. degree from Princeton University, Princeton, NJ, in 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, in 1988, all in electrical engineering.

From 1981 to 1983, he was with AT&T Bell Laboratories, Holmdel, NJ. In 1988, he joined the faculty of Rice University, Houston, TX, where he is currently an Associate Professor of Electrical and Computer Engineering, having received tenure in 1994. His research interests include computer arithmetic, fault tolerance, VLSI design and microlithography, and DSP and VLSI architectures and algorithms for applications in wireless communications and robotics.

Dr. Cavallaro is a recipient of the NSF Research Initiation Award 1989–1992 and the IBM Graduate Fellowship 1987–1988, and is a member of IEEE, Tau Beta Pi and Eta Kappa Nu. During the 1996–1997 academic year, he served at the U.S. National Science Foundation as director of the Prototyping Tools and Methodology Program in the Computer (CISE) Directorate. He is currently the Associate Director of the Center for Multimedia Communication at Rice University. Web:http://www.ece.rice.edu/∼cavallar/
cavallar@rice.edu

Behnaam Aazhang received his B.S. (with highest honors), M.S., and Ph.D. degrees in Electrical and Computer Engineering from University of Illinois at Urbana-Champaign in 1981, 1983, and 1986, respectively.

From 1981 to 1985, he was a Research Assistant in the Coordinated Science Laboratory University of Illinois. In August 1985, he joined the faculty of Rice University, Houston Texas, where he is now the J.S. Abercrombie Professor in the Department of Electrical and Computer Engineering and the Director of Center for Multimedia Communications. He has been a Visiting Professor at IBM Federal Systems Company, Houston, Texas, the Laboratory for Communication Technology at Swiss Federal Institute of Technology (ETH), Zurich Switzerland, the Telecommunications Laboratory at University of Oulu, Oulu, Finland, and at the U.S. Air Force Phillips Laboratory, Albuquerque, New Mexico. His research interests are in the areas of communication theory, information theory, and their applications with emphasis on multiple access communications, cellular mobile radio communications, and optical communication networks.

Dr. Aazhang is a Fellow of IEEE, a recipient of the Alcoa Foundation Award 1993, the NSF Engineering Initiation Award 1987–1989, and the IBM Graduate Fellowship 1984–1985, and is a member of Tau Beta Pi and Eta Kappa Nu. He is currently serving on Houston Mayor's Commission on Cellular Towers. He has served as the Editor for Spread Spectrum Networks of IEEE Transactions on Communications 1993–1998, as the Treasurer of IEEE Information Theory Society 1995–1998, the Technical Area Chair of 1997 Asilomar Conference Monterey, California, the Secretary of the Information Theory Society 1990–1993, the Publications Chairman of the 1993 IEEE International Symposium on Information Theory San Antonio, Texas, and as the co-chair of the Technical Program Committee of 2001 Multi-Dimensional and Mobile Communication (MDMC) Conference in Pori, Finland. Web:http://www.ece.rice.edu/∼aaz/
aaz@rice.edu