International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 4, No. 2, July 2015, pp. 82~98
ISSN: 2089-4864

Real-Time Algorithms and Architectures for Several User Channel Detection in Wireless Base Station Receivers

Nitish Meena*, Nilesh Parihar**
* Dept. of ECE, pratap University (MPGI) Jaipur, Rajasthan, India, Dept. Of E.C.E Phd (pur.)
** Principal in Bhartiya Institute of Engineering & Technology, Sikar, Rajasthan, India

Article Info

Article history:

Received Dec 15, 2014
Revised Mar 21, 2015
Accepted Apr 15, 2015

Keyword:

Architecture
Base station receivers
Channel Detection
Real-time algorithms
Wireless

ABSTRACT

In this paper presents algorithms and architecture designs that can meet real-time requirements of for several user channel estimation and detection in code-division multiple-access-based wireless base-station receivers. Entangled algorithms proposed to implement several user channel assessment and demodulation make their real-time execution difficult on current digital signal processor-based receivers. A based several user channel assessment scheme requiring matrix conversion is draft again from an demodulation perspective for a reduced intricacy, repetitive scheme with a simple fixed-point very large scale integration architecture. A reduced-intricacy, bit-streaming several user demodulation algorithms that avoids the need for demodulation is also developed for a simple, pipelined VLSI architecture. Thus, we develop real-time solutions for several user channel assessment and demodulation for third-generation wireless systems by: 1) designing the algorithms from a fixed-point execution perspective, without significant loss in error rate performance; 2) task partitioning; and 3) designing bit-streaming fixed-point VLSI architectures that explore pipelining, correspondence, and bit-level computations to achieve real-time with minimum area overhead.

Corresponding Author:

Nitish Meena,
Dept. of ECE, pratap University (MPGI) Jaipur,
Rajasthan, India
Email: lsntl@ccu.edu.tw

1. INTRODUCTION

A (3G) wireless cellular system is being designed to support very strange data rates (in Mb/s) and quality-of-service (QoS) guarantees that are required for multimedia communication. Wideband code-division multiple-access has been chosen as the multiple-access protocol to support these features. A existing pertaining to a small bandwidth CDMA standard supports only voice and low-data rates up to 9.8 kb/s and uses single-user algorithms at the base-station receiver that ignore multiple access interference between different users. To achieve improved presentation at these high data rates, highly intricate and complex several user algorithms for channel assessment and demodulation need to be executed. These algorithms combat MAI by jointly processing the signals of all users at the base-station receiver. The multiuser algorithms engage matrix multiplications and conversions require block-based reckoning and resting point accuracy and significantly increase the execute intricacy of the receiver. A direct execution of these several user algorithms using current generation digital signal processor based base-station receivers' fails to meet third-generation real-time necessity. Therefore, only single user algorithms for channel assessment and detection have been execution in all current practical CDMA systems, such as IS-95. execute for several user detection for the base station have been studied in and while low power versions intended to at mobile handsets have been studied in. However, these detector execution either assume perfect channel assessment or assume single user assessment using sliding- mutually releted type structures. The detector execution also

assumes that channel assessment is done in real-time and the data rates are considered to be dependent only on the detector. However, many advanced several user channel assessment schemes have high reckoning intricacy, even more than that for several user detection, due to matrix inversions involved and cannot be performed in real-time. Also, algorithms for assessment and detection are block-reckoning based due to the need for repeated conversion updates for assessment multishot detection, which make their real-time execute more difficult. Matrix-inversion free schemes such as those based on conjugate gradient descent and recursive least squares exist in the literature. We have appraise the applicability of such schemes for several user channel assessment and presented one such scheme with low computational intricacy and suitable for assessment. Jointly performing multiuser channel assessment and detection is shown to have lower computational intricacy and better error rate performance than performing several user estimation and detection separately. Hence, we shall consider this joint algorithm for several user channel assessment and demodulation for draft again from a very large scale integration architecture perspective. Similar work on a joint channel assessment and detection scheme for time division multiple access systems with a systolic execution for Kalman filterin is presented in. They have also studied word-length effects and provided comparisons with least mean square and RLS schemes. In this paper, we present efficient algorithms for several user channel estimation and detection, designed from an implementation perspective and their mapping to real-time VLSI architectures. We redesign a several user channel assessment algorithm, based on the maximum-likelihood principle and present an iterative scheme, which is reckonable effective, suitable for a fixed point execution and is equivalent to matrix inversion in terms of error rate performance. A new bit-streaming several user detection scheme based on parallel interference revocation is presented that avoids the need a multishot detection for a simple bit-streaming pipelined VLSI architecture. Fixed-point a execution of the draft again algorithms are presented. First, we determine the maximum data rate achievable with no area restriction. Then, we obtain the data rate achieved by an area-constrained architecture. Finally, we present area-time tradeoffs for real-time VLSI architectures to achieve the intended to data rates with minimum area overhead. Thus, the main contribution of this paper is to show real-time performance for several user algorithms-1 designing the algorithms from a fixed-point architecture perspective, without significant loss in error rate performance; 2) task partitioning; and 3) designing bit-streaming fixed point VLSI architectures to exploit available pipelining, parallelism and bit-level computations.

2. SEVERAL USER CHANNEL ESTIMATION AND DETECTION

A. Real-Time Requirements

Data transmission in 3G wireless systems such as third-generation partnership project (3GPP) or universal mobile telecommunications systems (UMTSs) is possible at varying rates such as from 32 kb/s to 2 Mb/s depending on the spreading factor which varies from 256 (for vehicular traffic) to 4 (for indoor environments), respectively (for example, see). The standards assume a chip rate of 4.097 Mcps and quadrature phase-shift keying modulation (2 bits/ symbol). We have assumed binary phase-shift keying modulation (1 bit/symbol) in our work for simplicity. Hence, we target data rates in the range of 18 kb/s to 1 Mb/s. However, our proposed algorithms as well as our work on fixed-point analysis, pipelining, and parallelism can be extended to higher modulation schemes as well. We propose different architectures which explore area-time trade-offs in order to achieve these data rates. We seek to design architectures that meet real-time requirements to within an order-of magnitude. Specifically, we target architecture designs for different ($N = K =$ 4, 16, 32, 128, 256). Achieve data rates of 16 kb/s, 64 kb/s, 128 kb/s, 256 kb/s, and 1 Mb/s, respectively. Note that the reference to 3G systems is solely as an example to illustrate important system features such as the varying data rates which we seek to target and the use of training sequences for channel assessment.

B. Received Signal

We assume BPSK modulation and use direct sequence spread spectrum signaling, where each active mobile unit possesses a unique signature sequence (short repetitive spreading code) to modulate the data bits (1). The base station receives a addition of the signals of all the active users after they travel through different paths in the channel. The multipath is caused due to reflections of the transmitted signal that arrive at the receiver along with the line-of-sight component. These channel paths induce different delays, attenuations and phase-shifts to the signals and the mobility of the users causes fading in the channel. Moreover, the signals from different users interfere with each other in addition to the additive white Gaussian noise present in the channel. Several user channel assessment refers to the joint assessment of these unknown parameters for all users to mitigate these undesirable effects and accurately detect the received bits of different users. Several user detection refers to the detection of the received bits for all users jointly by canceling the

interference between the different users. The performance of several user demodulation depends greatly on the accuracy of the channel estimates. The model for the received signal at the output of the multipath channel [13] can be expressed a $\mathbf{r}_i = \mathbf{A}\underline{\mathbf{d}}_i + \mathbf{n}_i$ ..(1) where $\mathbf{r}_i \in \mathbb{C}^N$ is the received signal vector after chip-matched filtering [5], [20], $\mathbf{A} \in \mathbb{C}^{N \times 2K}$ is the effective spreading code matrix, containing information about the spreading codes (of length) attenuation and delays from the various paths, $\underline{\mathbf{d}}_i \in \{-1, +1\}^{2K} = [d_{1,i-1}, d_{1,i}, \ldots, d_{K,i-1}, d_{K,i}]^\top$ are the bits of users to be detected, $\mathbf{n}_i$ is AWGN and is the time index. The size of the K data bits of the users is as we assume that all paths of all users $\underline{\mathbf{d}}_i$ is 2K are coarse operating simultaneously to within one symbol period from the arbitrary timing reference. Hence, only two symbols of each user will overlap in each observation window. This model can be easily extended to include more general situations for the delays, without affecting the derivation of the channel assessment algorithms. The assessment of the matrix **A** contains the effective spreading code of all active users and the channel effects and is used for accurately demodulation the received data bits of different users. We will call this estimate of the effective spreading code matrix, **A** our channel estimate as it contains the channel information directly in the form needed for demodulation. This approach is chosen as it provides: 1) a single frame work for both channel assessment and detection and 2) both reckonable and performance gains. Most other several user channel estimation techniques try to assessment the individual channel attenuations and hinder instead of the effective spreading code.

C. Several User Channel Tracking

The block diagram of the base-station receiver is shown in Figure 1. The several user channel assessment algorithm proposed in [13] is redesigned for execution in this paper. The ML channel assessment is obtained using the knowledge of training symbols. Most proposed 3G systems [3] allow for the use of training symbols. When training symbols are not available the a channel can be updated, to track time-variations, using decision feedback from the detector. This approach provides a simple linear channel assessment technique and its properties are similar to those associated with the ML approach discussed in.

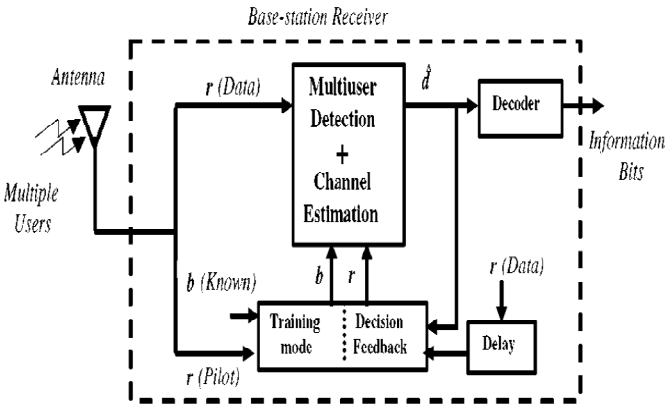

Figure 1. Simplified view of the base station receiver. This figure shows the several user channel assessment and detection blocks in the receiver. A training sequence (pilot) is used for channel assessment and decision feedback is used to update the assessments in the absence of a pilot.

A basic summary of the algorithm and its reckonable aspects are presented here. More details can be found in [13]. L Consider observations of the received vector $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_L$ corresponding to the known training bit vectors . Given the knowledge of the training bits, the discretized received vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L$. $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_L$ are independent and each of them is Gaussian distributed. Thus, the likelihood function becomes

$$p(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_L|\mathbf{A}, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L) = \frac{1}{\pi^{NL}} \exp\left\{-\sum_{i=1}^{L}(\mathbf{r}_i - \mathbf{A}\mathbf{b}_i)^H(\mathbf{r}_i - \mathbf{A}\mathbf{b}_i)\right\}, \quad (1)$$

After eliminating terms that do not affect the maximization, the log likelihood function becomes

$$\left\{\sum_{i=1}^{L}(\mathbf{r}_i - \mathbf{A}\mathbf{b}_i)^H(\mathbf{r}_i - \mathbf{A}\mathbf{b}_i)\right\}. \quad (2)$$

The estimate ,A, that maximizes the log likelihood, satisfies the following

$$\mathbf{R}_{bb}\hat{\mathbf{A}} = \mathbf{R}_{br}. \quad (3)$$

The matrices Rbb and Rbr are defined as follows:

$$\mathbf{R}_{bb} = \sum_{i=1}^{L} \mathbf{b}_i\mathbf{b}_i^H \qquad \mathbf{R}_{br} = \sum_{i=1}^{L} \mathbf{b}_i\mathbf{r}_i^H. \quad (4)$$

Thus, the computations required to obtain the estimate are:
1) the computation of the correlation matrices Rbb and Rbr .
2) the computation required to solve the linear equation in (3).

D. Several user Detection

A several user detection cancels the interference from other users to improve the error rate performance, compared with the traditional single user detection using only a matched filter [20].We implement multistage detection, based on the principle of Parallel Interference Cancellation. This scheme cancels the interference from different users, iteratively in stages and is shown to have computational complexity quadratic with the number of users. It is also possible to feed the channel assessment matrix directly into the multistage detector instead of explicitly extracting the parameters.

The channel matrix A is rearranged into its odd and even columns $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{C}^{N\times K}$ which corresponds to the successive bit vectors $\mathbf{d}_{i-1}$ and $\mathbf{d}_i$; where $\mathbf{d}_i \in \{-1, +1\}^K = [d_{1,i}, \ldots, d_{K,i}]^\top$ are the bits of the users at time instant that need to be detected. In vector form, the received signal is

$$\mathbf{r}_i = [\mathbf{A}_0\mathbf{A}_1]\begin{bmatrix}\mathbf{d}_{i-1}\\ \mathbf{d}_i\end{bmatrix} + \mathbf{n}_i, \quad (5)$$

1) Matched Filter (MF) Detector

The bits $\mathbf{d}_i$ of the K users to be detected lie between the received signal $\mathbf{r}_i$ and $\mathbf{r}_{i-1}$ boundaries. The MF detector [5], [20] does a correlation of the input bits with the received bits. Hence, the MF detector can be represented as

$$\hat{\mathbf{d}}_i = sign\left(\Re\left[\mathbf{A}_1^H\mathbf{r}_{i-1} + \mathbf{A}_0^H\mathbf{r}_i\right]\right). \quad (6)$$

The multistage detector uses the MF to get an initial estimate of the bits and then iteratively subtracts the interference from all other users.

2) Multistage Detector

The multistage detector performs parallel interference cancellation iteratively in stages. The desired user's bits suffers from interference caused by the past or future extending over symbols of different asynchronous users. Detecting a block of bits simultaneously (multi shot detection) can give performance gains [5]. However, in order to do multi shot detection, the above model should be enlarged to include multiple bits. Let us D consider bits at a time $(i = 1, 2, \ldots, D)$. so, we from the multishot recived vector $\mathbf{r} \in \mathbb{R}^{ND}$ $(\mathbf{r}_i, i = 1, 2, \ldots, D)$

$$\mathbf{r} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & 0 & 0 \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & 0 \\ \vdots & \ddots & \ddots & \mathbf{A}_1 \\ 0 & 0 & \mathbf{A}_0 & \mathbf{A}_0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_D \end{bmatrix} + \mathbf{n}_i, \tag{7}$$

Let $\mathcal{A} \in \mathbb{C}^{ND \times KD}$ represent the new multi shot channel matrix. The initial soft decision outputs $\mathbf{y}^{(0)} \in \mathbb{R}^{KD}$: and hard decision outputs $\hat{\mathbf{d}}^{(0)} \in \mathbb{R}^{KD}$ of the detector are obtained from a MF using the channel estimates as

$$\mathbf{y}^{(0)} = \Re\left[\mathcal{A}^H \mathbf{r}\right] \tag{8}$$

$$\hat{\mathbf{d}}^{(0)} = \operatorname{sign}(\mathbf{y}^{(0)}) \tag{9}$$

$$\mathbf{y}^{(\ell)} = \mathbf{y}^{(0)} - \Re\left[\mathcal{A}^H \mathcal{A} - \operatorname{diag}\left(\mathcal{A}^{\mathrm{H}} \mathcal{A}\right)\right] \hat{\mathbf{d}}^{(\ell-1)} \tag{10}$$

$$\hat{\mathbf{d}}^{(\ell)} = \operatorname{sign}(\mathbf{y}^{(\mathbf{l})}) \tag{11}$$

Where $\mathbf{y}^{(\ell)}$ and $\hat{\mathbf{d}}^{(\ell)}$ are the soft and hard decisions, respectively, after each stage of the multistage detector. These computations are iterated for L=1,2……M where M is the maximum number of repetition chosen for desired performance. The structure $\mathcal{A}^H \mathcal{A} \in \mathbb{C}^{KD \times KD}$ is as shows.

$$\begin{bmatrix} \mathbf{A}_0^H \mathbf{A}_0 & \mathbf{A}_0^H \mathbf{A}_1 & 0 & 0 \\ \mathbf{A}_1^H \mathbf{A}_0 & \mathbf{A}_0^H \mathbf{A}_0 + \mathbf{A}_1^H \mathbf{A}_1 & \mathbf{A}_0^H \mathbf{A}_1 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \mathbf{A}_1^H \mathbf{A}_0 & \mathbf{A}_0^H \mathbf{A}_0 + \mathbf{A}_1^H \mathbf{A}_1 \end{bmatrix} \tag{12}$$

The block tri-diagonal nature of the matrix arises due to the hypothesis that the not simultaneous hinder of the different users are coarse synchronized within one symbol duration [13], [21]. If the channel is static, the matrix is also block-To eplitz. We exploit the block tri-diagonal nature of the matrix later, for reducing the intricacy and pipelining the algorithm effectively. The hard decisions, made at the end of the final stage, are fed back to the assessment block in the decision feedback mode for tracking in the absence of the pilot signal. Detectors using differencing methods have been proposed [23] to take advantage of the convergence behavior of the iterations. If there is no sign change of the detected bit in succeeding stages, the difference is zero and this fact is used to reduce the reckoning. However, the advantage is useful only in case of sequential execution of the detection loops, as in DSPs. Hence, we do not implement the differencing scheme in our design for a VLSI architecture.

3. REAL-TIME ALGORITHMS & SEVERAL USER CHANNEL DETECTION

A. Iterative Scheme for Channel Estimation

A direct reckoning of the ML based channel estimate A involves the computation of the correlation matrices Rbb and Rbr then the reckoning of the solution to (3), at the end of the pilot. A direct inversion at the end the pilot by calculation expensive and delays the start of detection beyond the pilot. This delay limits the information rate. In our iterative algorithm, we approximate the ML solution based on the following ideas.

1) The product R_{bb} and R_{br} can be directly approximated using iterative algorithms such as the gradient descent algorithm [16]. This reduces the reckoning intricacy and is applicable in our case because $\mathbf{R}_{bb}$ (as long as $L \geq 2K$) .

2) The iterative algorithm can be modified to update the assessment as the pilot is being received instead of waiting until the end of the pilot. Therefore, the reckoning per bit is reduced by distribution the computation over the entire training duration. During the th bit duration, the channel estimate, A, is updated iteratively in order to get closer to the ML estimate for training length of. Therefore, the channel estimate is available for use in the detector without delay the end of the pilot sequence. The reckonable in the repetitive scheme during the th bit duration are given below.

$$\mathbf{R}_{bb}^{(i)} = \mathbf{R}_{bb}^{(i-1)} + \mathbf{b}_i \mathbf{b}_i^T \qquad (13)$$

$$\mathbf{R}_{br}^{(i)} = \mathbf{R}_{br}^{(i-1)} + \mathbf{b}_i \mathbf{r}_i^H \qquad (14)$$

$$\hat{\mathbf{A}}^{(i)} = \hat{\mathbf{A}}^{(i-1)} - \mu \left(\mathbf{R}_{bb}^{(i)} * \hat{\mathbf{A}}^{(i-1)} - \mathbf{R}_{br}^{(i)} \right). \qquad (15)$$

The terms $(\mathbf{R}_{bb}^{(i)} * \hat{\mathbf{A}}^{(i-1)} - \mathbf{R}_{br}^{(i)})$ in step 3 is the gradient of the probability function in (2) at. $\hat{\mathbf{A}}^{(i-1)}$ for a training length of i The constant μ is the step size along the direction of the gradient. Since the gradient is known exactly, the repetitive channel assessments can be made arbitrarily close to the ML estimate by repeating step 3 and using a value that is lesser than the recipeating step3 and using a value μ that is lesser than the reciprocal of the largest eigen value of $\mathbf{R}_{bb}^{(i)}$. In our simulations, we observe that a single repetition during each bit duration is sufficient in order to reach very close to the true ML estimate by the end of the training sequence. The solution converges in a single unvarying tone to the true estimate with each repetition and the final error is negligible for realistic system parameters. A detailed analysis of the deterministic gradient descent algorithm can be found in [16] and [17] and a similar iterative algorithm for channel estimation for long code CDMA systems is analyzed in [24]. An important advantage of this iterative scheme is that it lends itself to a simple fixed point execute, which was difficult to achieve using the previous conversion scheme based on ML [13]. The multiplication by the convergence parameter can be implemented as a right shift, by making it a power of two as the algorithm converges for a wide range of μ [24]. The proposed repetitive channel assessment can also be easily enlarged to track slowly time-varying channels .During the tracking phase, bit decisions from the several user detector are used to update the channel estimate. Only a few iterations need to be performed for a slowly fading channel and the previous estimate serves as a very good initialization. The correlation matrices a maintained over a sliding window of length *L* as follows.

$$\mathbf{R}_{bb}^{(i)} = \mathbf{R}_{bb}^{(i-1)} + \mathbf{b}_i \mathbf{b}_i^T - \mathbf{b}_{i-L} \mathbf{b}_{i-L}^T \qquad (16)$$

$$\mathbf{R}_{br}^{(i)} = \mathbf{R}_{br}^{(i-1)} + \mathbf{b}_i \mathbf{r}_i^H - \mathbf{b}_{i-L} \mathbf{r}_{i-L}^H. \qquad (17)$$

B. Comparisons

Iterative algorithms have been proposed earlier for channel assessment and detection in [15] and [25]–[28]. In [15] and [25], several iterative methods for general adaptive filter and equalizer applications are discussed in detail. Specific algorithms applicable for CDMA systems are developed in [26]–[29]. Most of these algorithms are based on the method of gradient descent or the method of least squares. These papers

mainly target bit-error rate performance and they do not consider hardware intricacy for a real-time implementation. In this paper, we propose an iterative channel assessment algorithm for multiuser channel estimation suitable for real-time executetion and we show that it has almost the same performance as schemes based on least squares. As discussed in [15], the gradient descent algorithms can be broadly classified into two categories, deterministic and statistics gradient descent. The well know least mean Least mean square algorithm is a statistics gradient algorithm, where the actual gradient is not known and is approximated by an assessment noisy gradient. In this paper, we use the deterministic slope descent algorithm from [15]–[17], where the gradient of the objective function is known exactly, to solve the linear equation in (3). The proposed iterative algorithm to obtain the ML estimate is related to the RLS approach for minimum mean-square-error estimation. In both cases, the assessment for preamble length L aims to minimize the squared error for particular length L. However, we use the known gradient to obtain the estimate as opposed to the RLS algorithm which does not rely on gradient descent. Another difference between our repetitive approach and RLS is that we use a sliding window update as opposed to RLS which uses an exponential weight factor update (λ). For the case of AWGN noise, we note that the ML and MMSE assessment approaches lead to the same solution for obtaining the channel assessment.

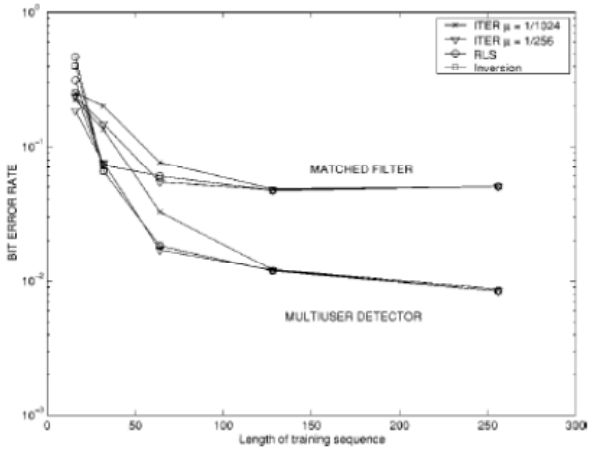

Figure 2. BER performance comparison of the iterative scheme with RLS and true inversion for different preamble lengths. This figure shows the error performance for two detectors, a MF detector and a multiuser detector.

The presence of slow fading at 12 km/h mobile velocity at a carrier frequency of 1.8 GHz. The matrix inversion based scheme assumes a static channel and is not updated with decision feedback, while the iterative scheme is updated every bit. The convergence parameter, μ is chosen as 1/1024. A pilot sequence of 128 bits was used initially to obtain the channel estimates.

A comparison of the performance of our iterative scheme against the RLS algorithm. The simulations were performed for 8 equal power users with a spreading code of length 16 for a AWGN channel having three multipath reflections at 10 dB signal-to-noise ratio. The BER is calculated using the channel assessment after the end of the pilot phase for two types of detectors, a MF detector [5], [20] and a multistage multiuser detector (MUD) [14]. The users are all transmitting at the same power over a static channel with three paths of relative strengths 1, 0.5, and 0.33. Although the detection algorithm can handle the near–far problem, we simulated the equal power scenario as it generates the worst case for multistage detection. To use a sliding window update, we choose λ= 1 as the exponential weighting factor for RLS in our simulations. From Figure 2, it can be seen that our iterative scheme performs almost as well as the RLS algorithm and the actual matrix inversion. The value of should be less than the reciprocal of the largest value. Largest eigen value of $\mathbf{R}_{bb}^{(i)}$ for convergence. Since the maximum eigen value $\mathbf{R}_{bb}^{(i)}$ of increases with i, Since the maximum eigen value of increases with, a larger μ is possible for a smaller preamble length. Therefore, faster convergence can be achieved for smaller preambles. The maximum value μ of that can provide stability for a given preamble can chosen at the receiver for fastest convergence. Therefore, the performance of our iterative algorithm is almost the same as that achieved by the RLS algorithm or the exact ML algorithm. From Fig., we can see that the performance curves almost flatten out after a window length of 128

and, henceforth, we use $L = 128$ as our window length for simulations .Since for this window length $\mu = 1/256$ and $\mu = 1/1024$ have the same performance, we will use $\mu = 1/1024$ simulations for greater stability. Our iterative scheme is less computationally complex than RLS as we avoid the computation of the gain vector with every iteration. The RLS algorithm uses the matrix inversion lemma [15] to avoid matrix inversion but requires scalar division. Though the order of complexity in terms of multiplication and addition is the same for both the iterative scheme and RLS $[O(K^2N)$ [per bit], the RLS scheme requires $O(KN)$ more divisions. The complexity difference may be thought of as the additional intricacy to find a new μ (gain vector) for every iteration in RLS compared with the fixed μ used in our repetitive scheme. Our iterative scheme is also more suitable for a hardware execute than RLS.

A systolic implementation, our proposed iterative algorithm uses only truncated multipliers and adders and does not require any special boundary cells. For implementation of RLS, matrix decomposition techniques such as QR have been used [15]. The QR decomposition can also be execution efficiently in fixed-point using systolic arrays [30], [31]. However, the cells in the array (especially, the boundary cells, which need to compute the Givens rotation) [15], [31] have more computational complexity than the cells used in our iterative algorithm. Thus, we show that our proposed iterative algorithm has a lower computational intricacy than RLS and is also more suitable for a hardware execution. We now evaluate the performance of the iterative scheme with respect to the original ML scheme for different SNRs and for fading channels. The analysis of the system for a multipath fading channel with tracking is as shown in Figure 3. Here, we see that the proposed tracking scheme based on the update of (16) and (17) is able to effectively track the time-varying channel. The poor performance of the static channel hypothesis for this Rayleigh fadin channel (with mobile velocity 10 km/h) at a carrier frequency of 1.8 GHz shows the importance of tracking. The simulation was done for 15 equal power users with a window length of 128 (and preamble length of 128). For faster fading, the window length needs to be decreased appropriately. The original channel assessment scheme requires a matrix inversion and matrix multiplication for every update while the iterative scheme reduces the intricacy to a matrix multiplication per update.

C. Pipelined Detection

The multishot detection scheme [14], [32] proposed in the earlier section is block-based. Such a block-based implementation needs a windowing strategy and has to wait until all the bits needed in the window D are received and are available for computation. This results in taking a window of bits and using it to detect D-2 bits as the edge bits are not detected accurately due to windowing effects. Thus, there are two additional computations per block and per iteration that are not used. The detection is done in blocks and the two edge bits are thrown away and recalculated in the next iteration. However, the stages in the multistage detector can be efficiently pipelined [19] to avoid edge computations and to work on a bit-streaming basis. This is equivalent to the normal detection of a block of infinite length, detected in a simple pipelined fashion. Also, the computations can be reduced to work on smaller matrix sets. This can be done due to the block tri-diagonal nature of the matrix $\hat{\mathcal{A}}^H\hat{\mathcal{A}}$ as seen from (12). The computations performed on the intermediate bits reduce to

$$\mathbf{L} = \Re\left[\hat{\mathbf{A}}_1^H\hat{\mathbf{A}}_0\right] \tag{18}$$

$$\mathbf{C} = \Re\left[\hat{\mathbf{A}}_0^H\hat{\mathbf{A}}_0 + \hat{\mathbf{A}}_1^H\hat{\mathbf{A}}_1 - \mathrm{diag}\left(\hat{\mathbf{A}}_0^H\hat{\mathbf{A}}_0 + \hat{\mathbf{A}}_1^H\hat{\mathbf{A}}_1\right)\right] \tag{19}$$

$$\mathbf{y}_i^{(l)} = \mathbf{y}_i^{(0)} - \mathbf{L}\hat{\mathbf{d}}_{i-1}^{(l-1)} - \mathbf{C}\hat{\mathbf{d}}_i^{(l-1)} - \mathbf{L}^H\hat{\mathbf{d}}_{i+1}^{(l-1)} \tag{20}$$

$$\hat{\mathbf{d}}_i^{(l)} = \mathrm{sign}\left(\mathbf{y}_i^{(l)}\right). \tag{21}$$

Equation (20) may be thought of as subtracting the interference from the past bits of users, who have more delay, and the future bits of the users, who have less delay than the desired user. The left matrix $\mathbf{L} \in \mathbb{R}^{K\times K}$, stands for the partial correlation between the past bits of the interfering users and the desired user, the right matrix, stands for the partial correlation between the future bits of the interfering users and the desired user. The center matrix $\mathbf{C} \in \mathbb{R}^{K\times K}$ is the correlation of the current bits of interfering users and the diagonal elements are made zeros since only the interference from other users, represented by

the non diagonal elements, needs to be canceled. The lower index i, represents time, while the upper index l, represents the iterations. The initial assessments are obtained from the matched filter. Equation (20) is similar to the model chosen for output of the matched filter for several user detection in [32]. Equations (20) and (21) are equivalent to (10) and (11), where the block-based nature of the computations are replaced by bit-stream in computations. The detection can now be pipelined as shown in Figure 4. An example highlighting the calculation of bit 3 in the detector is shown. An initial assessment of the received signal is done using a MF detector, which depends only on the current and the past received bits. The stages of the multiuser detector need bits 2 and 4 of all users to cancel the interference for bit 3. Hence, the first-stage can cancel the interference only after the bits 2 and 4 estimates of the matched filter are available. The other stages have a similar structure. Hence, while bit 3 is being assessment from the final stage, the matched filter is estimating bit 9, the first-stage bit 7 and the second-stage bit5.

Figure 3 Pipelined bit-streaming detection. This figure shows how the detection process can be streamlined to work on a bit basis rather than in blocks. As soon as the immediate future bits are available, the next iteration of detection is carried out. Bit 3 is highlighted as an example for pipelined detection.

Edge bit computations in this scheme and, hence, they can be avoided and we get 2/D savings in computation per detection stage, where D is the detection window length including the edge bits. Also, instead of detecting a block of bits, each bit is detected in a streaming fashion, reducing the worst case latency by the detection window length D/2 and eliminating the memory requirements of block computation by a factor of D^2.

D. Fixed-Point Implementation

A developed a model of the system in C++ using fixed-point "classes" in order to study the performance of the system with different precision requirements. The multiplications and addition operations were "over-loaded" so as to saturate if the available precision were to be exceeded. Since the received signal amplitude depends on the number of users in the system, the number of multiple path reflections, the spreading gain and the SNR(standing noise ratio) the amount of precision required by the A/D converter is given by precision (in bits) =

$$\left\lceil \log_2 \left(K * \sum_{p=1}^{P} \frac{1}{p} + 4 * 10^{-(\mathrm{SNR}/20)} * \frac{\sqrt{N}}{\sqrt{2}} \right) \right\rceil + 4,$$

Equation (22) is due to the fact that the maximum value of the received signal would be $K * \sum_{p=1}^{P} (1/p)$ where K is the number of users and P is the number of multipath reflections. The noise would be less than $4 * \sigma * (\sqrt{N}/\sqrt{2})$ with a probability of more than 0.99, where σ is the variance of the noise and is the spreading gain. Four more bits for additional precision are provided with one bit for the sign. This gives precisions in the range of 8–12 bits for different users and spreading gains which is possible with current A/D converters. We study the effects of finite precision on the estimation and detection algorithms based on their performance using simulations. A detailed analysis of the algorithms for finite precision (as in [33]) is challenging and is not the focus of this paper. We present two simulation results of the algorithms for finite precision with different spreading gains. Figure 5 shows the BER performance of the channel estimation and detection algorithms for a spreading gain of 16 with 8 users. Figure 6 shows the performance for a spreading gain of 32 with 15 users. In each case, we choose a preamble length128 and μ a of 1/1024 [chosen to be smaller than the reciprocal of the largest eigen value $\mathbf{R}_{bb}^{(i)}$ of for all ι in order to ensure convergence].

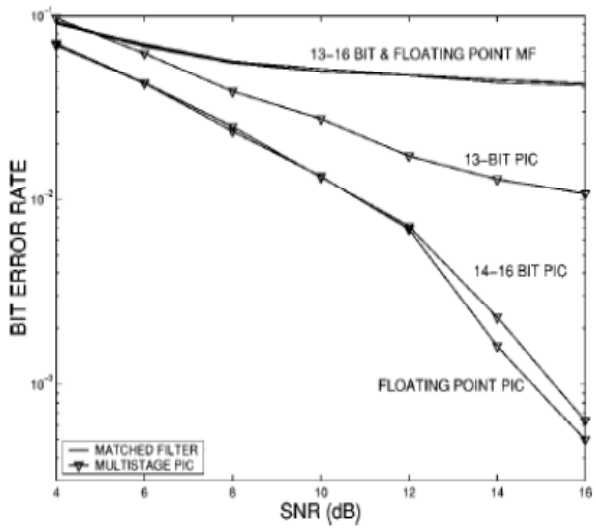

Figure 3. Fixed point error rate performance for N = 16, K = 8. The figure shows the effects of quantization on the MF and MUD for different precisions.

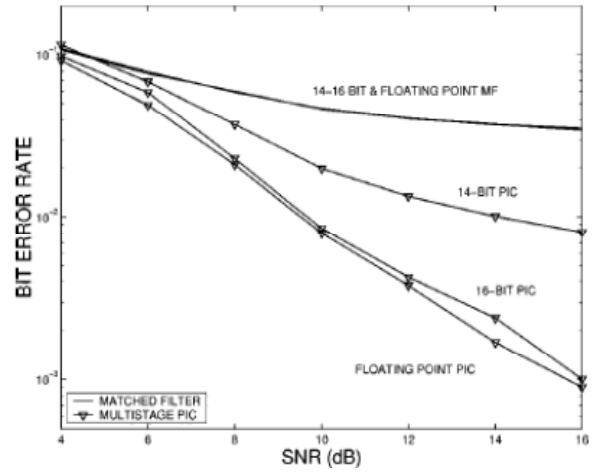

Figure 4. Fixed point error rate performance for N = 32, K = 15. The figure shows the effects of quantization on the MF and MUD for different precisions.

Based on the simulations performed, we have made the following observations:-

1) We see that 16-bit fixed point multiuser channel estimation and detection performs almost as well as floating point precision multiuser estimation and detection. In fact for N= 16 and K= 8 the performance begins to degrade only at 13-bit precision and for N=32 and K=8 the performance degrades at 14-bit precision.
2) The A/D quantization of the received chip-matched filter output does not require as much precision as required for the computations. Reasonable precision of 8–12 bits for A/D conversion is sufficient. For very high SNR, there could be some degradation due to the A/D quantization as the quantization noise could be significant compared with the background noise.
3) The finite accuracy of the reckoning has greater impact on the performance of multiuser algorithms than on single-user algorithms. The matched filter receiver starts degrading only at 8-bit precision. This is reasonable to expect as the computations required for interference cancellation are more complex than that for matched filter detection. While matched filter demodulation requires just an inner product computation, multiuser demodulation requires us to solve a linear equation. Furthermore, significant

performance gain is achieved in multiuser detection (compared with matched filter demodulation) with the extra precision.

4) Higher extend gains and larger number of users implies larger number of multiply-and-accumulates, which may easily filled to capacity the multipliers and adders. Hence, we see that going from N= 16 to N= 32 shows a slight increase in exactness requirements (from 14 to 16).

4. TASK DECOMPOSITION AND VLSI ARCHITECTURES

A. A Decomposition of various user Channel Estimation and Detection

The various sub blocks in the joint various user channel estimation and detection algorithm are as shown in Figure 7. The figure shows the blocks required for channel assessment, the glue matrices L, C between channel estimation and detection and the blocks in the detector. The blocks that are pipelined are shown on the horizontal time axis while the blocks that have coarse-grained parallelism are shown along the vertical axis. The dynamic range of the input is dependent on SNR, the MAI, and the number of users in the system. We assume a 16-bit precision for the architectures. The area and time requirements of the architecture do not vary significantly with the precision. Also note that the blocks r, $\mathbf{R}_{br}$ and $\bar{\mathbf{A}}$ are complex-valued while b and $\mathbf{R}_{bb}$ are real-valued. For the sake of convenience, we henceforth represent the current inputs $\mathbf{b}_i, \mathbf{r}_i$ as b, r and as $\mathbf{b}_{i-L}, \mathbf{r}_{i-L}$, $\mathbf{b}_0, \mathbf{r}_0$ respectively. All the architectures assume a single-cycle multiplication and addition as both multiplication and addition can be implemented in $\log(n)$ type reckoning [34] where n is the number of bits and the single cycle assumption also helps us with the DSP comparisons. We assume that a Wallace or Dadda multiplier tree [34] is used for multiplication requiring $O(n^2)$ 1-bit full adders (FA) for an -bit multiplication. Since the multiplication by in (15) (implemented as a shift) results in truncation of the output, a truncated multiplication using significantly less hardware [35] can be used. The delays of blocks such as multiplexers and gates are assumed to be included in the single-cycle delay. For an area estimate of the architectures, we consider the number of 1-bit FA cells in the design. It can be observed from Figure that the bottlenecks in the pipeline are the matrix multiplications $\mathbf{R}_{bb} * \hat{\mathbf{A}}$ for channel assessment and the calculation L, C of the matrices for multiuser detection. We explore different area-time tradeoffs to develop real-time architectures with minimum area overhead. We explain the design in detail for a time-constrained architecture which shows the upper bound on data rates with no constraints on hardware and then show that by constraining hardware, we are able to design different architectures to meet real-time requirements with minimum area overhead. We have considered only the computational complexity for our analysis and have ignored the analysis of the memory requirements. This is because the focus of

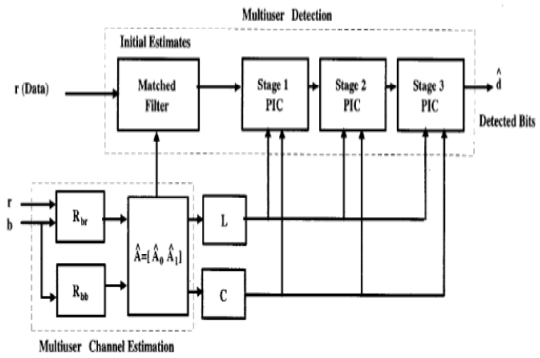

Figure 5. Task decomposition of joint multiuser channel assessment

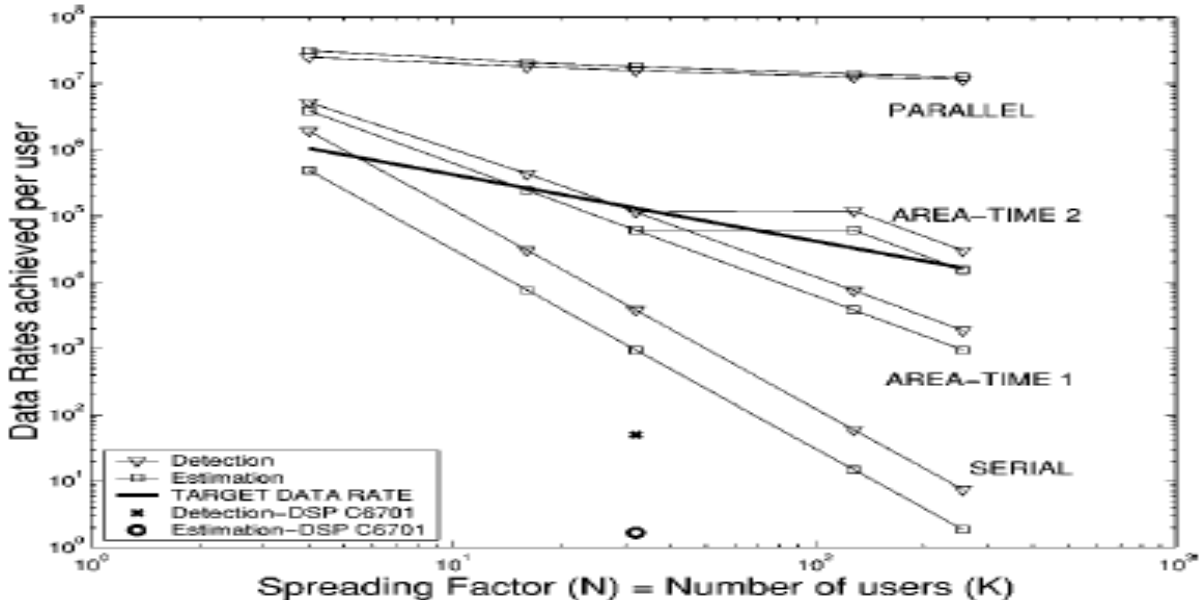

Figure 6. Data rates achieved by different VLSI architecture tradeoffs with varying spreading gains. The figure shows the data rates achieved for a serial, a parallel and two data rate target our paper was on the computational intricacy and area-time tradeoffs needed to meet real-time requirements. We have done an analysis for the memory requirements in previous work for channel assessment [36]. Figure 6 shows the achievable data rates and Figure 7 shows the transistor count for the architectures discussed below. We assume 28 transistors per 1-bit standard FA cell as in [34].

B. Area-Time Tradeoffs for Channel Estimation Architectures

1) Time-Constrained Architecture*:* The block diagram of a time-constrained architecture is as shown in Figure 6. In this architecture, the available parallelism in the algorithm is exploited to the maximum extent. Hence, all the elements needed to perform a parallel matrix multiplication are computed simultaneously. The entire matrices $\bar{\mathbf{R}}_{bb}$ and $\mathring{\mathbf{A}}$ are multiplied using an array of multipliers. The entire product matrix is subtracted by the autocorrelation matrix, shifted and a new channel estimate is formed. Thus, as the time taken by the other computations is pipelined with the time for the multiplication, the output matrix can be formed in parallel every $\log_2(2K)+1$ using multipliers. This is because each element of an $N * N$ product matrix can be computed in $\log_2(N)+1$ time using multipliers and using $4K^2N$ a tree structure to compute the inner products [37], in a time-constrained architecture. We also exploit the bit-level arithmetic and parallel structure of the correlation matrices to form the correlation matrices simultaneously within a cycle. Since the autocorrelation matrix update is a symmetric matrix and all the diagonal elements are 1s $(\overline{a \oplus a} = 1)$, we need to compute only the strictly upper triangular (or lower triangular) part of the autocorrelation matrix. Also, as the updates are all +1s or -1s, this can be obtained from a simple single-bit XNOR gate structure. As the autocorrelation matrix is always updated and down-dated by $\bar{\pm}$ 1s, increment/ decrement counters can be used in place of general adders in our design. Also, the elements in the cross-correlation update are +r or -r and hence, the vector could be directly added or subtracted with every column of the cross-correlation matrix based on the sign of the bit vector b.

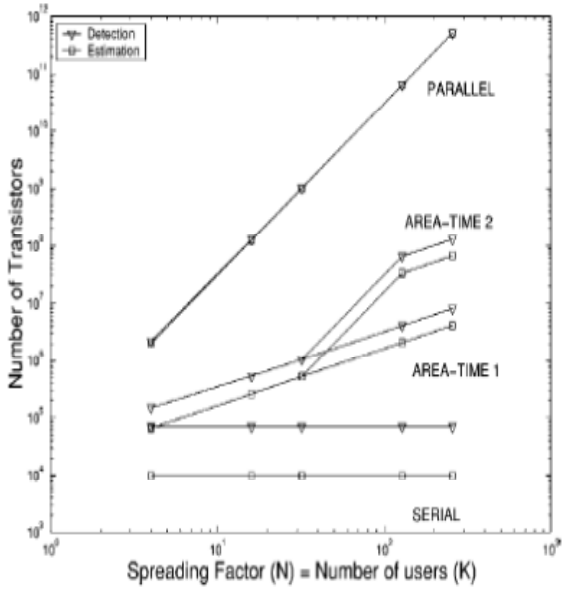

Figure 7. Area requirements for different VLSI architecture tradeoffs with varying spreading gains. Ashows the number of transistors required for a serial, a parallel and two data rate tradeoff architectures with 16-bit precision.

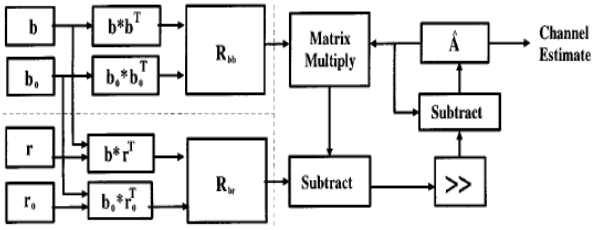

Figure 8. Time-constrained VLSI architecture block diagram. The dotted lines show the parts corresponding to. All operations within a block are being computed by exploiting the maximum parallelism available in that block. The parallel matrix multiplication.

The area requirements for the time-constrained architecture are as shown in Figure 8. The area requirements vary from 10^6 to 10^{12} transistors. This is a highly hostile solution with today's technology and it is not feasible to devote so many FA cells just for channel estimation, which is only a part of the complete receiver. However, this states the theoretical minimum time requirements by exploiting the available parallelism as $\log_2(2K) + 1$ which is the time required $2KN(2K–1)$ to do the parallel multiplication and pipelined integration with the other blocks. We require adders for doing the recurrent doubling [37] in $\log_2(2K)$ time [adding $2K$ elements in $\log_2(2K)$ time requires $(2K–1)$adders] and $2KN$ adders for the subtraction following the multiplication. The data rates achieved by this fully parallel architecture is shown in Figure 8. We can see that we are able to get one to two orders of magnitude performance more than necessary using the amount of parallelism in the algorithms. Therefore, we propose better area-time tradeoffs more closely matched to the target data rates in Section IV B3.

2) Area-Constrained Architecture: For an area-constrained architecture, we assume that only a single multiplier and adder are available. Thus, the matrix–matrix multiplication serially takes $4K^2N$ cycles. The data rates achieved and area requirements for this architecture are shown in Figs. We see that though the serial architecture uses very little area, it falls below real-time requirements by one to two orders of magnitude.

3) Data Rate Targeted Area-Time Tradeoffs*:* In this section, we use part of the available correspondence to achieve real-time performance with minimum area overhead. We use a vector multiplier calculating each row of the multiplication in parallel. This is shown in Figs. as *AREA-TIME1*. Thus, the multiplication now takes $2KN$ cycles at an $2K$ increase in the number of multipliers. This seems to meet real-time requirements up to

N= 32 as seen in Figure 8. However, for N>32, it can be seen that greater amounts of parallelism need to be used to meet real-time. For N>32, we found that additionally 16 columns of the matrix need to be computed in parallel. This implies that the matrix multiplication is done in KN/ 8 cycles and at a further 16 increase in the number of multipliers. This is shown in Figures8 as *AREA-TIME2*.

C. Area-Time Tradeoffs for Multiuser Detection Architectures

1) Time-Constrained Architecture*:* A detailed task divider of the blocks for multiuser detection are as shown in Fig. The blocks consist of a MF detector which provides the initial hard (d) A soft assessment (y) to the parallel interference cancellation stages. A three-stage detector is chosen for execute as it provides sufficient convergence [23]. An array of parallel multipliers is used for computing the entire matched filter assessment $\hat{\mathbf{A}}_0^H \mathbf{r}$ and $\hat{\mathbf{A}}_1^H \mathbf{r}$ vectors in parallel. As the imaginary parts of the products need not be computed, this requires 4KN multipliers. To form the inner product addition in parallel for every row of $\hat{\mathbf{A}}_0^H$ and $\hat{\mathbf{A}}_1^H$ we use an adder tree utilizing $K(2N-1)$ adders. The matched filter estimate can be computed in $\log_2(N)+2$ time. The glue matrices L,C, between the channel assessment and detection schemes require a significant amount of computation. Since $\hat{\mathbf{A}}_0^H \hat{\mathbf{A}}_0$ and $\hat{\mathbf{A}}_1^H \hat{\mathbf{A}}_1$ are symmetric and their diagonal elements and imaginary parts need not be computed to get the matrix products in a time-constrained architecture, we require $2K(K-1)N$ multipliers and $K(K-1)(4N-1)/2$ adders to find the dot products in a tree fashion. This requires $\log_2(K)+3$ time. Similarly, for the computation of $\hat{\mathbf{A}}_1^H \hat{\mathbf{A}}_0$, we require $2K^2N$ multipliers and $K^2(2N-1)$ adders in time. Each stage of the multiuser detector uses only adders as multiplication by single bits can be reduced to addition and subtraction. In order to form the various vectors such as Cd_i in (20), an adder tree of $2K-1$ adders. Thus, for computing Ld_{i-1}, Cd_i and $L^H d_{i+1}$ we need $3(2K-1)$ adders followed by $3K$ more adders (for a four-operand tree addition) to get the final soft decisions y. Each stage of the multistage detector can be computed in $\log_2(K)+3$ time, assuming two cycles for the final four-operand addition and a single cycle for the multiplication. The achieved data rates and area requirements for detection are also shown in Figures. The detector architecture also takes $10^6 - 10^{12}$ transistors, which is not an efficient solution with today's technology but serves to reveal the parallelism and pipelining in the algorithm and determine the maximum data rate.

2) Area-Constrained Architecture*:* For an area-constrained architecture, we use a single multiplier for the L one for C and one for the matched filter. The latency time depends on the matrix– matrix multiplications for the L matrix, which takes K^2N cycles. The data rates achieved and area requirements for the area constrained architectures are shown in Figures.

3) Data Rate Targeted Area-Time Tradeoffs: The area-time intricacy for multiuser detection is found to be similar to that for channel assessment and hence, we use the same type of area-time tradeoffs as before. This is shown in Figures 8 and 9 as *AREA-TIME1*. Thus, the multiplication now takes KN cycles at a increase in the number of multipliers. This potentially can meet real-time requirements up to N=32 as observed in Fig. However, for N>32, it can be seen that greater amounts of parallelism need to be used to meet real-time. Hence, for N>32, we found that 16 columns of the matrix also needs to be computed in parallel. This implies that the multiplication is done in KN/16 cycles and at a further 16 times increase in the number of multipliers. This is shown in Figs as *AREA-TIME2.*

5. RESULTS AND COMPARISONS

A. Computational Savings

The computational advantages of the newly proposed schemes over the previous schemes are shown in Table I. The original algorithm for channel assessment required a matrix inversion and a matrix multiplication requiring $O(6K3+4K^2N)$ cycles on a sequential uni-processor machine such as a DSP while estimation using the iterative method requires only a matrix multiplication $O(4K^2N)$ on a sequential machine. As N and K are of the same order, this only implies a savings of the order of two times. However, a fully parallel VLSI solution for implementation can accelerate the time requirements to $O(\log_2(2K)+1)$Similarly, for comparing the demodulation schemes, we assume that a window of D bits need to be detected. For every window, we save $O(2MK^2)$ computations, assuming an -stage detector as the edge bits do not need to be calculated. A fully pipelined time-constrained detector can reduce the time requirements to $O(\log_2(N)+3)$ by exploiting available parallelism. Note that the enhanced algorithms, as seen from Table I do not have inherent computational savings but are designed to benefit from exploiting parallelism and pipelining in an

architecture. Thus, significant benefits in performance can be achieved by enhancing the existing schemes for channel assessment and detection with schemes having an efficient hardware implementation and exploiting the available parallelism.

Table I. Comparisons of computational time savings. This table shows the computational savings achieved by the enhanced schemes for multiuser estimation and detection over the previous schemes. K—number of users, N—spreading gain, D—detection window, M—number of stages.

Blocks	Architecture	Original (Cycles)	Enhanced (Cycles)
Channel	Uni-processor	$O(6K^3 + 4K^2N)$	$O(4K^2N)$
Estimation	Parallel	-	$O(\log_2(2K) + 1)$
Multiuser	Uni-processor	$O(DNK + M(D+2)K^2)$	$O(DNK + MDK^2)$
Detection	Parallel	-	$O(\log_2(N) + 3)$

B. Comparisons With DSPs

Though DSPs and general purpose processors with MMX enhanced instruction sets exploit byte-wide parallelism, they are inefficient for processing on bits. Storage of bits as bytes on such processors is inefficient as there is a large overhead involved in packing and unpacking these bits. Also, the compiler may not replace bit-level multiplications with additions and subtractions. Using a control structure instead, also limits the utilization of available parallelism. Formation of bit-level matrix updates is much more effective and simpler to build in parallel with XNOR gates than as sequential multiplications on DSPs. Figure 8 also compares the VLSI architectures at 500 MHz with the single processor DSP implementation of the multiuser channel assessment and demodulation algorithms on a TI C6701 floating-point DSP at 167 MHz. We did the DSP analysis in an earlier work [38] and hence, have comparison points only for the N=K= 32 case. The channel assessment DSP implementation takes 600 ms for all 32 users. This poor performance is due to the computation of a matrix multiplication per received bit on the DSP. The frequency of updates to the channel assessment can be reduced for slow fading channels for better time performance. Similarly, detection takes 20 ms for all 32 users. The low data rate performance of the detector is because we consider a more realistic and complete system with continuous updating of channel estimates to the detector as compared with a static channel assumption and neglecting effects of channel estimation in other detector DSP implementations [6], [23].

6. CONCLUSION

We first present computationally efficient algorithms to meet real-time requirements of multiuser channel estimation and detection in future wireless base stations. Existing algorithms for multiuser channel estimation and detection are redesigned from an implementation perspective for a reduced intricacy solution. The ML based channel assessment algorithm requiring matrix inversions, block-based computations and floating point accuracy is redesigned for an iterative scheme, which has a simpler fixed point VLSI architecture and reduced intricacy. Multi user detection is also redesigned for a pipelined structure, that reduces the memory requirements by a factor of D^2 and worst case latency by D/2. The edge bit computations in the block scheme are eliminated 2/D a improvement in computational intricacy per detection stage is achieved. We then present fixed point, real-time VLSI architectures for several user channel assessment and demodulation. The proposed VLSI architecture schemes can be integrated with DSP architectures as a coprocessor support to build single DSP base-station solutions. Bit-level extensions can also be similarly developed to utilize bit-level parallel on DSPs a accelerate wireless communication algorithm.

REFERENCES

[1] G. Xu and J.R. Cavallaro, "*Real-time implementation of multistage algorithm for next generation wideband CDMA systems*", in Proc. Advanced Signal Processing Algorithms, Architectures, and Implementations IX, SPIE, vol. 3807, Denver, CO, July 1998.

[2] G.Woodward, P.B. Rapajic, and B.S. Vucetic, "*Adaptive algorithms for asynchronous DS-CDMA receivers*", in Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC), vol. 2, Taipei, Taiwan, Oct. 1998.
[3] R.D. Gitlin and S.B. Weinstein, "On the required tap-weight precision for digitally implemented, adaptive, mean-squared equalizers", *Bell Syst. Tech. J.*, vol. 58 Feb. 1979.
[4] N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1992.
[5] A. Gatherer, T. Stetzler, M. McMahan, and E. Auslander, "DSP-based architectures for mobile communications: Past, present and future", *IEEE Commun. Mag.*, vol. 38 Jan. 2001.
[6] K.J. Raghunath and K.K. Parhi, "Finite-precision error analysis of QRD-RLS and STAR-RLS adaptive filters", *IEEE Trans. Signal Processing*, vol. 45, May 1998.
[7] T. Long and N.R. Shanbhag, "*Low-power CDMA multiuser receiver architectures*", in Proc. IEEE Workshop on Signal Processing Systems, Taipei, Taiwan, 2000.
[8] Third Generation Partnership Project.. [Online]. Available http://www.3gpp.org
[9] E. Ertin, U. Mitra, and S. Siwamogsatham, "*Iterative techniques for DS/CDMA multipath channel estimation*", in Proc. Allerton Conf. Communication, Control and Computing, Monticello, IL, Oct.1999.
[10] S. Moshavi, "Multi-user detection forDS-CDMA communications", *IEEE Mag.*, Sept. 1996.
[11] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels", *IEEE Trans. Inform. Theory*, vol. 32, pp. 85–96, 1986.
[12] E. Ertin, U. Mitra, and S. Siwamogsatham, "*Iterative techniques for CDMA multipath channel estimation*", in Proc. Allerton Conf. Communication, Control and Computing, Monticello, IL, Sept. 1999, pp.672–681.
[13] P.S.R. Diniz and M.G. Siqueira, "Fixed-point error analysis of the QR-recursive least square algorithm", *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing*, vol. 42, pp. 334–348, May 1995.
[14] P.S.R. Diniz "Fixed-point error analysis of the QR-recursive least square algorithm", *IEEE Trans. Circuits Syst. II: Digital Signal Processing*, vol. 42, pp. 234–248, May 1995.
[15] Information capacity and power control in single-cell communication in IEEE, June 1992.
[16] Admission contol schemes for wirless communication networks with adjustable transmitter power IEEE INFOCOM 94.
[17] R.Horn and C.Johnson, Matrix Analysis Cambridge university press 1988.
[18] P. Black Link Budget of CDMA-2000 1Xev-do wireless Internet access system", InProc.13th IEEE International symposium on personal, Indoor and mobile radio communication-2002, Vol.3.
[19] J. Walrand, "fair pir to pir window based congestion control", IEEE networking
[20] Data throughput of CDMA – HDR a high data rate personal communication wirless system, may 2000, pp-1855-1858.
[21] 3rd Generation partnership project 2(3Gpp2). CDMA 2000 high rate packet data Interfernce Specification Technical report C.S 0024.A.V2.0 July 2006.
[22] Optimal resource allocation in multi-service CDMA Network."IEEE Trans. Wirless communication.
[23] Vehicular technology coference 2007 VTC 2006, IEEE 63 (volume).
[24] Wirless network Evolation 3G upper Saddle River, prentice hall 2003.
[25] Wirless network a Optical communication, Evolation 3G upper, prentice hall 2002.
[26] Note on effective bandwidth in Stochastics Networks. Theory and application.I.Ziedins EDS. Oxford university press 2001.
[27] Berouz A Forouzan, "Data Communication Gigabit Ethernet Handbook", McGraw Hill vol-3.
[28] IEEE 802.3-2007; part3: Carrier Sense Multiple Access with Collision Detection Acess Method and Physical layer Specification.
[29] IEEE 802.3an proposed in November 2010 BASE Is the latest proposed.
[30] M Allman, S. Dawson, W. Fenner, J. Griner, I. Heavens, K. Lahey, J. Semke, B. Volz, "Known TCP Implementation Problems.
[31] Q.O. Snell, A.R. Mikler, and J.L Gustafson"Net-PIPE: Network Protocol Independent Performance Evaluator", Ames Laborator Scalable ComputingLab.

BIOGRAPHIES OF AUTHORS

Prof. Nitish Meena have been done his Graduation in B.TECH from Rajasthan techanical university, Kota (RTU). Which is Affiliated from AICTE, New Delhi & post Graduation in M.TECH from (S.U), Rajasthan, Affilited From AICTE& U.G.C After that he is pursuing of Doctrate (Ph.D) Department of Eletronics & Communcation. In Pratap University, Jaipur (Rajasthan). He have three year teaching Experience in his field & recently Works as a vice-principal in SIET Engineering college, Lalsot (Jaipur) and continuously researching in his field.

Dr. Nilesh Parihar had been done his Graduation in B.TECH from Bangalor techanical university, (Tumkur).which is & post Graduation in M.TECH from M.B.M Engineering college, J.N.V University, Jodhpur Rajasthan, affilited From AICTE. After that he is Doctrate (Ph.D) Department of Eletronics & Communcation. In Jodhpur National University, Jodhpur (Rajasthan). He have Fifteen year teaching Experience in his field & recently Works as a principal in Bhartiya Institute of Engineering & Technology, sikar (Rajasthan) And continuously researching in his field.