A BIT-STREAMING, PIPELINED MULTIUSER DETECTOR FOR WIRELESS COMMUNICATION RECEIVERS

Sridhar Rajagopal and Joseph R. Cavallaro

Center for Multimedia Communication
Department of Electrical and Computer Engineering MS-366
Rice University
6100 Main St., Houston, TX 77005-1892.
{sridhar,cavallar}@rice.edu

ABSTRACT

This paper presents a bit-streaming, pipelined and reduced complexity architecture to meet real-time requirements for asynchronous multiuser detection in wireless communication CDMA receivers. Typically, asynchronous multiuser detection involves multishot detection, which involves block-based computations and matrix inversions. Hence, iterative based suboptimal schemes have been studied to decrease the computational complexity and eliminate the need for matrix inversions. However, we show that such low-complexity schemes can have an added advantage of avoiding multishot detection if they start from a matched filter estimate. The stages of the iteration can be pipelined and bits processed in a streaming fashion. We show that such an implementation scheme reduces the latency of the bits by the detection window length D and eliminates the storage requirements for block computation, which helps in DSP implementations. We also avoid edge-bit computation effects, which reduces the computation by $2/D$ per detection stage. This scheme also results in a simple, bit-streaming and pipelined architecture. DSP simulations show that data rates of 800 Kbps for a single user to 50 Kbps for 32 users can be processed in real-time with additional FPGAs in a pipelined fashion for a spreading gain of 31, giving atleast a 4X speedup over a single DSP implementation.

1. INTRODUCTION

Next generation wireless cellular systems [1], are being designed to support extremely high data rates (in Mbps) and Quality-of-Service (QoS) guarantees required for multimedia communication. Wideband Code-Division Multiple Access (W-CDMA) [2] has been chosen as the multiple access protocol to support these features. The existing narrowband CDMA IS-95 standard supports only voice and low-data rates up to 9.6 Kbps and uses single-user detection at the base-station receiver which ignores Multiple Access Interference (MAI) between different users. To achieve improved performance at these high data rates requires the implementation of highly sophisticated and complex multiuser algorithms. These algorithms combat MAI by jointly processing the signals of all users at the base-station receiver. The multiuser algorithms involving matrix multiplications and inversions [3, 4, 5] require block-based computations and floating point accuracy and significantly increase the implementation complexity of the receiver. These algorithms also need multishot detection [3, 4, 6] (*detecting a block of bits*) due to the interference from the future (succeeding) bits of the other users in an asynchronous system. This results in block-based computations where a block of users' bits are computed together. Also, this produces a latency in the detection process and has increased storage requirements. The windowing causes the edge bits to be in error and hence, they are recomputed in the next window.

In [6], a pipelined Cholesky scheme has been proposed to decrease the multishot detection window length for decorrelating detectors, reducing the block-computations and latencies. But, this scheme still suffers due to the need of computing edge bits every iteration and this results in an overhead of 1 bit per detected bit (100% overhead). Sub-optimal multiuser detection schemes based on iterative techniques [3, 7] have been proposed to eliminate the need for matrix inversions in multiuser detectors and to reduce the implementation complexity. In this paper, we present an additional advantage of iterative-based multiuser detection schemes that start from a matched filter estimate. We show that not only can we pipeline the algorithm effectively, but also eliminate the edge computations and produce a bit-streaming, pipelined architecture. A DSP implementation of the pipelined detector is also carried out for evaluating real-time performance potential.

This work was supported in part by Nokia Corporation, Texas Instruments Inc., the Texas Advanced Technology Program under grant 1999-003604-080, and by NSF under grant ANI-9979465.

2. MULTIUSER DETECTION

Multiuser detection cancels the interference from other users to improve the error rate performance, compared to traditional single user detection using only a matched filter [4]. The desired user's bits receive interference from the past or future overlapping symbols of different users along with their current symbols because they are asynchronous. This is as shown in Figure 1. Here, we assume that the delays of the different users are coarse-synchronized within one symbol duration. We show the effects of pipelining on multistage detection [7], based on the principle of Parallel Interference Cancellation (PIC). This scheme cancels the interference from different users, successively in stages and is shown to have computational complexity linear with the number of users.

For an asynchronous system with Binary Phase Shift Keying (BPSK) modulation, the channel estimate can be arranged as

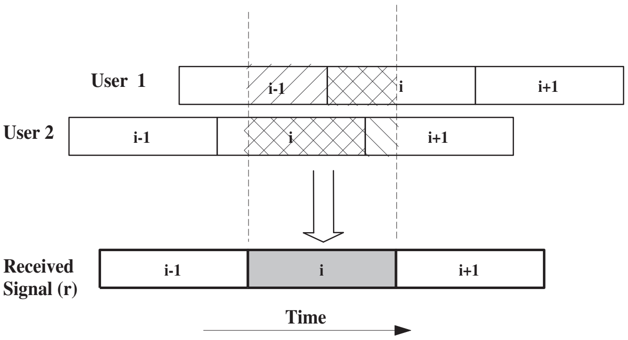

Figure 1: Multiple Access Interference. This figure shows the interference from past, current and future bits (each shaded differently) of other users.

$\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{C}^{N \times K}$ which corresponds to partial correlation information for the successive bit vectors $\mathbf{d}_{i-1}, \mathbf{d}_i \in \{+1, -1\}^K$, which are to be detected. N represents the spreading gain and K represents the number of users in the system. In vector form, the received signal is

$$\mathbf{r}_i = [\mathbf{A}_0 \mathbf{A}_1] \begin{bmatrix} d_{1,i-1} \\ \vdots \\ d_{K,i-1} \\ d_{1,i} \\ \vdots \\ d_{K,i} \end{bmatrix} + \eta_i. \tag{1}$$

2.1. Matched filter 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 matched filter detector [3, 4] does a single user detection for each user, neglecting the effects of other users. The matched filter detector can be represented as

$$\hat{\mathbf{d}}_i = sign(\Re[\mathbf{A}_1^H \mathbf{r}_{i-1} + \mathbf{A}_0^H \mathbf{r}_i]). \tag{2}$$

The multistage detector uses the matched filter to get an initial estimate of the bits and then subtracts the interference from all other users. Note that the matched filter detector is causal and does not require succeeding bits of other users for an estimate. This fact is exploited in pipelining the multistage detector.

2.2. Multishot multistage detection

The multistage detector [7, 8] performs PIC iteratively in stages. To cancel the interference due to the succeeding (future) bits of the users (Figure 1), multishot detection is needed. In order to do multishot detection, the above model should be extended to include multiple bits. Let us consider D bits at a time ($i = 1, 2, \cdots, D$). So, we form the multishot received vector $\mathbf{r} \in \mathbb{R}^{ND}$ by concatenating D vectors ($\mathbf{r}_i, i = 1, 2, \cdots, D$).

$$\mathbf{r} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & 0 & 0 \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \mathbf{A}_0 & \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} d_{1,1} \\ \vdots \\ d_{K,1} \\ \vdots \\ d_{1,D} \\ \vdots \\ d_{K,D} \end{bmatrix} + \eta_i. \tag{3}$$

Let $\mathcal{A} \in \mathbb{C}^{ND \times KD}$ represent the new multishot 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 matched filter using the channel estimates as

$$\begin{aligned} \mathbf{y}^{(0)} &= \Re[\mathcal{A}^H \mathbf{r}], & (4)\\ \hat{\mathbf{d}}^{(0)} &= sign(\mathbf{y}^{(0)}), & (5)\\ \mathbf{y}^{(l)} &= \mathbf{y}^{(0)} - \Re[\mathcal{A}^H \mathcal{A} - diag(\mathcal{A}^H \mathcal{A})]\hat{\mathbf{d}}^{(l-1)}, & (6)\\ \hat{\mathbf{d}}^{(l)} &= sign(\mathbf{y}^{(l)}), & (7) \end{aligned}$$

where $\mathbf{y}^{(l)}$ and $\hat{\mathbf{d}}^{(l)}$ are the soft and hard decisions respectively, after each stage of the multistage detector. These computations are iterated for $l = 1, 2, \cdots, M$ where M is the maximum number of iterations chosen for convergence (typically, 3). The structure of $\mathcal{A}^H \mathcal{A} \in \mathbb{C}^{KD \times KD}$ is as shown:

$$\begin{bmatrix} \mathbf{A}_0^H \mathbf{A}_0 & \mathbf{A}_0^H \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_1^H \mathbf{A}_0 & \mathbf{A}_0^H \mathbf{A}_0 + \mathbf{A}_1^H \mathbf{A}_1 & \mathbf{A}_0^H \mathbf{A}_1 & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_1^H \mathbf{A}_0 & \mathbf{A}_1^H \mathbf{A}_1 \end{bmatrix}. \tag{8}$$

The tri-diagonal block-Toeplitz nature of the matrix arises due to the assumption of the static nature of the channel during the detection window. It is important to note that other multiuser detectors based on matrix inversions [5] lose their block-Toeplitz structure due to matrix inversions and hence, cannot be pipelined. We exploit the tri-diagonal nature of the matrix in the next section, for reducing the complexity and pipelining the algorithm effectively.

The multishot detection scheme [7] proposed is block-based as it requires computation of succeeding incoming bits of users, which are not available. This results in taking a window of D bits and using it to detect (D-2) bits as the edge bits are not detected accurately due to windowing effects. Thus, there are 2 additional

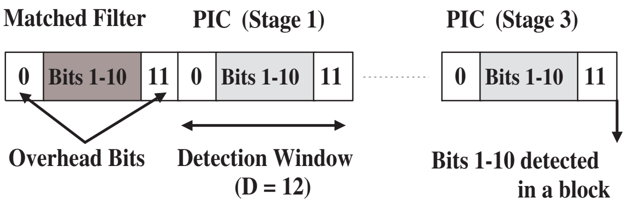

Figure 2: Multishot Detection. This figure illustrates the normally used block-based multishot detection scheme where a block of 12 bits is detected simultaneously and the edge bits are discarded and recomputed in the next iteration.

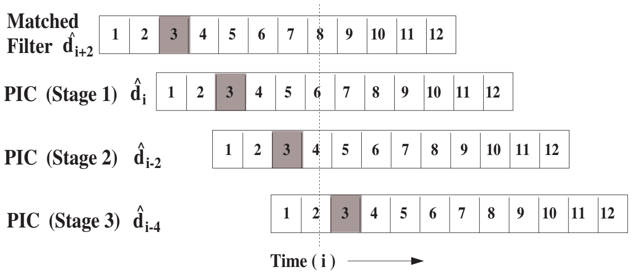

Figure 3: Pipelined Detection. The detection process can be streamlined to work on a bit basis rather than in blocks. As soon as the immediate future bits are available, the next iteration of detection is carried out. Bit 3 is highlighted as an example.

computations per block and per iteration that are not used. Also, such a block-based implementation needs a windowing strategy and has to wait until all the bits needed in the window are received and are available for computation. This is shown in Figure 2 for a detection window of length 12. The figure shows the computation for the matched filter (darkly shaded), followed by 3 stages of PIC (lightly shaded). The edge bits are thrown away and recalculated in the next iteration.

3. PIPELINED MULTISTAGE DETECTION

In this paper, we show that the stages in the multistage detector can be efficiently pipelined 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 are reduced to work on smaller matrix sets due to the block Toeplitz nature of the matrix $\hat{\mathcal{A}}^H\hat{\mathcal{A}}$ in (8). The computations performed on the intermediate bits reduce to

$$\begin{aligned}
\mathbf{L} &= \Re[\hat{\mathbf{A}}_1^H\hat{\mathbf{A}}_0] \\
\mathbf{C} &= \Re[\hat{\mathbf{A}}_0^H\hat{\mathbf{A}}_0 + \hat{\mathbf{A}}_1^H\hat{\mathbf{A}}_1 - diag(\hat{\mathbf{A}}_0^H\hat{\mathbf{A}}_0 + \hat{\mathbf{A}}_1^H\hat{\mathbf{A}}_1)] \\
\mathbf{y}_i^{(l)} &= \mathbf{y}_i^{(0)} - \mathbf{L}\hat{\mathbf{d}}_{i-1}^{(l-1)} - \mathbf{C}\hat{\mathbf{d}}_i^{(l-1)} - \mathbf{L}^T\hat{\mathbf{d}}_{i+1}^{(l-1)} \qquad (9) \\
\hat{\mathbf{d}}_i^{(l)} &= sign({\mathbf{y}_i}^{(l)}). \qquad (10)
\end{aligned}$$

Equation (9) may be thought of subtracting the interference from the past bit 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 $\mathbf{R} = \mathbf{L}^T$, 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 estimates are obtained from the matched filter. The above equation (9) is similar to the model chosen for output of the matched filter for multiuser detection in [9]. The equations (9)-(10) are equivalent to the equations (6)-(7), where the block-based nature of the computations are replaced by bit-streaming computations.

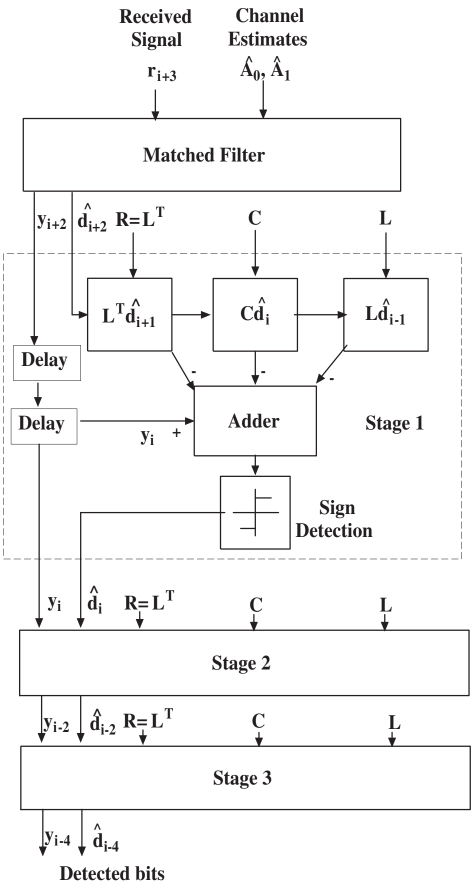

Figure 4: Pipelined multiuser detector architecture. The matched filter provides the initial estimates for PIC. A 3-stage PIC detector is shown. The first stage of the detector is expanded in detail.

The detection can now be pipelined as shown in Figure 3. An example highlighting the calculation of bit 3 in the detector is shown. An initial estimate of the received signal is done using a matched filter 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 estimated from the final stage, the matched filter is estimating bit 9, the first stage bit 7 and the second stage bit 5. By avoiding edge bit computations, we get $2/D$ savings in computation per detection stage, where D is the detection window length. Also, instead of detecting a block of bits, each bit is detected in a streaming fashion, reducing the latency by the detection window length D and eliminating the storage requirements of block computation.

Figure 4 shows a VLSI architecture for the pipelined multiuser detector. The detector takes in the matched filter estimates $\mathbf{y}_i$ and the hard decisions, $\hat{\mathbf{d}}_i$. The partial correlation matrices $\mathbf{L}, \mathbf{C}$ (channel estimates) are also loaded. The detector subtracts the interference iteratively in stages from the matched filter output. Each stage of the detector sends a delayed-by-2 match-filtered estimate and the updated hard decisions to the next stage of the detector.

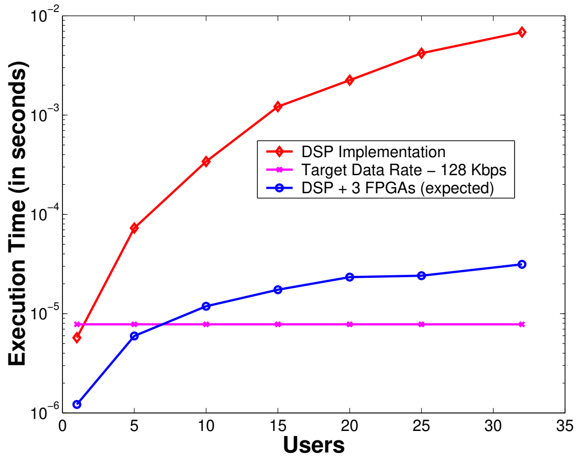

Figure 5: DSP implementation of the pipelined multiuser detector. The figure shows the data rates achieved by a single DSP and the expected data rates using additional FPGAs for pipelining.

4. SIMULATIONS

A software implementation of the pipelined detector is carried out on a TI 'C6701 DSP at 167 MHz as shown in Figure 5. The figure shows the execution time for this pipelined algorithm implemented in software on the DSP for varying number of users. The system was simulated for a Gold Code of length $N = 31$. Both the matched filter (4) and three stages of the multistage detector (9)-(10) were implemented. The top curve in Figure 5 refers to a single processor DSP implementation which achieves only 200 Kbps for a single user to 0.0125 Kbps for 32 users. We have shown [10] that the time taken for each stage of PIC on the DSP ($O(K^2)$) is less than the time taken for the matched filter detector ($O(KN)$). If the pipelining of the multistage detector is carried out in hardware using a high density FPGA per stage with only the matched filter detector (4) implementation on the DSP (bottom curve), we can achieve a performance of 800 Kbps for a single user to 50 Kbps for 32 users. The bottom curve serves as an upper bound to the data rates that can be achieved. The real-time requirements of 128Kbps/user are now satisfied for larger number of users (≤ 7). Thus, the pipelined detection scheme with additional FPGAs provides atleast a 4X speedup and helps meet real-time requirements.

The I/O time between the DSP and the FPGAs is ignored in this plot and it is assumed that the FPGAs can process each stage of PIC faster than matched filtering on the DSP. This is a valid assumption as FPGAs can better exploit the parallelism in the matrix-vector multiplications in the PIC stages using arrays of adder trees, that can be done in $O(\log_2(K))$ time [11]. Note that the multiplications by $\mathbf{d}'s \in \pm 1$ in (9) can be replaced by additions and subtractions on FPGAs and also, bits can be stored more effectively on FPGAs compared to byte-wide storage on DSPs.

5. CONCLUSIONS

We present a bit-streaming, pipelined and low-complexity multiuser detector for wireless communication receivers. Low complexity multiuser detectors have previously been studied to decrease computational complexity. In this paper, we show that they can also be pipelined and a bit-streaming architecture can be developed for asynchronous multiuser detection. Such an implementation scheme reduces the throughput latency by the detection window length D and eliminates the storage requirements for block computation. This can greatly assist in real-time implementations with limited on-chip memory. We also avoid edge-bit computations, which reduce the number of computations by $2/D$ per detection stage. A hardware-software implementation of the pipelined multiuser detection algorithm provides atleast a 4X speedup over a single DSP implementation and helps to meet the 128 Kbps/user requirements of W-CDMA third generation systems.

6. ACKNOWLEDGMENTS

We would like to thank Dr. Behnaam Aazhang for his comments and guidance on this paper.

7. REFERENCES

[1] F. Adachi, M. Sawahashi, and H. Suda, "Wideband DS-CDMA for Next-Generation Mobile Communication Systems," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 56–69, September 1998.

[2] T. Ojanpera and R. Prasad, Eds., *Wideband CDMA for Third Generation Mobile Communications*, Artech House Publishers, 1998.

[3] S. Moshavi, "Multi-User Detection for DS-CDMA Communications," *IEEE Communications Magazine*, pp. 124–136, October 1996.

[4] S. Verdú, *Multiuser Detection*, Cambridge University Press, 1998.

[5] I. Seskar and N. B. Mandayam, "A Software Radio Architecture for Linear Multiuser Detection," *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 5, pp. 814–823, May 1999.

[6] S. Das, J. R. Cavallaro, and B. Aazhang, "Fast Multiuser Detector for a Time-varying CDMA System," in *Proceedings of SPIE - Advanced Signal Processing Algorithms, Architectures, and Implementations VII*, San Diego, CA, July 1997, vol. 3162, pp. 569–580.

[7] M. K. Varanasi and B. Aazhang, "Multistage Detection in Asynchronous Code-Division Multiple -Access Communications," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 509–519, April 1990.

[8] 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.

[9] M. J. Juntti and B. Aazhang, "Finite memory-length multiuser detection for asynchronous CDMA communications," *IEEE Transactions on Communications*, vol. 45, no. 5, pp. 611–622, May 1997.

[10] S. Das, S. Rajagopal, C. Sengupta, and J. R. Cavallaro, "Arithmetic Acceleration Techniques for Wireless Communication Receivers," in 33^{rd} *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 1999, pp. 1469–1474.

[11] S. Rajagopal, S. Bhashyam, J. R. Cavallaro, and B. Aazhang, "Efficient Algorithms and Architectures for Multiuser Channel Estimation and Detection in Wireless Base-station Receivers," *Submitted to IEEE Journal in Selected Areas in Communication (JSAC)*, August 2000.