Real-time implementation of the multistage detector for next generation Wideband CDMA systems

Gang Xu Joseph R. Cavallaro

Electrical and Computer Engineering Department,
Rice University, MS 366, 6100 South Main Street
Houston, TX 77005-1892
(gxu,cavallar)@rice.edu

ABSTRACT

The multistage detection algorithm has been widely accepted 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 implementation of this detection algorithm in the uplink system, where we have achieved both high performance in the interference cancellation and computational efficiency. When interference cancellation converges, the difference of the detection vectors between two consecutive stages is mostly zero. We recode the estimation bits, mapping from ± 1 to 0 and ± 2 . Bypassing all the zero terms saves computations. Multiplication by ± 2 can be easily implemented in hardware as arithmetic shifts. The system delay of a three-stage detector can be reduced by half with satisfactory bit error rate. We also propose a VLSI implementation of this algorithm that has the potential of real-time performance. The detector handling up to eight users with 12-bit fixed point precision was fabricated using a $1.2 \mu\text{m}$ CMOS technology.

Keywords: CDMA, interference cancellation, real-time implementation, reverse link, fixed-point

1. INTRODUCTION

The fast growing cellular telephony industry provides higher and higher capacities for more and more 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 in this paper is based on the short code DS-CDMA scheme.² We place particularly emphasis on the uplink (from mobiles to the base station) system, where all the subscribers share the common channel (shown in Figure 1). In such an environment, the only way to distinguish these users is to use orthogonal or nearly orthogonal codes (or so-called spreading sequences) to modulate the transmitted bits.

Figure 1. System diagram for a multiuser communication system (CDMA uplink physical layer)

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

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

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 each other. It requires a complicated control system on both base stations and mobile phones. Users at the far end of the cell usually consume extremely large amount of power, which would inevitably shorten the battery life.

The basic assumption to simply consider 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 non-linear multiuser detection⁴ shown in Figure ??, 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 Figure 2) with different codes for each user. The channel estimation block in Figure 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, on-line 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 multiuser detector block by using computer arithmetic techniques to reduce the complexity.

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 estimations, 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 the fact that as the iterations progress, the solution becomes more and more invariant, i.e. more and 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 invented a differential matched filter⁸ 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 ASICs. 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 multiple-user binary phase shift keying (BPSK) modulated DS-CDMA synchronous communications 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:

$$r(t) = \sum_{k=1}^K \sqrt{\varepsilon_k} d_k(i) s_k(t) + \eta(t). \quad (1)$$

In Equation 1, K is the number of users. We can get the estimation of the k^{th} 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 k^{th} user, where T is the duration of one bit. A Gold code sequence is used to generate s_k to achieve a better performance than random codes. Finally, AWGN is represented by $\eta(t)$.

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, which is

$$y_i = \frac{1}{T} \int_0^T r(t) s_i(t) dt \quad i = 1, 2, \dots, K. \quad (2)$$

Equation 2 can also be expressed in the following simpler matrix form when we substitute Equation 1,

$$\mathbf{y} = \mathbf{R} \mathbf{A} \mathbf{d} + \boldsymbol{\eta}, \quad (3)$$

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:

$$r_{ij} = \begin{cases} \frac{1}{T} \int_0^T s_i(t) s_j(t) dt & i \neq j \\ 1 & i = j. \end{cases} \quad (4)$$

We can normalize the auto-correlation coefficients in Equation 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, i.e. in Equation 5's format:

$$\mathbf{R} = \mathbf{D} + \mathbf{L} + \mathbf{L}^T, \quad (5)$$

where $\mathbf{D} = \text{diag}(\mathbf{R}) = \mathbf{I}$ and $\mathbf{L}$ is the lower triangular part of matrix $\mathbf{R}$. Since $\mathbf{R}$ is symmetric (shown in Equation 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:

$$\mathbf{A} = \text{diag}\{\sqrt{\varepsilon_1}, \sqrt{\varepsilon_2}, \dots, \sqrt{\varepsilon_K}\}. \quad (6)$$

Our differencing multistage detector is based on estimating the transmitted bits from Equation 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 ($\hat{\mathbf{I}}$ in Equation 7) 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 l^{th} iteration is:

$$\begin{aligned}\mathbf{Az}^{(l)} &= \mathbf{y} - (\mathbf{L} + \mathbf{L}^T)\mathbf{A}\hat{\mathbf{d}}^{(l-1)} \stackrel{\text{def}}{=} \mathbf{y} - \hat{\mathbf{I}}^{(l-1)} \\ \hat{\mathbf{d}}^{(0)} &= \text{sign}(\mathbf{y}) \\ \hat{\mathbf{d}}^{(l-1)} &= \text{sign}(\mathbf{Az}^{(l-1)}).\end{aligned}\tag{7}$$

We have several observations from the above algorithm. After l iterations, it is more and more possible to observe $\hat{\mathbf{d}}^{(l)} = \hat{\mathbf{d}}^{(l-1)}$, which reflects the convergence property. So instead of dealing with each estimated bit vector $\hat{\mathbf{d}}^{(l)}$, as we did in Equation 7, we calculate the difference of the 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. By subtracting the outputs of two consecutive stages represented by Equation 7, we have the following equations (here we denote $\mathbf{B} = \mathbf{L} + \mathbf{L}^T$)

$$\begin{aligned}\mathbf{Az}^{(l)} - \mathbf{Az}^{(l-1)} &= -\mathbf{B}\hat{\mathbf{x}}^{(l-1)} \\ \Rightarrow \mathbf{Az}^{(l)} &= \mathbf{Az}^{(l-1)} - \mathbf{B}\hat{\mathbf{x}}^{(l-1)} \\ \hat{\mathbf{d}}^{(l)} &= \text{sign}(\mathbf{Az}^{(l)}).\end{aligned}\tag{8}$$

Using this differencing algorithm, computations can be saved by computing Equation 8 instead of Equation 7 because more and more elements in the vector $\hat{\mathbf{x}}^{(l)}$ tend to be zero after several iterations. Moreover, all the non-zero terms in $\hat{\mathbf{x}}^{(l)}$ are equal to $+2$ or -2 . The constant multiplication by ± 2 in Equation 8 can be implemented by arithmetic shifts. Therefore, dedicated multipliers are not necessary for this algorithm. Finally, because our manipulation, which subtracts two consecutive stages, is a linear transformation, 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:

```

 $\hat{\mathbf{d}}^{(0)} = \text{sign}(\mathbf{y})$ 
for  $k = 1$  to  $K$                                 /* first stage conventional multistage detection */
     $z_k^{(1)} = y_k - \sum_{j=1}^{j=K} B_{kj} \hat{d}_j^{(0)}$ 
end
 $\hat{\mathbf{d}}^{(1)} = \text{sign}(\mathbf{z}^{(1)})$ 
for  $l = 1$  to  $L$                                 /* second and later stages: differencing multistage detection */
     $\hat{\mathbf{x}}^{(l)} = \hat{\mathbf{d}}^{(l)} - \hat{\mathbf{d}}^{(l-1)}$             /* differencing vector generation */
    for  $k = 1$  to  $K$ 
         $z_k^{(l+1)} = z_k^{(l)} - \sum_{j=1}^{j=K} B_{kj} \hat{x}_j^{(l)}$ 
    end
     $\hat{\mathbf{d}}^{(l+1)} = \text{sign}(\mathbf{z}^{(l+1)})$         /* hard decision generation */
end

```

2.4. Numerical results

The differencing multistage detector is tested by Monte Carlo method with extensive simulations to estimate the convergence rate (shown in Figure 3), where iterations are forced to stop at the eighth stage. 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. This is because the higher SNR, the less noise and lower error rate, which reduces the possibilities for wrong estimations. The other observation is that three stages are enough for most cases, which guides the implementation of this algorithm.

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

We find that 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. Equation 7 and 8 are essentially equivalent to each other. The BER plot versus SNR and MAI in a ten- and twenty-user system is shown in Figure 4. These figures show that the performance of the matched filter degrades dramatically when MAI increases or the number of users increases, which is the near-far and multiple access interference problem. In contrast, the performance of the differencing multistage detector approaches the bound of a single user system, which is given by $P_e = Q(\sqrt{2E_b/N_0})$, for moderate MAI and number of users.

An observation of the percentage of zeros in the differencing vector is illustrated in Figure 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 iteration converges 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 computations we 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 2X speedup in a three-stage system according to Figure 5(b). With more stages in the system, the higher the speedup would be relative to the conventional multistage detector.

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

3. REAL-TIME IMPLEMENTATION

The detector can be implemented in real time by both DSPs and ASICs. Although high performance general purpose DSPs meet the real-time requirement, 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

Generally speaking, converting an algorithm from floating point to fixed point requires two major procedures. One is that we need to estimate the dynamic range of the input data and all the variables used in the algorithm. The other procedure is 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

$$R_r = 2 \times (2^r - 1) \quad (9)$$

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,

$$R_a = 10^{\frac{MAI}{20}}. \quad (10)$$

Figure 6. Dynamic range estimation

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. Since a matched filter treats all the interfering users as noise, the probability density function (PDF) of the matched filter output follows a Gaussian distribution, as illustrated in Figure 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

$$R_m = 2 \times (|\mu| + n\sigma), \quad (11)$$

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 Equations 9 and 10, we can conclude that the number of bits¹⁴ needed to represent the result of matrix product $\mathbf{RA}$ in Equation 3 is

$$\begin{aligned} L_{min} &= \lceil \log_2(R_r \times R_a) \rceil \\ &= 1 + r + \frac{MAI}{20} \log_2 10. \end{aligned} \quad (12)$$

Here we assume a binary representation of the integers. If MAI=10dB 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=10dB case for up to 20 users (shown in Figure 7(a)). In Figure 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 Equation 12, 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

After further investigating this algorithm, we find that the differencing vector $\hat{\mathbf{x}}$ has over 80% zeros after the first iteration in general (see Figure 5), which can be regarded as a sparse vector. When solving Equation 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 are essential to achieve the real-time performance. Such techniques are full carry look-ahead adder, on-line arithmetic unit,⁵ etc.

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

Figure 8. The structure of the differencing multistage detector

3.3. Prototyping the differencing multistage detection algorithm

The structure of the first three stages of the differencing multistage detector is shown in Figure 8. In the first stage, the PIC uses the previous estimations (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

Figure 9. Single ALU implementation of differencing multistage detector

in hardware using a single custom chip. 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 parallel for each user and serial for all users. The timing of inputs and outputs is controlled by a hand shaking mechanism. We assume that the later stage is always ready since we are using the differencing algorithm and workload decreases with each stage due to convergence. The number of cycles needed for an earlier stage must be greater than or equal to that for later stages. 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 logic. The recoder block (highlighted in Figure 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 hand-shaking protocols to the next stage.

3.4. Chip specifications

In Table 1, we summarize 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 8-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=15dB and SNR=6dB 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.¹⁵

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 12.5MHz
Throughput	190kb/s @12.5MHz
Transistor count	6K
Die size	$2.2 \times 2.2mm^2$

Table 1. The chip specification

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

3.5. Cascade mode and system performance

Our chip implements a single stage of the conventional/differencing multistage detector. We can implement a complete multistage detector by simply cascading the chips together. The number of chips that are cascaded depends on the number of the stages in the detector. The flow of data between the chips is controlled by a hand shaking mechanism.

A cascade-mode three-stage differencing multistage detector is shown in Figure 11. Three ASICs are cascaded in a chain, driven by the same clock. The throughput is determined by the slowest stage (which is the first stage obviously) and the delay is governed by all the three stages. From our hardware testing (shown in Figure 10 (b)), the three-stage system delay with the differencing algorithm is less than 100 cycles. Working at a clock rate of 12.5MHz, the system delay is about $5\mu s$, much less than that of the conventional multistage detector, which is around $12\mu s$. Using our design, the system can reach a throughput up to 190kb/s with proper buffering. This rate meets the 144kb/s 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

Figure 11. Cascade mode of three-stage 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.

cells are $900 \times 8 + 30 \times 16 \approx 8kbit$. If a conservative static register cell consists of approximately 10 transistors, we presume that the total number of transistors would be around 100K.

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

Table 2. Scalable ASIC design for the multistage detector

4. CONCLUSION

In this paper, we have focused on the real-time implementation issues for the multistage detection algorithm in Wideband CDMA communication systems. 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 $1.2\mu m$ CMOS technology with a die size of $2.2 \times 2.2mm^2$. Three cascaded chips perform a three-stage multistage detection with a throughput of 190kb/s/user and around $5\mu s$ delay in an eight-user system. The architecture is scalable for a larger design.

5. ACKNOWLEDGMENTS

This work was supported by Nokia Inc., by Texas Instruments, by the Texas Advanced Technology Program under grants 1995-#003604-049 and 1997-#003604-044, and by NSF under grants NCR 9506681 and CDA-9617383.

REFERENCES

1. J.G. Proakis, *Digital Communications*, McGraw-Hill, Inc., New York, NY, 3rd edition, 1995.
2. European Telecommunications Standards Institute, “The ETSI UMTS terrestrial radio access (UTRA ITU-R RTT candidate submission,” <http://www.etsi.org/smg/UTRA/utra.pdf>, May/June 1998.
3. S. Moshavi, “Multi-user detection for DS-CDMA communications,” *IEEE Communications Magazine*, pp. 124–136, Oct. 1996.

4. S. Verdu, *Multiuser Detection*, Cambridge University Press, New York, NY, 1st edition, 1998.
5. M.D. Ercegovic, "On-line arithmetic: an overview," in *Real Time Signal Processing VII, Proceedings of The International Society for Optical Engineering (SPIE)*, 1984, vol. 495, pp. 86–93.
6. M.K. Varanasi and B. Aazhang, "Multistage detection in asynchronous Code-Division Multiple -Access communications," *IEEE Trans. Commun.*, vol. 38, no. 4, pp. 509–519, Apr. 1990.
7. M.K. Simon D. Divsalar and D. Raphaeli, "Improved parallel interference cancellation for CDMA," *IEEE Trans. Commun.*, vol. 46, no. 2, pp. 258–268, Feb. 1998.
8. W.C. Lin, "Differentially matched filter for a spread spectrum system," United States Patent 5,663,983, Sep. 2 1997.
9. K.C. Liu and W.C. Lin, "Pipelined digital differential matched filter FPGA implementation & VLSI design," in *Proc. of the 1996 IEEE Custom Integrated Circuits Conference*, San Diego, CA, USA, May 1996, pp. 75–78.
10. J.K. Hinderling T. Rueth, K. Easton and et. al., "CDMA mobile station modem ASIC," *IEEE J. Solid-State Circuits*, vol. 28, pp. 253–260, Mar. 1993.
11. I. Kang and A.N. Willson Jr., "Low-power Viterbi decoder for CDMA mobile terminals," *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 473–482, Mar. 1998.
12. C. Sengupta S. Das, J. Cavallaro and B. Aazhang, "Fixed point error analysis of multiuser detection and synchronization algorithms for CDMA communication systems," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Seattle, WA, May 1998, vol. 6, pp. 3249–3252.
13. C. Sengupta S. Das and J. Cavallaro, "Hardware design issues for a mobile unit for next generation CDMA systems," in *Advanced Signal Processing Algorithms, Architectures, and Implementations VIII, Proceedings of The International Society for Optical Engineering (SPIE)*, 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 Trans. Circuits and Systems*, vol. 45, no. 11, pp. 1455–64, Nov. 1998.
15. G. Xu, "Implementation issues of multiuser detection in CDMA communication systems," M.S. thesis, Rice University, May 1999.