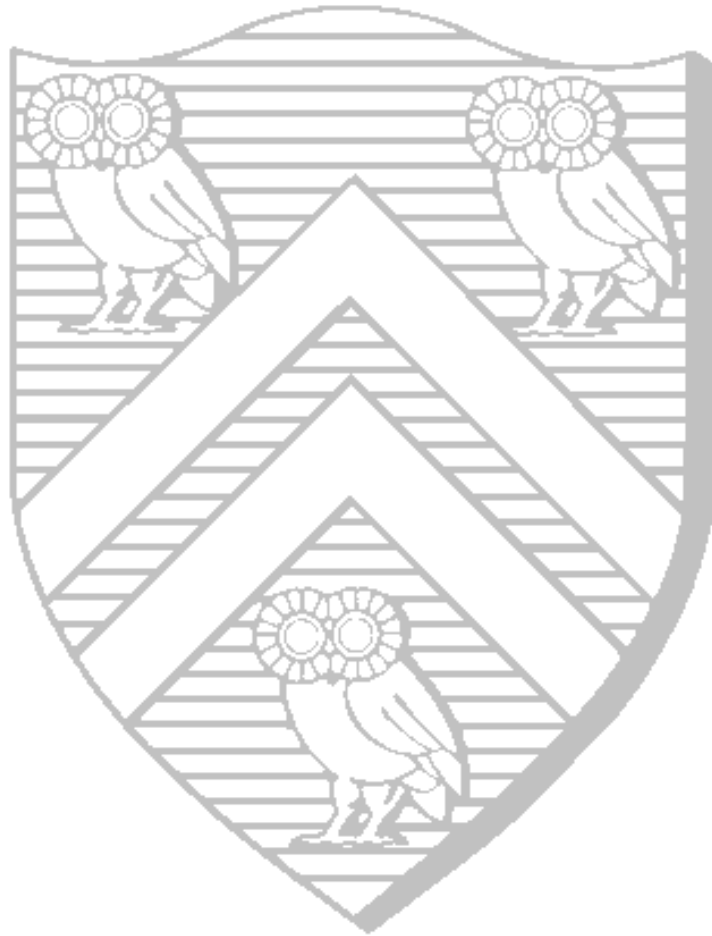

Implementation Issues of Multiuser Detection in CDMA Communication Systems

Gang Xu



Thesis: Master of Science
Electrical and Computer Engineering
Rice University, Houston, Texas (May 1999)

RICE UNIVERSITY

**Implementation Issues of Multiuser Detection
in CDMA Communication Systems**

by

Gang Xu

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

APPROVED, THESIS COMMITTEE:

Joseph R. Cavallaro, Chair
Associate Professor of Electrical and
Computer Engineering

Behnaam Aazhang
Professor of Electrical and Computer
Engineering

Richard Baraniuk
Associate Professor of Electrical and
Computer Engineering.

Houston, Texas

May, 1999

ABSTRACT

Implementation Issues of Multiuser Detection in CDMA Communication Systems

by

Gang Xu

Multistage detectors have been accepted in designs for next generation CDMA base stations because they are less complex than some other multiuser detectors. In this thesis, we propose a differencing method to further reduce complexity. It achieves 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 five-stage detector will be reduced by 75% with a satisfactory bit error rate. We also investigated fixed-point implementation issues and implemented this algorithm in a real-time system using both TI's TMS320C62 DSP and ASICs.

Acknowledgments

I would like to thank Dr. Cavallaro, for his warm encouragement, valuable guidance, constant support and considerate understandings through my entire stay at Rice University. I also thank Dr. Aazhang, for his timely and accurate advice. Thanks also to Dr. Baraniuk, for serving on my committee.

I also wish to thank all members in the CDMA research group, especially Chaitali and Suman, for their enlightenment on the differencing method. Vishwas gave me a lot of good advice and it is always my great pleasure to work with him. Sridhar and Praful contributed their time and energy in the ASIC design for the multistage detector. I greatly appreciate their efforts and advice. I should also give many thanks to Jin and Yeli for their spiritual encouragement and help on both research work and my daily life.

And finally, I want to thank Ziqian, for her love and encouragement. Her smiles make all my struggles worthwhile. Life would be meaningless without her.

The work documented in this thesis has been supported by Nokia Inc. and Texas Instruments.

Contents

Abstract	ii
Acknowledgments	iii
List of Illustrations	vii
List of Tables	ix
1 Introduction	1
1.1 Wireless communications	1
1.2 Addressing the multiuser interference problem: assumptions and conditions	3
1.3 Previous work	4
1.4 Contributions of this thesis	7
1.5 Organization	8
2 Differencing Multistage Detection	9
2.1 Multiuser communication model	9
2.2 Matched filters and cross-correlation matrix	10
2.2.1 Conventional code matched filters	10
2.2.2 Chip matched filter and joint synchronization and detection method	12
2.3 Multistage detection	13
2.4 Derivation of the differencing multistage detector	14
2.5 Convergence analysis	15
2.5.1 Linear Jacobi method analysis	15
2.5.2 Non-linear multistage method	16

2.6	Numerical results	18
2.7	Fixed-point implementation analysis	23
2.7.1	Range estimation	23
2.7.2	Wordlength analysis	25
3	DSP Implementations	28
3.1	Texas Instrument 'C6x DSP	28
3.2	Real-time implementation issues	29
3.2.1	Real-time deadline	29
3.2.2	Optimization methods	30
3.3	The testbed environment	35
3.4	Performance and results	36
4	ASIC Implementations	40
4.1	Prototyping the multistage detection algorithm	40
4.2	Implementation issues of the detector chip	43
4.2.1	Chip specifications	43
4.2.2	Block descriptions	44
4.2.3	Cascade mode	46
4.3	Chip performance	46
4.4	Scalable ASIC design	48
4.5	Comparison of DSP and ASIC implementations	50
5	Conclusion and Future Work	52
5.1	Conclusion	52
5.2	Future work	53
A	Appendix	54
A.1	How to get Figure 1.3	54

Bibliography

Illustrations

1.1	The generation of CDMA transmitted signals	2
1.2	Spreading result with short codes	2
1.3	The degradation factor	5
2.1	System diagram for a multiuser communication system	9
2.2	Percentage of not converged bits in the multistage detection	17
2.3	BER of the differencing multistage detector K=5 and 10	20
2.4	BER of the differencing multistage detector K=15 and 20	21
2.5	Percentage of zeroes in the differencing vector	22
2.6	Flops comparison between differencing and non-differencing method	22
2.7	Performance of joint synchronization and detection	23
2.8	Distribution of the matched filter output	25
2.9	Dynamic range estimation	27
3.1	Conventional matrix-vector multiplication	31
3.2	Sparse matrix-vector multiplication	31
3.3	Software-pipelined loops	33
3.4	The efficiency of the optimization methods	34
3.5	The real-time processing speed of 'C6x DSP	37
3.6	Real-time performance of the multistage detector MAI=0dB.	39
3.7	Real-time performance of the multistage detector MAI=12dB.	39

4.1	The structure of the differencing multistage detector	41
4.2	Single ALU implementation of differencing multistage detector	41
4.3	ASIC chip layout	44
4.4	The architecture of the recoder	45
4.5	Multistage detector by ASICs	47
4.6	The SPICE analysis of the ALU	48
4.7	Chip testing result	49

Tables

3.1	The complexity analysis	36
4.1	The chip specification	43
4.2	Scalable ASIC design for the multistage detector	49
4.3	Comparison between DSP implementation and ASIC implementation	51

To Ziqian,

with love

— Gang

Chapter 1

Introduction

1.1 Wireless communications

Wireless communications have become one of the hottest research areas in the world. The fast growing cellular industry provides higher and higher capacities for more and more subscribers each year. Major companies use low-cost, multi-functional and highly reliable services to expand their market. “Connecting people” is not only a slogan for such companies as Nokia, but also the goal for both research and development of new wireless communication technologies.

After a long discussion about the best method for multiple access, CDMA (Code-Division Multiple Access) has emerged as one of the best multiple access schemes [1]. One of the major reasons is that the first CDMA based standard IS-95 (Interim Standard) for North American cellular communications has been very successful. Some special features of CDMA are capacity increase, improved call quality, enhanced privacy, simplified system planning, improved coverage and increased talk time for mobiles. These benefits lead to the wide acceptance of this standard.

In CDMA communication systems, all the subscribers share the common channel. The only way to distinguish them is to use orthogonal or nearly orthogonal codes (or so-called spreading sequences) to modulate the transmitted bits (Figure 1.1). Figure 1.2 shows an example of the spreading result. The base station uses the knowledge of these codes to detect and estimate each user’s bits.

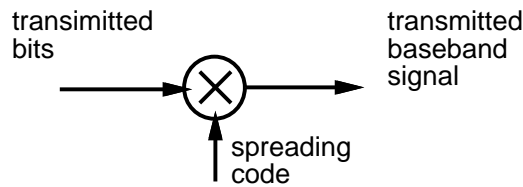


Figure 1.1 : The generation of CDMA transmitted signals

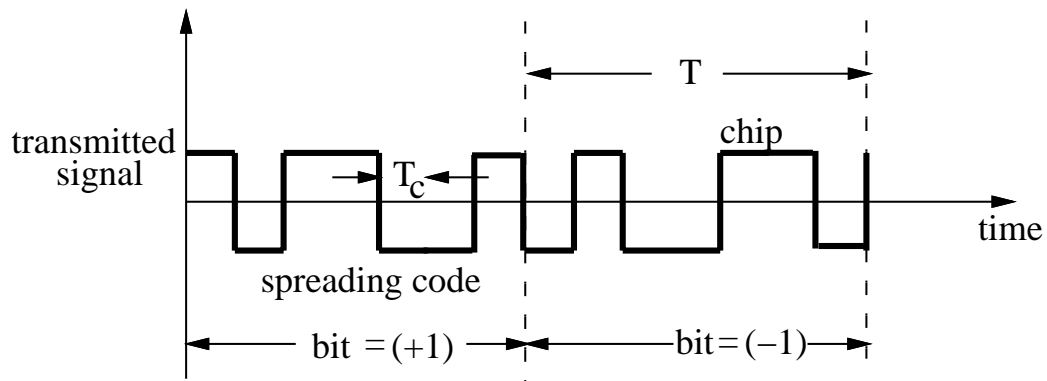


Figure 1.2 : Spreading result with short codes: spreading code = $\{1, -1, 1, 1, -1, -1, 1\}$; chip duration T_c ; bit duration T ; spreading gain 7.

Unlike TDMA (Time-Division Multiple Access) and FDMA (Frequency-Division Multiple Access), where each user is assigned a unique time slot or channel, users in CDMA experience direct interference from the other users. This is called MAI (multiple access interference), which is the major limitation in capacity for the current IS-95 CDMA standard. The other related problem is called the near-far problem [2]. When a user is far from the base station, it is likely that his signal would be overshadowed by the nearer users. 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 far end of the cell usually consume extremely large amount of power,

which would inevitably shorten the battery life or even damage the amplifiers.

In bi-directional CDMA communication systems, transmission from mobile users to the base station is called an uplink and from the base station to mobile users is called a downlink. The uplink problem is a multiple points to one point communication problem, where MAI and near-far problems are the major limitations. The downlink problem, however, is a one point to multiple points broadcasting communication situation, where there are no interfering users in the system. Therefore it no longer has MAI and near-far problems in the downlink.

The focus of most current research is on Wideband CDMA (W-CDMA) or NG (next generation) CDMA. In W-CDMA, the multimedia wireless network will become feasible. Not only voice, but also images, video and data can be transmitted by mobile phones or other portable devices. Achieving a higher data rate and higher capacity are two major goals for W-CDMA, which makes the multiuser interference problem more and more crucial.

1.2 Addressing the multiuser interference problem: assumptions and conditions

The W-CDMA system we researched on is a proposed short-code uplink system [3]. Short code is the spreading code that is repetitive bit after bit, while different from user to user. One case is to use the Gold code, which is one of the best orthogonal code sets ever found. Our research is mostly based on the commonly used Gold code 31 system, where the spreading gain is 31.

Most proposed future W-CDMA systems use BPSK (Binary Phase Shift Keying) modulation for uplink communications. We assumed the channel to be an AWGN

(Additive White Gaussian Noise) channel. If the system only has one user, the bit error rate (BER) versus signal to noise ratio (SNR) is:

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (1.1)$$

where E_b/N_0 is the SNR.

However, if the system contains more than one user, the desired user will treat all the other users as noise. The implementation of this scheme is to use a conventional matched filter, which has been applied to the IS-95 standard. Therefore at this time, the bit error rate for the desired user would be:

$$P_i = Q\left(\sqrt{\frac{E_i}{\frac{N_0}{2} + \sum_{i \neq j} \alpha_{ij} E_j}}\right) \quad (1.2)$$

where α_{ij} is the cross-correlation coefficient between the interfering user j and the desired user i .

In order to measure the negative effect of the interference, a degradation factor is defined by showing how many extra dBs we need to achieve the same bit error rate in the multiple users environment as in a single user system. A sample of a degradation factor is shown in Figure 1.3. Here we assume all the users have the same power and the cross-correlation coefficients are identical for all the users.

From Figure 1.3, we can see that bigger the coefficient is, the higher the interference would be. The other aspect of this figure shows the ways to reduce the degradation factor, that is, either by designing a better spreading code to minimize cross-correlation coefficients, or by removing the interference from the desired user.

1.3 Previous work

As mentioned, simply considering all the other users as noise causes the multiuser interference problem [4, 5]. One viable scheme is to use the cross-correlation informa-

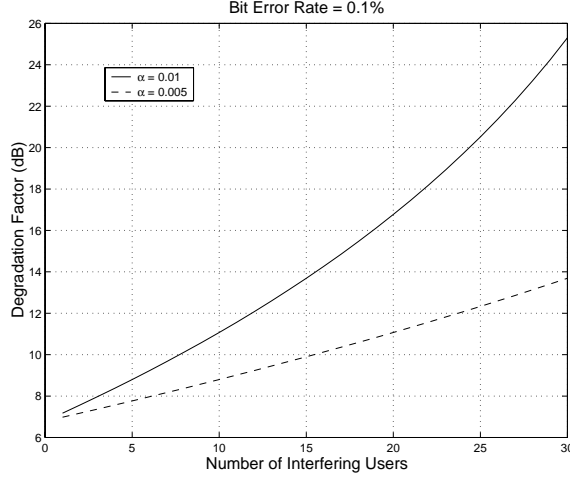


Figure 1.3 : The degradation factor to achieve a bit error rate of 0.1% at $\alpha = 0.01$ and 0.005.

tion of all users to do the multiuser detection or interference cancellation. It requires a short code spreading scheme so that the cross-correlation information is determined.

The optimal multiuser detector proposed by Verdú [6] eliminates the MAI and offers a significant improvement over the conventional detector. The mechanism is to find the maximum-likelihood sequence (MLS) for one user's received signal. However, for a K -user N -bit communication system, it requires 2^{NK} times exhaustive searches to find a maximum likelihood sequence, which is computationally prohibitive.

This led researchers to find sub-optimum multiuser detectors, such as decorrelating detectors and minimum mean-squared error (MMSE) detectors [7–9]. Those detectors need to compute the inverse of the cross-correlation matrix or the matrix which has the same scale [4, 10, 11], the complexity of which is $O(n^3)$. There are some approximative implementation methods, such as [12–14]. They either compromise on the performance or use very complex architecture, leading to high cost. Another branch is adaptive detectors [15–17], which could also be a trend for multiuser detection in the future. In [18], the author discussed the performance of different multiuser

detectors.

The other group of detectors is based upon interference cancellation (IC). The idea is to cancel the interference generated by users other than the desired user. Lower computation demand and hardware related structures are the major advantages of this strategy. This category includes serial interference cancellation (SIC) [19,20] and parallel interference cancellation (PIC). One of the most effective PICs comes from the iterative multistage method, first proposed by Varanasi and Aazhang [21]. 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, come out to be fed into the next stage. The later researchers developed this multistage idea and introduced some other types of PICs [22–26]. Most of them were trying to increase the speed of the convergence and to enhance the performance [27–29]. 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 last several stages, the multistage detector will almost compute from the same input to generate the same output. This is a substantial waste of the computation power and it increases the system delay.

Lin, et al, invented a differential matched filter [30] and gave a FPGA implementation of it [31], which 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.

1.4 Contributions of this thesis

In this thesis, we propose a differencing multistage detection algorithm. Unlike the conventional multistage detector, the number of computations in each stage is not constant. It will decrease 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 detection is preserved.

We have implemented multiuser detectors on Texas Instruments' TMS320C62 and 'C67 DSPs. We used several optimization methods to get the best real-time performance for 'C6x. The implementation shows that by using the differencing multistage detection, the real time performance is 150kb/s/user in a 12-user system for 16-bit fixed-point or single-precision floating-point implementation.

We have also implemented both the conventional multistage detection and the proposed differencing multistage detection in ASICs [32]. [31,33,34] illustrate various kinds of CDMA related matched filter, detector and decoder structures. Compared to their approaches, our design focuses on the multiuser detector for the next generation Wideband CDMA. The chip was fabricated by MOSIS 1.2 μm CMOS technology. The real-time throughput of the system is 190kb/s/user under 12.5 MHz clock rate in an eight synchronous users system. The detection delay is less than 15 μs for a three-stage detector with the same clock rate. We have also investigated the possible scalable design.

1.5 Organization

In the next chapter, we present the mathematical model of the multiuser communication systems and our new differencing multistage detection algorithm. We will also analyze the convergence, joint channel estimation and detection, and fixed-point word length issues. Chapter 3 presents DSP implementations of the multistage detection algorithms and optimization techniques. An ASIC hardware implementation of this algorithm in real-time communications is shown in Chapter 4. We draw our conclusions and project some future work in the last chapter.

Chapter 2

Differencing Multistage Detection

2.1 Multiuser communication model

We assume a K -user binary phase-shift keying (BPSK) modulated DS-CDMA communications system. The channel is a single path channel with additive white Gaussian noise (AWGN). Figure 2.1 shows the structure of the multiuser communication system.

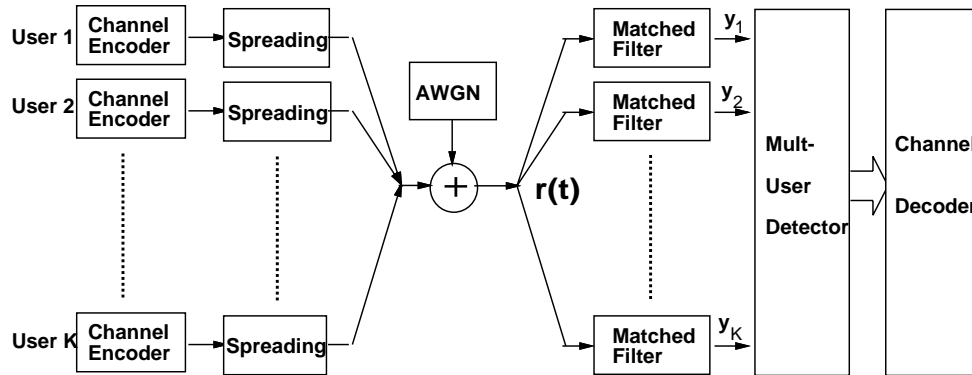


Figure 2.1 : System diagram for a multiuser communication system

At the receiver end, the continuous received signal is given by

$$r(t) = \sum_{k=1}^K \sum_{i=1}^N \sqrt{\varepsilon_k} b_k(i) s_k(t - iT - \tau_k) + \eta(t) \quad (2.1)$$

In equation 2.1, K is the number of users and N is the the detection window size for the multi-shot multiuser detection(multi-bits detection simultaneously). We

can get the estimation of the k th user's signal power $\sqrt{\varepsilon_k}$ by the channel estimation block. The source data bits are represented by $b_k(i)$. Here because we use BPSK modulation, $b_k(i) \in \{-1, +1\}$. s_k is the signature sequence (spreading code) of the k th user, where T is the duration of one bit. In order to get the best performance, s_k is generated by a Gold code sequence. AWGN is represented by $\eta(t)$

2.2 Matched filters and cross-correlation matrix

Matched filter bank is usually the first stage in the baseband signal detection. Almost all modern multiuser detection techniques deal with the output of the matched filter bank and the cross-correlation information of all users in the system. Therefore, we discuss these two topics first and then present the multiuser detection algorithms.

2.2.1 Conventional code matched filters

The conventional code matched filter bank is the major signal detection block in the IS-95 standard. The technique of the matched filter bank is to use one matched filter to detect one user's signal. There are no cross links among the filters. Each branch of the matched filter bank consists of the correlation operation of the received signal with one particular user's signature sequence, which is

$$y_i = \frac{1}{T} \int_0^T r(t) s_i(t) dt$$

$$i = 1, 2, \dots, K \tag{2.2}$$

Equation 2.2 can also be expressed in a simpler matrix format

$$\mathbf{y} = \mathbf{RAd} + \boldsymbol{\eta} \tag{2.3}$$

where vector \mathbf{y} and \mathbf{d} are the output of the matched filter bank and the transmitted user bits respectively. There are NK elements in each vector. In a general asyn-

chronous system, the scale of matrix \mathbf{R} is $NK \times NK$ cross-correlation coefficients.

The elements in the cross-correlation matrix can be represented by:

$$\mathbf{R} = \begin{bmatrix} R(0) & R(1)^T & 0 & \cdots & \cdots \\ R(1) & R(0) & R(1)^T & 0 & \cdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \cdots & 0 & R(1) & R(0) & R(1)^T \\ \cdots & \cdots & 0 & R(1) & R(0) \end{bmatrix} \quad (2.4)$$

$$[R(0)]_{ij} = \begin{cases} \frac{1}{T} \int_0^{2T} s_i(t - \tau_i) s_j(t - \tau_j) dt & i \neq j \\ 1 & i = j \end{cases}$$

$$[R(1)]_{ij} = \frac{1}{T} \int_0^{2T} s_i(t - \tau_i) s_j(t - \tau_j + T) dt \quad (2.5)$$

We do not care the value of auto-correlation coefficients in our multistage detection algorithm, because all the estimated bits are +1 or -1 within the multistage detector (we take only the sign of these bits). The amplitude of each user is not relevant for the final hard decision. Therefore, all the auto-correlation terms are normalized to one. If we need to provide soft decision output for later decoding block, we should also compute the values of the auto-correlation coefficients.

The cross correlation matrix \mathbf{R} can split into three parts, i.e. in equation 2.6 format:

$$\mathbf{R} = \mathbf{D} + \mathbf{L} + \mathbf{L}^T \quad (2.6)$$

where $\mathbf{D} = \mathbf{diag}(\mathbf{R}) = \mathbf{I}$, \mathbf{L} is the lower triangular part of matrix \mathbf{R} . Since \mathbf{R} is symmetric, the upper triangular matrix should be the transpose of the lower triangular matrix.

\mathbf{A} is the amplitude matrix of the signal, which is represented as:

$$\mathbf{A} = \text{diag}\{A(1), A(2), \dots, A(N)\} \quad (2.7)$$

where

$$A(i) = \text{diag}\{\sqrt{\varepsilon_1}, \sqrt{\varepsilon_2}, \dots, \sqrt{\varepsilon_K}\} \quad i = 1, 2, \dots, N \quad (2.8)$$

if $A(i) = A(j)$ for all i, j where $i \neq j$, we call such kind of system time invariant system, otherwise time variant system. Our differencing multistage detector is based on getting non-linear estimated detection bits from linear equations 2.3.

2.2.2 Chip matched filter and joint synchronization and detection method

The newly published methods of joint channel estimation and multiuser detection are widely accepted [35, 36] due to their high performance. In joint channel estimation and detection, we notice that we could use chip matched filter to get the chip matched filter output \mathbf{r}_i .

$$\mathbf{r}_i = \mathcal{U}\mathbf{Z}\mathbf{b}_i + \boldsymbol{\nu}_i \quad (2.9)$$

where, $\mathcal{U} = [\mathcal{U}_1^R \ \mathcal{U}_1^L \ \dots \ \mathcal{U}_k^R \ \mathcal{U}_k^L \ \dots \ \mathcal{U}_K^R \ \mathcal{U}_K^L]$ and $\mathbf{Z} = \text{diag}(\mathbf{z}_1, \mathbf{z}_1, \dots, \mathbf{z}_K, \mathbf{z}_K)$. \mathcal{U} consists of spreading sequence of all the users, delayed by all possible delays. \mathbf{Z} is the composite channel impulse response vector, which includes delay, multipath and multi-sensor information.

The code matched filter output and the cross-correlation matrix are given by expression 2.10:

$$\begin{aligned} \mathbf{y} &= (\mathcal{U}\mathbf{Z})^H \mathbf{r}_i \\ \mathbf{R}\mathbf{A} &= (\mathcal{U}\mathbf{Z})^H (\mathcal{U}\mathbf{Z}) \end{aligned} \quad (2.10)$$

Therefore the multiuser detection, using joint channel estimation and detection scheme, is able to combat multipath fading. The signal model is still valid in equation 2.3.

2.3 Multistage detection

The multistage detector uses basic interference cancellation scheme. In each stage of the multistage detector, PIC parallelly removes the component of other users from the received signal to get a better estimated signal for one particular user. Because we do not know the exact bit information for any user, we use the estimated (hard decision) bits in each stage. The output of the l th iteration is:

$$\begin{aligned} \mathbf{DAz}^{(l)} &= \mathbf{y} - (\mathbf{L} + \mathbf{L}^T)\mathbf{A}\hat{\mathbf{d}}^{(l-1)} \stackrel{\text{def}}{=} \mathbf{y} - \hat{\mathbf{I}}^{(l-1)} & (2.11) \\ \hat{\mathbf{d}}^{(0)} &= \text{sign}(\mathbf{y}) \\ \hat{\mathbf{d}}^{(l-1)} &= \text{sign}(\mathbf{Az}^{(l-1)}) = \text{sign}(\mathbf{z}^{(l-1)}) \end{aligned}$$

Term $\hat{\mathbf{I}}$ is defined as the estimated interference given by the other users to the desired user. Since $\hat{d}_k^{(l)} \in \{-1, +1\}$ and \mathbf{L}, \mathbf{L}^T is pre-calculated, there are not any multiplication operations in equation 2.11. From the assumption made in last section, $\mathbf{D} = \mathbf{I}$. We take hard decisions (sign bit) of the soft detections, therefore the amplitude matrix \mathbf{A} has no impact on the final detection output. However, if the next process after the detection is channel decoding such as Viterbi decoding, soft decisions would be more useful than hard decisions. Therefore, a suitable adjustment of the final output is necessary for such kind of applications. Here we just assume only hard decisions are observed after the detector. Therefore, the multistage detection algorithm is a non-linear algorithm. The following algorithm describes this process. To simplify the notation, here we simply denote $\mathbf{B} = (\mathbf{L} + \mathbf{L}^T)\mathbf{A}$.

```

 $\hat{\mathbf{d}}^{(0)} = \text{sign}(\mathbf{y})$ 
for  $l = 1$  to  $L$ 
  for  $k = 1$  to  $NK$ 
     $z_k^{(l)} = y_k - \sum_{j=1}^{j=NK} B_{ij} \hat{d}_j^{(l-1)}$ 
  end
   $\hat{\mathbf{d}}^{(l)} = \text{sign}(\mathbf{z}^{(l)})$ 
end

```

2.4 Derivation of the differencing multistage detector

From the algorithm described in Section 2.3, we have several observations. After l iterations, it is greatly possible to observe $\hat{\mathbf{d}}^{(l)} = \hat{\mathbf{d}}^{(l-1)}$, which reflects the exact property of the convergence. So instead of dealing with each estimated bit vector $\hat{\mathbf{d}}^{(l)}$, as we did before, we calculate the difference of the bits in two consecutive stages, i.e. the input of each stage becomes $\hat{\mathbf{x}}^{(l)} = \hat{\mathbf{d}}^{(l)} - \hat{\mathbf{d}}^{(l-1)}$ ($j = 1, 2, \dots, K$). $\hat{\mathbf{x}}$ is called differencing vector. By subtracting the outputs of two consecutive stages represented by equation 2.11, we get:

$$\begin{aligned} \mathbf{z}^{(l)} - \mathbf{z}^{(l-1)} &= -\mathbf{B}\hat{\mathbf{x}}^{(l-1)} \\ \Rightarrow \mathbf{z}^{(l)} &= \mathbf{z}^{(l-1)} - \mathbf{B}\hat{\mathbf{x}}^{(l-1)} \end{aligned} \quad (2.12)$$

The updated estimated bit vector $\hat{\mathbf{d}}^{(l)}$, can be worked out by

$$\hat{\mathbf{d}}^{(l)} = \text{sign}(\mathbf{z}^{(l)}) \quad (2.13)$$

Using this differencing algorithm, we are going to save a lot of computations during computing equation 2.12 instead of equation 2.11 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 of $\hat{\mathbf{x}}^{(l)}$ equal to $+2$ or -2 . Such kind of constant multiplication in equation 2.12

can be implemented by arithmetic shifts, which will not introduce any multiplication operations. Further, because our action which subtracts two consecutive stages is a linear transformation, the BER after each stage will not change, compared with the conventional multistage detection. It makes the final BER of the differencing multistage detector be the exact same as the conventional multistage detector.

The complete algorithm is described below:

```

 $\hat{\mathbf{d}}^{(0)} = \text{sign}(\mathbf{y})$ 
for  $k = 1$  to  $NK$ 
   $z_k^{(1)} = y_k - \sum_{j=1}^{j=NK} B_{ij} \hat{d}_j^{(0)}$ 
end
 $\hat{\mathbf{d}}^{(1)} = \text{sign}(\mathbf{z}^{(1)})$ 
for  $l = 1$  to  $L$ 
   $\hat{\mathbf{x}}^{(l)} = \hat{\mathbf{d}}^{(l)} - \hat{\mathbf{d}}^{(l-1)}$ 
  for  $k = 1$  to  $NK$ 
     $z_k^{(l+1)} = z_k^{(l)} - \sum_{j=1}^{j=NK} B_{ij} \hat{x}_j^{(l)}$ 
  end
   $\hat{\mathbf{d}}^{(l+1)} = \text{sign}(\mathbf{z}^{(l+1)})$ 
end

```

2.5 Convergence analysis

2.5.1 Linear Jacobi method analysis

If we did not use the hard decisions in the multistage detector, we would perform a Jacobi iterative method to solve linear equations 2.3. According to [37], the convergence is determined by the spectral radius of the iteration matrix G , which is defined as:

$$\rho(G) = \max\{|\lambda| : \lambda \in \lambda(G)\} \quad (2.14)$$

In equation 2.11, the iterative matrix \mathbf{G} is

$$\mathbf{G} = (\mathbf{DA})^{-1}(\mathbf{L} + \mathbf{L}^T)\mathbf{A} \quad (2.15)$$

$$\rho(\mathbf{G}) \leq \|\mathbf{G}\|_\infty$$

Here since we use linear method, \mathbf{D} is no longer a normalized identity matrix, but a diagonal matrix.

According to the theorem, if \mathbf{RA} is strict diagonal dominant matrix, the spectral radius of \mathbf{G} satisfies the inequality $\rho(\mathbf{G}) < 1$, then the iteration converges for any starting vector.

The other theorem shows if $\mathbf{R} \in R^{n \times n}$ is symmetric and positive definite, then the Jacobi iteration converges for any x . Since it is very easy to show that \mathbf{R} is a symmetric positive definite matrix, we can infer that Jacobi iterative method for this problem will converge eventually.

2.5.2 Non-linear multistage method

However, the multistage and the differencing multistage detector are not linear schemes because they take non-linear hard decisions between stages. Therefore it is not easy to analytically get the convergence property. We conduct an extensive experiment and get some empirical results about converging speed (shown in Figure 2.2), where iterations are forced to stop at the eighth stage. We draw the following three conclusions from these figures.

First of all, the differencing and conventional multistage detector works 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.

Secondly, generally speaking, three stages are enough for most cases. This is a

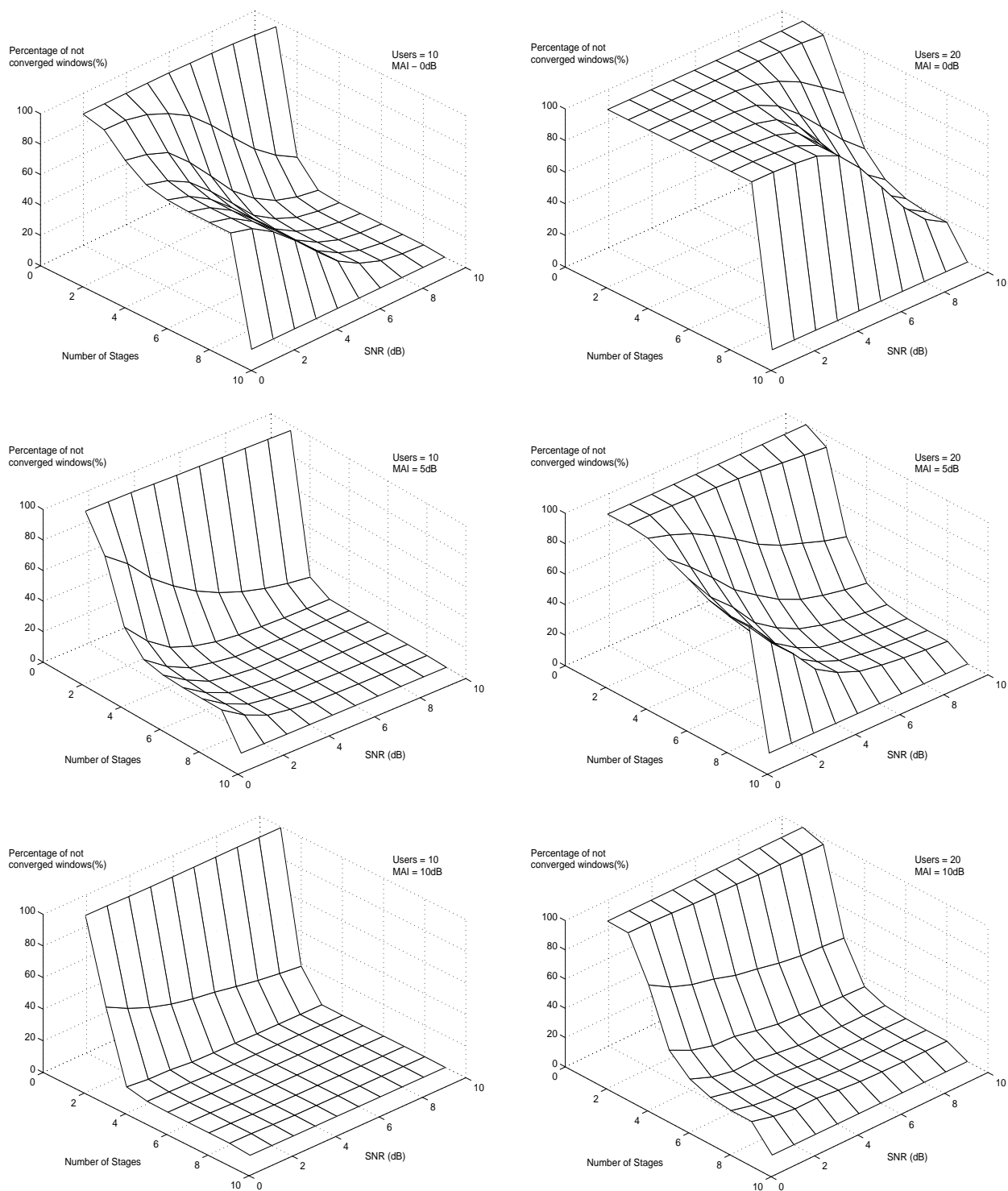


Figure 2.2 : Percentage of not converged bits at $K=10, 20$; MAI: 0, 5, 10dB respectively.

good rule of thumb, which guides the implementation of this algorithm.

Finally, when MAI is small (or users have relatively same power level), the detector's output usually oscillates (or converges slowly). The reason is that at this time it is hard to detect which user is an interfering user because they have almost the same power as the desired user. The other reason is due to the global signal to noise ratio, because we define SNR as the weakest user's energy over noise. In high MAI situation, the global SNR (all the users' total signal to noise ratio) is high, which is better than low MAI situation. This phenomena is the limitation not only for the multistage detector, but also for most of the other multiuser detectors.

2.6 Numerical results

The differencing multistage detector is fully tested by billions of bits. For bit error rate simulation, we check the error rate after getting 400 errors or more than 100,000 bits per simulation, which can guarantee the statistical stability [38]. We find that the bit error rate for the differencing multistage detector is exactly the same as the conventional multistage detector. This is because we do not change the framework of the iterative method, nor the convergence speed. Equation 2.11 and 2.12 are essentially equivalent to each other. The BER versus SNR and MAI in a five-user up to twenty-user system is shown in Figure 2.3 and 2.4. 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. On the contrast, the differencing multistage detector performs constantly along with different MAI and different number of users (for moderate MAI and number of users). So it can be regarded as a near-far resistant algorithm. Moreover its performance approaches a single-user communication system's BER bound, which is given

by $P_e = Q(\sqrt{2E_b/N_0})$, the ideal case of the interference cancellation.

An observation of the percentage of zeros in the differencing vector is illustrated in Figure 2.5. In this figure, we see that the percentage of zeros in the differencing vector increases as the iterations progress, which shows the iteration converges progressively. After the fourth stage, the number of zeros gets to the point of 98% in a 15-user communication system. It explicitly indicates that if we use the conventional multistage detector, almost 98% computation resource is wasted. Figure 2.6 gives us a clear view of how many computations we are going 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 go up linearly. On the contrary, the number of computations in the differencing multistage detector decreases as the iteration proceeds. Thus the overall savings are up to 75% in a five-stage system according to Figure 2.6. And more stages in the system, the higher the speed up it would be in relative to the conventional multistage detector.

If we use joint channel estimation and multiuser detection method, the matrix provided by channel estimation block $\mathcal{U}Z$ has already had the multipath combining function. Therefore, the differencing multistage detector has the capability to combat multipath fading. The input from the new cross-correlation matrix and chip matched filter output are given by equation 2.10. The performance of the joint channel estimation and detection is shown in Figure 2.7. In this figure, we conclude that the multistage detection has better performance than the decorrelating detector, which was proved in [4].

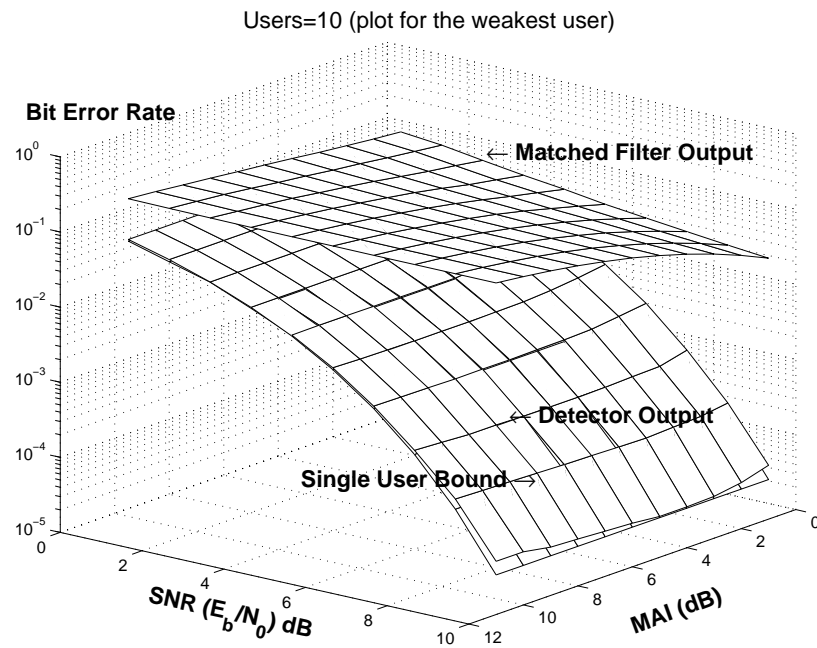
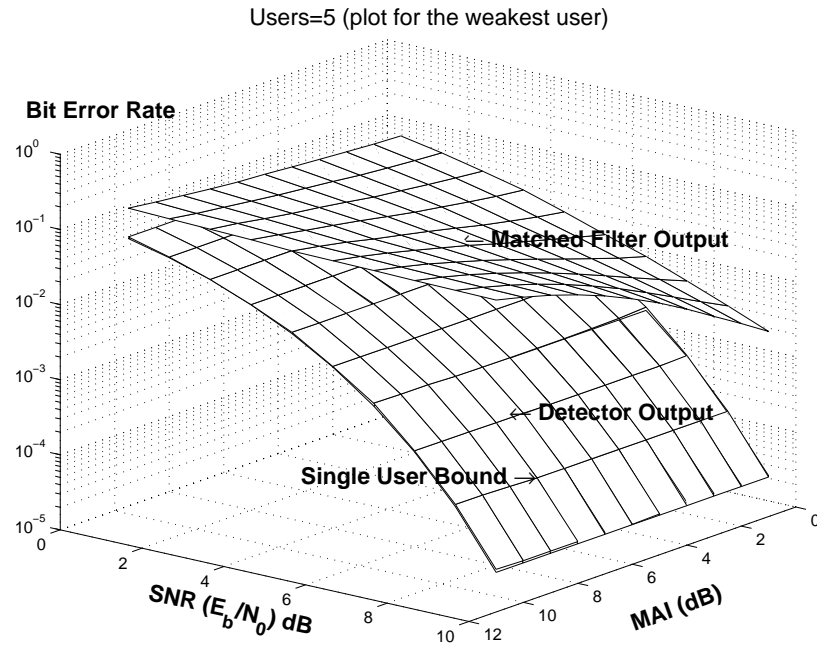


Figure 2.3 : BER of the differencing multistage detector $K=5$ and 10

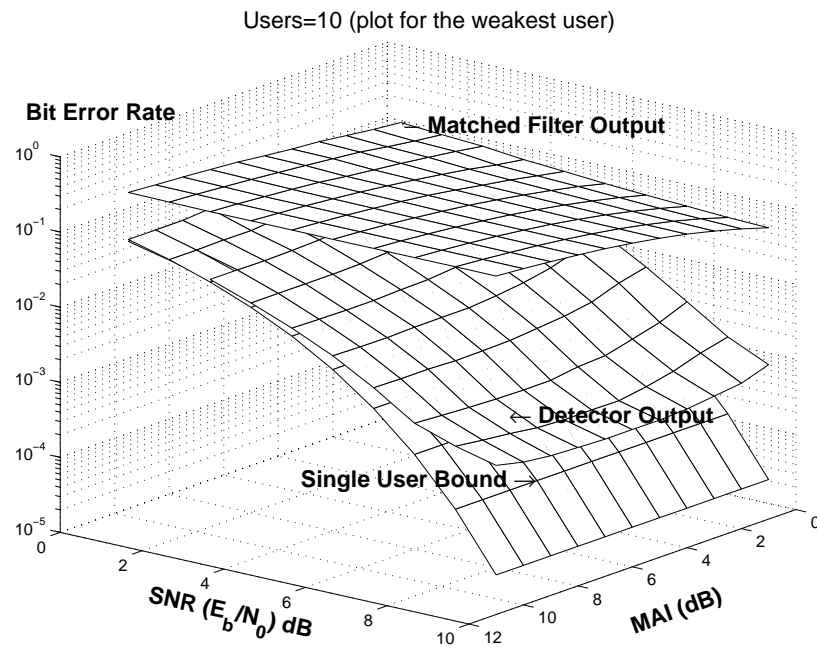
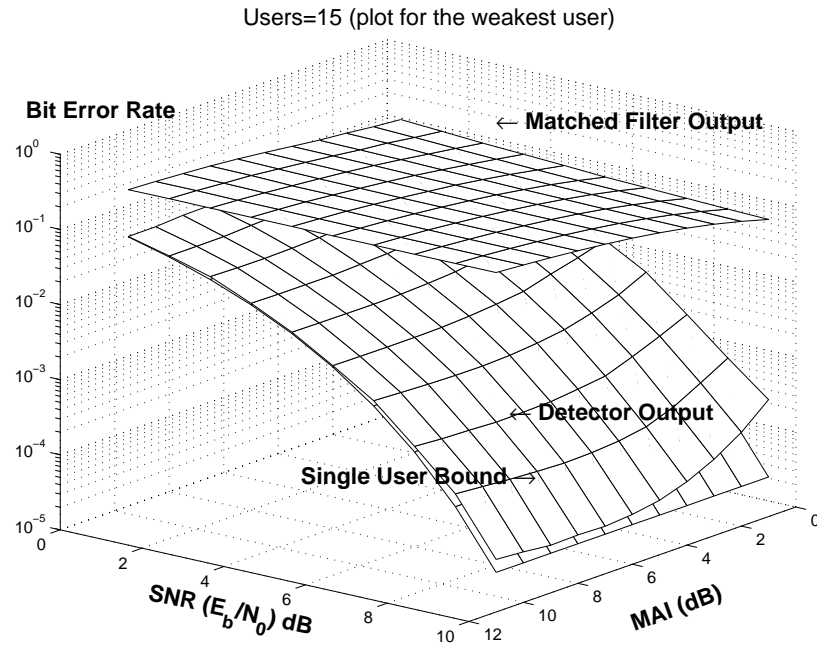


Figure 2.4 : BER of the differencing multistage detector $K=15$ and 20

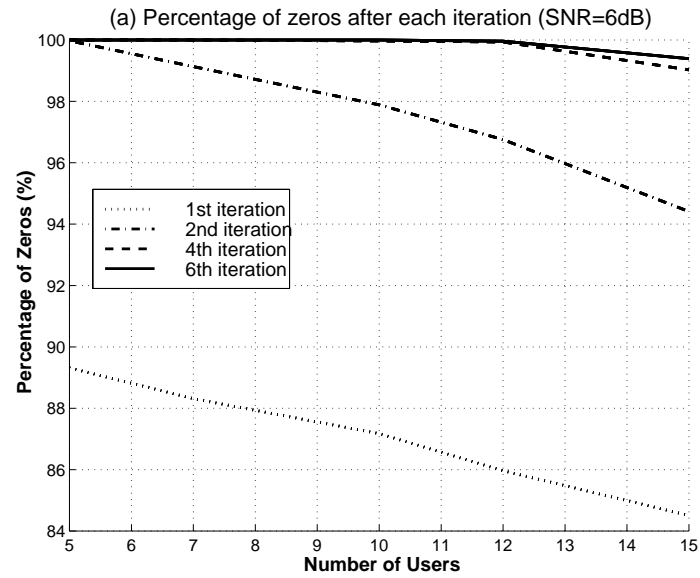


Figure 2.5 : Percentage of zeroes in the differencing vector

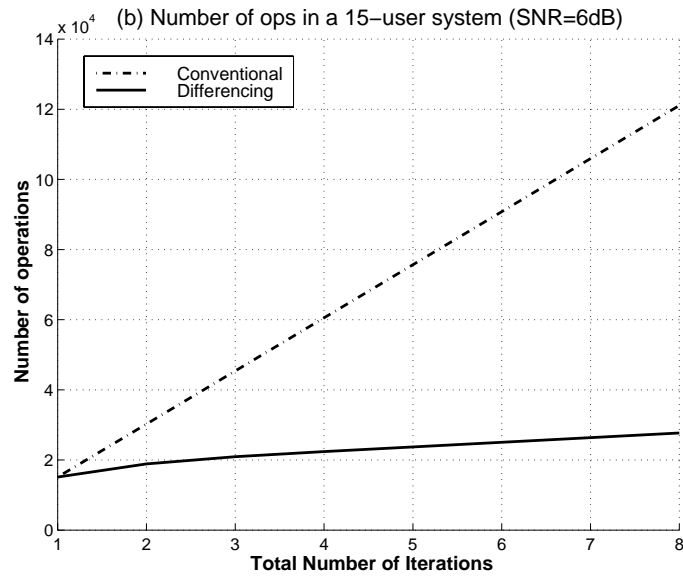


Figure 2.6 : Flops comparison between differencing and non-differencing method

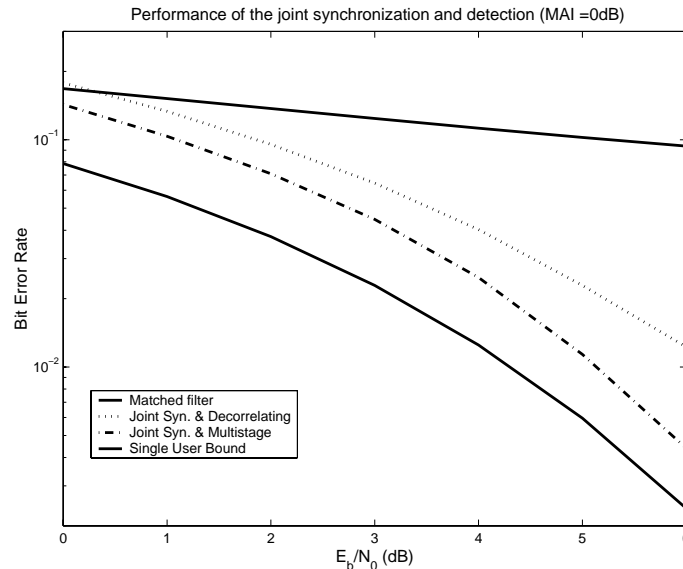


Figure 2.7 : Performance of joint synchronization and detection at MAI=0dB, K=10

2.7 Fixed-point implementation analysis

In order to reduce the cost and increase the speed, the algorithms should be implemented into fixed-point arithmetic finally [39–41]. 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 optimized wordlength to represent numbers and truncate the results. We will show some analysis and simulation result about fixed-point implementation of the differencing multistage detection in this section.

2.7.1 Range estimation

The data involved in differencing multistage detector are cross-correlation coefficients and the matched filter output. The former ones come from local code integrators and

channel estimation block, while the later ones are generated by the integrators. Both of them need A/D (analog to digital) converters to sample and digitize the analog input signals at front end.

From the characteristic 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

$$R_r = 2 \times (2^r - 1), r \in \mathcal{R}^+ \quad (2.16)$$

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

$$R_a = 10^{\frac{MAI}{20}} \quad (2.17)$$

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 Gaussian distribution, as illustrated in Figure 2.8. 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 kind of distribution is estimated as

$$R_m = 2 \times (|\mu| + n\sigma) \quad (2.18)$$

where μ is the mean of one peak and σ is the standard deviation of that peak. n is an empirical constant. For Gaussian distribution, $n = 3$ can guarantee 99.9% of all the samples fall in range R_m [41].

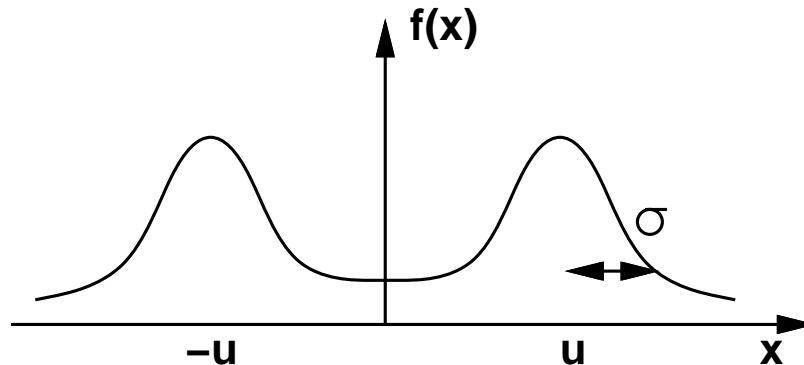


Figure 2.8 : Dynamic range estimation

2.7.2 Wordlength analysis

From equation 2.16 and 2.17, we can conclude that the number of bits needed to represent the result of matrix product \mathbf{RA} in equation 2.3

$$\begin{aligned} L_{min} &= \lceil \log_2(R_r \times R_a) \rceil \\ &= 1 + r + \frac{MAI}{20} \log_2 10 \end{aligned} \quad (2.19)$$

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

For matched filter output, we did some experiments by varying the number of users, MAI and SNR in the system (as shown in Figure 2.9). The number of bits needed is nine in a perfect power control case, ten in a MAI=10dB case for up to 20 users.

From Figure 2.9, we can also observe that if the number of users is small, SNR will dominate the variation of the dynamic range. While more users are involved in the system, MAI will control the number of bits.

For some applications, the optimized word length might not follow the relation in equation 2.19, but be usually smaller than L_{min} . The optimized word length is determined by simulation, in which the minimal mean square distortion is set corresponding to a particular performance requirement. The technique in details is not discussed in this thesis.

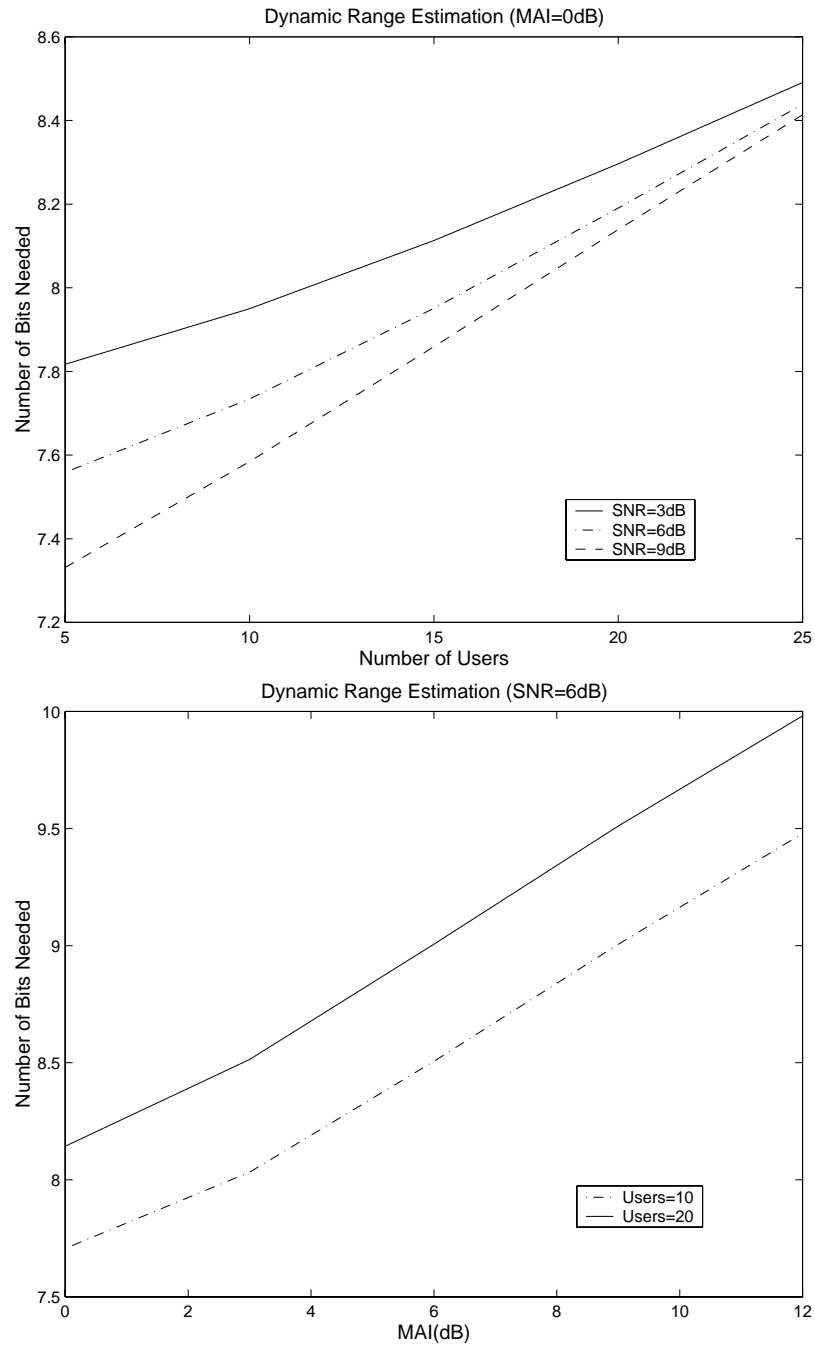


Figure 2.9 : Dynamic range estimation by varying MAI, SNR and K

Chapter 3

DSP Implementations

From this chapter, we start addressing the implementation issues of the multistage detection for CDMA base stations, especially the differencing multistage detection algorithm.

3.1 Texas Instrument 'C6x DSP

DSP implementation has attracted a lot of attention for the cellular market [42]. Compared to general purpose microprocessors, DSP has lower power consumption and special instruction sets, which is suitable for digital filter design, fast transform and real-time applications.

The newly announced Texas Instruments' TMS320C6201 [43] is very suitable for this differencing multistage detection algorithm because it has a powerful VLIW (Very Long Instruction Words) DSP core. Working at 200MHz clock rate, it has totally eight functional units, including 2 multiplication units, 2 data transfer units and 4 general arithmetic units. Ideally it can execute eight instructions per clock cycle, which is about 1600MIPS (million instructions per second). Although it is a fixed-point processor, its 16-bit precision is good enough for the multiuser detection (according to the dynamic range analysis in Section 2.7).

The advantage of the VLIW architecture is that it is a static scheduling processor. Compared to superscalar based microprocessors, it has low complexity. However, the

efficiency of the executable code mostly depends on the efficiency of the compiler. The other drawback is the code length is usually longer than the code for other non-VLIW processors.

Texas Instruments has paid great attention on compilers. Their compiler can do fair amount of optimization work (see next section). With hand coded assembly code, it can achieve around 2MAC (multiply accumulate) per second. Their VLIW architecture is also an improved version, which can execute variable length of packed instructions, which removes redundant NOP (no operation) so that greatly reduces the length of the executable code.

The other newly announced chip in 'C6x family is the floating point DSP 'C67. Its processing power is over 1GFLOPS (giga floating point instructions per second). However, the floating point operations are not as efficient as fixed-point operations, although the precision is higher. We will focus on 'C62 implementation of the differencing multistage detector.

3.2 Real-time implementation issues

3.2.1 Real-time deadline

In order to implement algorithms on 'C6201 DSP in real-time, we need to consider several optimization issues. The major difference between a general performance simulation program and a real-time embedded program is that real-time implementations usually have deadlines for their processing time.

In the third generation wireless communications, the base station should be able to process up to 144 kb/s/user for multiple vehicular users. The lower bound of throughput is 144 kb/s/user. If we implement one base station using one 'C62 (work-

ing at 200MHz), the complexity of all the baseband process for all users should be less than C in equation 3.1.

$$C = \frac{200MHz \times 2MAC/cycle}{144kb/s} \approx 2778MAC/b \quad (3.1)$$

Here 2MAC/s is the upper bound performance for 'C6x DSP in 16-bit fixed point or single precision floating point computations.

3.2.2 Optimization methods

Particularly to 'C6x processors, there are several effective optimization methods.

1. Algorithm level optimization

The original format of the differencing multistage detection algorithm contains two nested loops computing for each stage. Inside these nested loops, we need to do the interference cancellation. The basic operation is multiplying and accumulating. In numerical algebra, we know that there are several levels of computations, which are from the lowest to the highest: scalar-scalar, vector-vector and matrix-matrix. The higher the level is, the higher the speed would be in general. This is because higher level operation is beneficial to low-level optimization and memory access. We see the improvement when we use metrics and vectors instead of scalars in Matlab. The same enhancement can also occur if we use higher level operation. In this particular example, we replace the inner loops by dot products. Our assumption is the diagonal terms of the cross-correlation matrix are all zero, so that we do not need to check the indices.

We also observe the property of the cross-correlation matrix, which is a banded, symmetric matrix as shown in equation 2.4. If the window size is large, most of the elements in the cross correlation matrix are zero. Therefore, in the dot product

operation, we select the range in any row of the cross-correlation matrix very carefully so that only elements within the band (mostly non-zero) are involved in the computation. It will save nearly 75% of total computations if the window size is 12.

The memory access is critical to the efficiency of the implementation. For dot product, it requires row-oriented storage type for matrices, which is the exact type for C. However, after we pick out the non-zero elements in the differencing vector, we do not need to compute the dot product any more. Instead, we may update element by element for those non-zero ones. It is very inefficient because even if there is only one non-zero element, it has to go through the first row to the last to update matrix-vector multiplication result (shown in Figure 3.1).

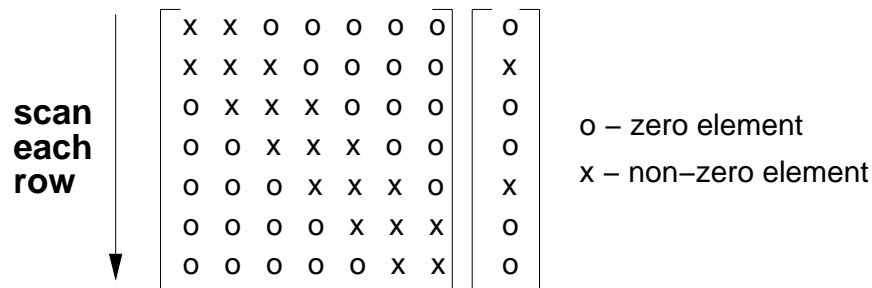


Figure 3.1 : Conventional matrix-vector multiplication: scanning and updating each row from the first to the last

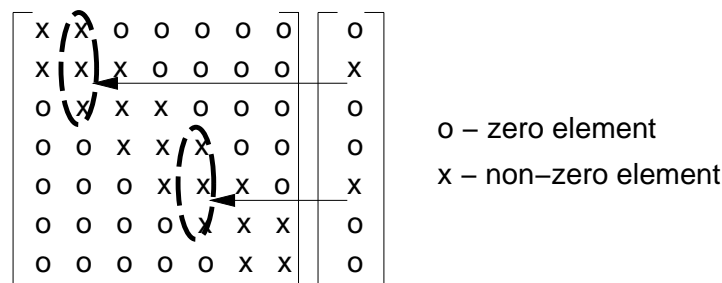


Figure 3.2 : Sparse matrix-vector multiplication: column-oriented updating

Therefore we change the memory access method to the column-oriented update. For each non-zero element in the differencing vector, we find the corresponding column, in which all the elements will be scaled by this non-zero element. Update all of them in one loop and go on to the next non-zero element. It avoids the scanning the whole matrix. This method is illustrated in Figure 3.2.

Since 'C6201 is a fixed-point 16-bit processor, we also consider to port the original floating-point algorithm to fixed point. We implement the algorithm in 16-bit fixed point in order to maximize the precision as well as to make it possible to use all the functional units in one cycle.

2. C compiler optimizer

This is the basic optimization tool. It can allocate registers, simplify expressions, remove redundant loops and use inline calls for some small functions. We could also integrate some intrinsic instructions, such as “_ext” for sign extensive in order to get the sign bit of a number. Those instructions are build-in for 'C6x use only, however they can replace some complicated function calls to get further speedup.

3. Software pipelining

Software pipelining is a major technique to optimize VLIW code. It can schedule instructions from a loop in order to make multiple iterations of this loop execute in a parallel manner. Figure 3.3 illustrates such kind of process. In this example, the loop contains four major actions, namely loading operands, executing, saving results and loop counting. The shaded area shows the maximum overlapping part, which means they can be executed parallelly. Using this pattern, we are able to figure out how to execute four loops together. Since VLIW architecture has eight functional units, in order to maximize the usage of these units, we should make more instructions to be able to run parallelly. To fill in and flush the software pipeline, we need some less

paralleled code, called prolog and epilog. If the number of loop iterations is large enough, their impact is negligible.

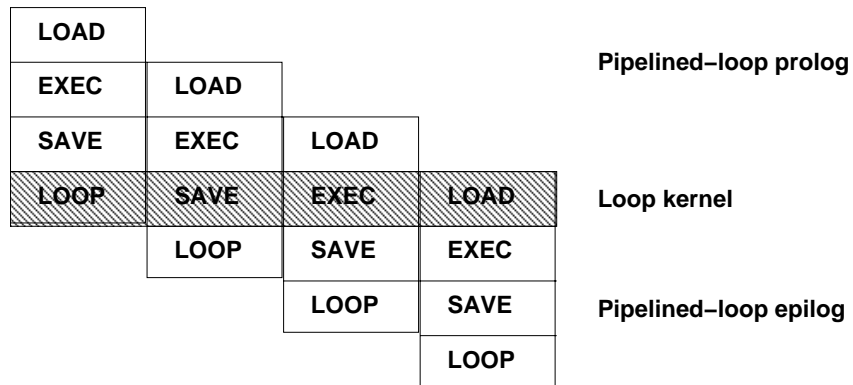


Figure 3.3 : Software-pipelined loops – LOAD: loading operands. EXEC: executing the operations. SAVE: saving results. LOOP: computing loop counter and branching

4. Inline assembly code:

Although C optimizer and software pipelining is good enough in some gentle applications, they can not maximize the performance of 'C6x. The best way to design a real-time embedded program is to use assembly language. However, due to the complexity of the VLIW architecture, prototyping cycle for an algorithm to the corresponding assembly code will be considerably long.

The common rule in computer architecture is to make the common instructions faster. It is not necessary to use assembly language everywhere because the number of critical sections is limited [44]. In the differencing multistage detector, most of the signal processing is done by matrix-vector multiplication. Therefore, if we could speedup this portion of the program, we would be able to get a dramatic overall speedup.

We use an assembly routine for dot products. This routine has a standard interface with C program. It implements a software pipelining, with a maximized usage of total

eight functional units, which reaches the performance of 2 MAC/s.

Figure 3.4 shows how well the above three optimization methods (2-4) work for the particular differencing multistage detection algorithm. The fully optimized code executes 15 times faster than the raw code (unoptimized). We can also see from the assembly code that global register optimization has a rough one functional unit usage per instruction, which means no parallelization. The software pipelining optimization approaches a five functional units usage per instruction ($109.2 \div 20.9$). After inserting inline assembly code, we almost get the best functional units usage rate – eight per instruction ($156.2 \div 20.9$). It also demonstrates that software pipelining and inline assembly code methods are two most effective ways to optimize real-time applications.

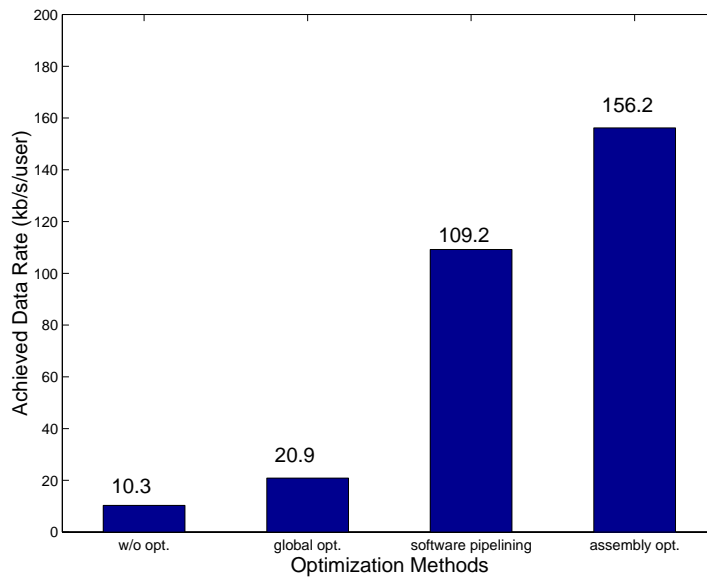


Figure 3.4 : The efficiency of the optimization methods. Optimization methods from left to right are: no optimization, global and local register level optimization (-o3 -pm), software pipelining and inline assembly language optimization.

3.3 The testbed environment

In order to test the real-time performance of 'C6x DSP implementation, we build a simulation testbed, which consists of three different kinds of software: CDMA parameters setup, Code Composer and Matlab.

CDMA parameters window, which is written in Visual Basic 5.0, is a simple GUI to set important CDMA uplink parameters, including the number of users, SNR and MAI.

Code Composer from Texas Instruments is able to control 'C6x DSP board. We set profile points at the beginning and end of the differencing multistage detection function, which allows us to count number of clock cycles for one sliding window detection.

Matlab program displays the histogram (normalized to probability density functions) of the transmitted bits, soft decisions before and after the differencing multistage detector.

The operating system uses a basic synchronization technique. PC will send a message "hello" to DSP. DSP is ready and waiting for this "hello" message. Upon getting it, DSP runs the program and write the result into files. At the same time, a "done" label is set by DSP. While DSP is running, PC waits for this "done" signal. With this signal asserted, Matlab on PC is activated and it displays the result. The communication protocol is using file transfer, where both signal checking and setting are protected by dual locks to guarantee the mutual exclusion. This simple operating system shows a good capability to control this real-time simulation environment.

3.4 Performance and results

After further investigating this algorithm, we find that the differencing vector $\hat{\mathbf{x}}$ has over 80% zeros after the first iteration in general. So it can be regarded as a sparse vector. When working on equation 2.12, instead of $(NK)^2$ times computations, we can deal with the non-zero terms only. In the second iteration for example, the total computations will shrink to $0.2(NK)^2$. With the algorithm level banded matrix optimization mentioned in Section 3.2, we can further save the computation. The following table is based on the assumption that two times (2X) speed can be achieved by the differencing method.

Methods	Total flops	Per user per bit flops
Conventional multistage	$2L(NK)^2$	$2LNK$
Differencing multistage	$2\frac{L}{2}(NK)^2$	LNK
Optimized differencing multistage	$2\frac{L}{2}3K \times NK$	$3LK$

Table 3.1 : The complexity analysis: L: the number of iterations, N: the number of bits per window, K: the number of users

The theoretical result shows that the flops per user per bit is linear to the number of users in the system.

Armed with all the optimization techniques mentioned in Section 3.2, the 'C6201 DSP ensures about $\eta = 2\text{MAC}$ per clock cycle in the kernel [44]. Since 'C6201 works at the clock cycle of $T_c = 5\text{ns}$, the maximum processing speed is about 150kb/s/user for a K=15, L=4 system. This rate is obtained by the relation below:

$$R = \frac{1}{3T_c L K^2 / \eta} = \frac{1}{3 \times 5\text{ns} \times 4 \times 15^2 / 2} \approx 150\text{kb/s/user} \quad (3.2)$$

The proposed W-CDMA has a data rate up to 144kb/s, So even if all the 15 users use the highest data rate, we can still expect that the computation power

in the 'C6201 is enough to handle the future W-CDMA communications using the differencing multistage detector.

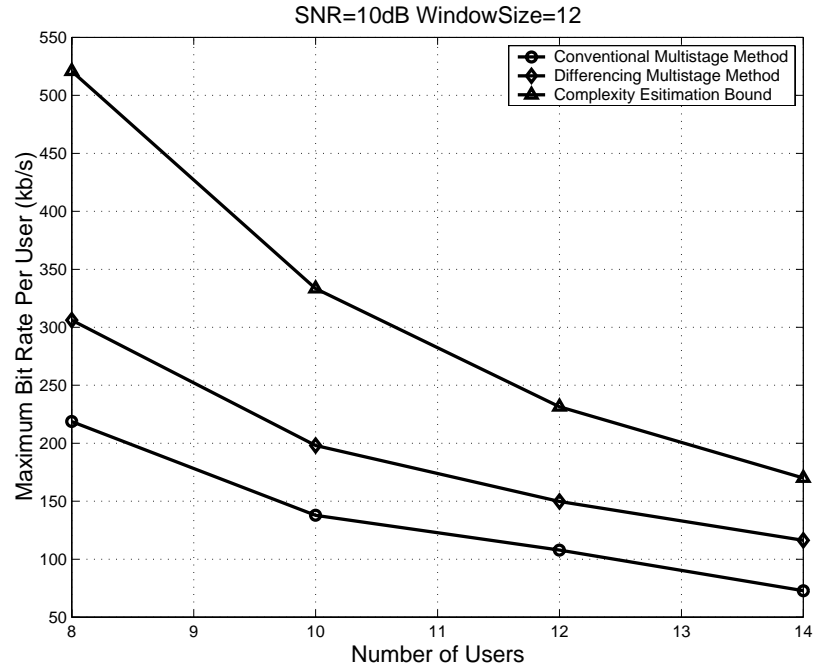


Figure 3.5 : The real-time processing speed of 'C6x DSP

Figure 3.5 shows the complexity estimation bound and the actual profiling result by 'C6201. In 12-user system, the differencing multistage detector can reach up to 150kb/s/user, while the conventional multistage detector can only process at 100kb/s/user. The speed curve does not approach the theoretical bound because besides flops in the algorithm, many other overhead operations such as memory access are involved.

Figure 3.6 and 3.7 are the testing results for this testbed system. They show the probability density functions (PDF) of the transmitted bits, soft decisions before and after the differencing multistage detector. The transmitted bits are binary distributed. where the probabilities of +1 and -1 are equal to each other. The multiuser

environment and AWGN channel expand the distribution of the soft decisions. In Figure 3.6, MAI = 0dB, which means all the users have the same power. There is a huge overlapping area around the threshold 0 due to the multiuser interference, which causes a great deal of errors. However, after using the multistage detector, we get a cleaned-up area around threshold. The effectiveness of the multistage detector is further shown in Figure 3.7, where MAI = 12dB. The weakest user's amplitude is about one fourth of the interfering users. who cover the desired (weakest) user's information completely. On the contrast, the multistage detector removes the interference and distinguishes the weakest user from any other users.

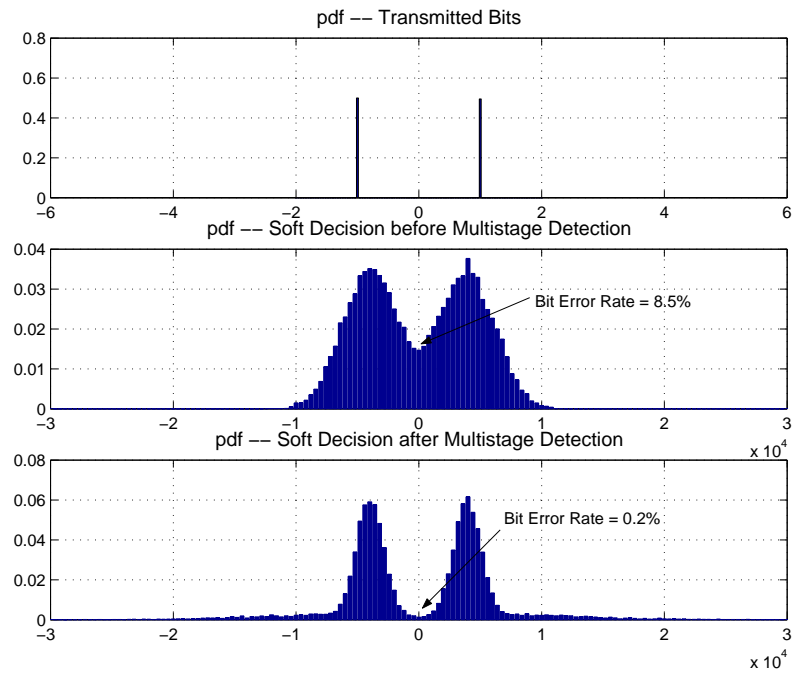


Figure 3.6 : Number of users = 12, Bits = 2000, SNR = 6dB, MAI = 0dB

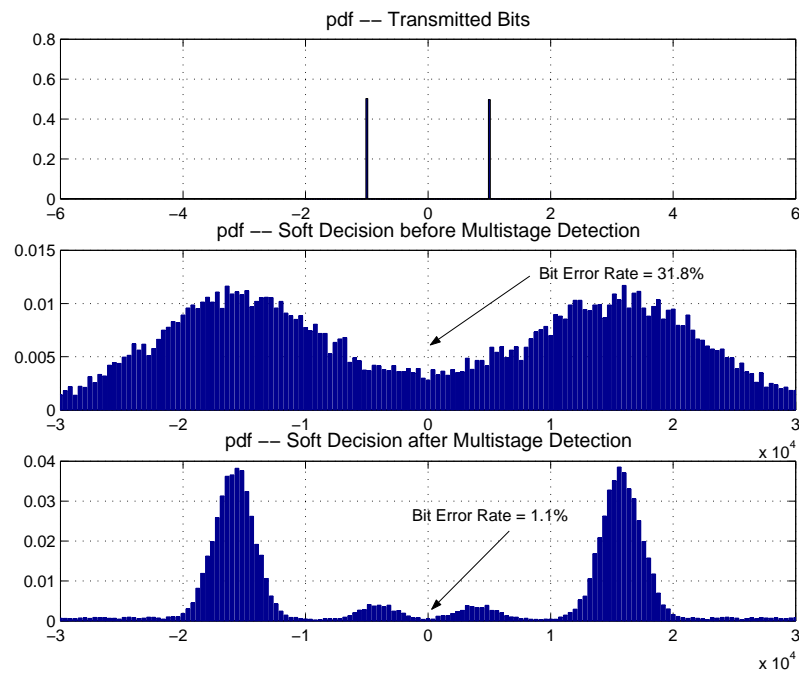


Figure 3.7 : Number of users = 12, Bits = 2000, SNR = 6dB, MAI = 12dB

Chapter 4

ASIC Implementations

In Chapter 3, we have discussed DSP implementations. Although high performance general purpose DSPs meet the real-time requirement, they are not cost-effective. In the real communication systems, sophisticated algorithms are mostly implemented by ASICs (Application Specific Integrate Circuits). They are so-called hardware implementations, which are potentially cheaper and faster. The power consumption is also lower. In this chapter, we present our ASIC implementation for the differencing multistage detector.

4.1 Prototyping the multistage detection algorithm

The first two stages' structure of the differencing multistage detector is shown in Figure 4.1. In this figure, we can get some sense about how interference cancellation works. In the first stage, it uses the previous estimations (from the matched filter output) to generate a new vector of signals. Then sum up all the interfering users and subtract them out from the matched filter output signal \mathbf{y} . In the end, a better estimation $\hat{\mathbf{d}}^{(1)}$ vector is produced. We need a conventional multistage detector as the first stage, so that two initial vectors are obtained for differencing method.

After the first stage, the differencing multistage detector starts to use the differencing vector $\hat{\mathbf{x}}^{(1)}$ as the input. Furthermore, the inputs for the interference cancellation is not the matched filter output, but previous stage's output $\mathbf{z}^{(1)}$.

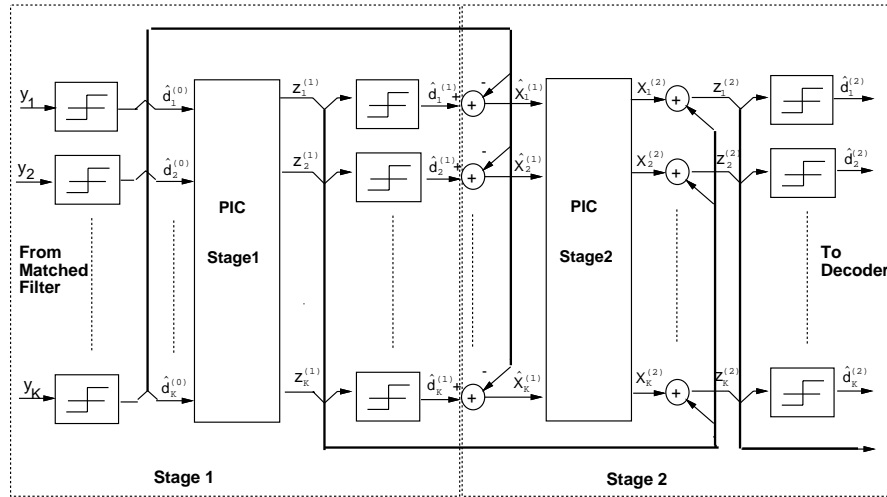


Figure 4.1 : The structure of the differencing multistage detector

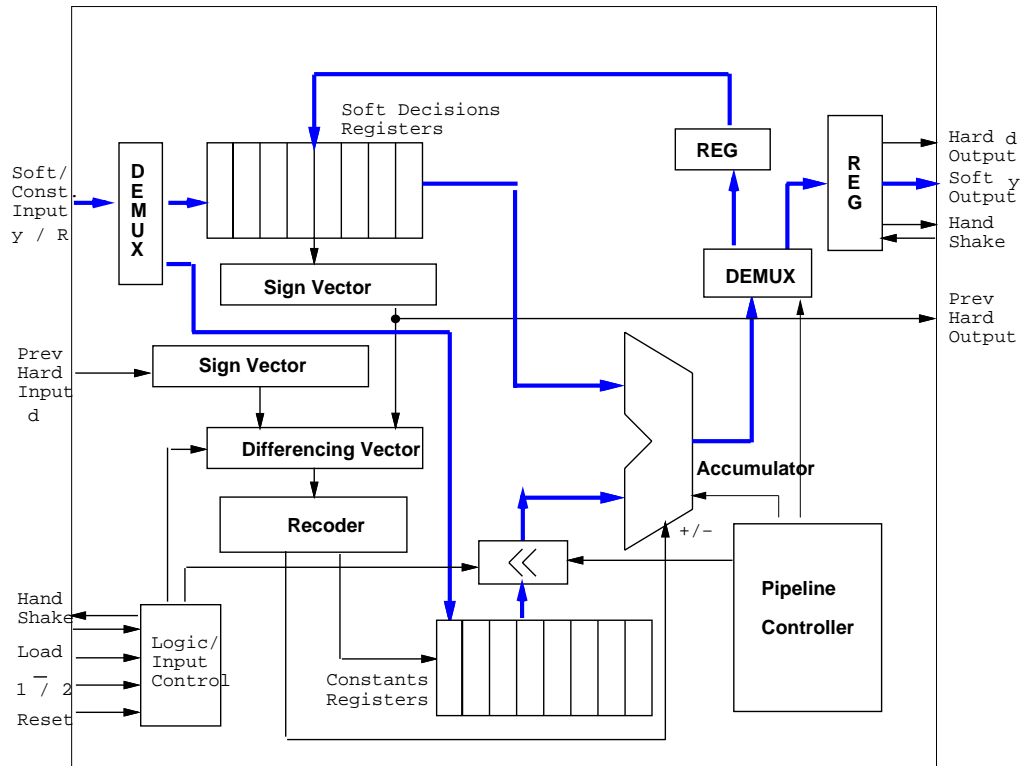


Figure 4.2 : Single ALU implementation of differencing multistage detector

Figure 4.2 is our design to implement the differencing multistage detector for synchronous users in hardware using a single ALU. If we skip the differencing vector and the shift of the cross-correlation constants, it can also be used as the conventional multistage detector. This design is based on 8-user short code spreading system. Soft decision inputs are parallel in bits for each user and time duplexing for all users. The output is also in this time duplexing format. The timing of inputs and outputs is controlled by hand shaking mechanism. We assume later stage is always ready since we are using differencing algorithm and it converges. The number of cycles needed for earlier stage must be equal to or more than 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 differencing vector is generated by combinational logics. The recoder block will record the non-zero elements by their positions in the order of the first come, the highest priority. 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 it. Loading, shifting, accumulating and writing back are organized as a simple pipeline machine, managed by two phase clock. It will not stall because no data and control dependencies exist. Finally the soft and hard decision are generated one by one with certain hand-shaking protocols to the next stage. By passing the differencing vector generation block, this device works as the conventional multistage detector.

4.2 Implementation issues of the detector chip

4.2.1 Chip specifications

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	bi-phase clock at 12.5MHz
Throughput	190kb/s @12.5MHz
Transistor count	6K
Die size	$2.0 \times 2.0mm^2$

Table 4.1 : The chip specification

In table 4.1, we summarize our chip specifications. To simplify the hardware design, we have focused on fixed-point implementation and synchronous system only. We choose eight-user system because all the control logic are basically binary counters. The number of users with a power of 2 would be most efficient. The input data bus is limited by the pin count of our chip. In order to meet the fixed point word length requirement, as the analysis in Section 2.7, we choose 10 bits as the input precision. The detector will allow us to detect eight users in a MAI=15dB and SNR=6dB environment. The internal data bus is wider than input or output bus to ensure no overflow will occur during computations.

In the differencing vectors, which are expressed as the difference of two consecutive hard decisions, there are only +2, -2 or 0s. Bypassing all the zeros, we are going to save the total computations. +2s and -2s can be easily implemented as arithmetic left shift once. So dedicated multipliers are not required.

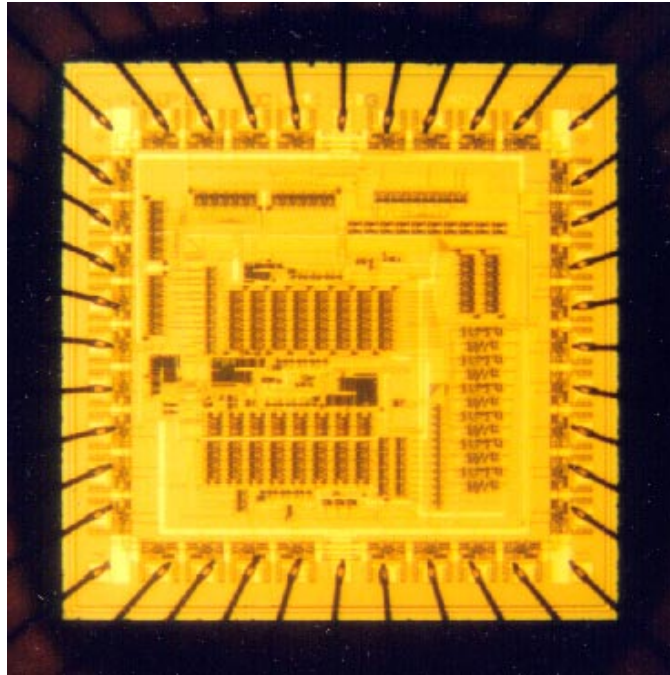


Figure 4.3 : ASIC chip layout: $1.2\mu m$, $2.0 \times 2.0mm^2$ die

4.2.2 Block descriptions

Figure 4.3 shows the actual chip layout by Magic. The chip has five major blocks: recoder, register A, B, C and ALU. Some PLA and temporary registers are auxiliary blocks.

Recoder: The recoder(Figure 4.4) is an innovative design in this chip. While we storing the soft decision values, it detects the position of all the non-zero elements in the differencing vector and stores their addresses and sign bits. The core part in this block is the up-down counter. It counts up when “abs” equals 1, i.e. a non-zero element in the differencing vector. In the interference cancellation stage, the “fetch” signal let the counter count down. When it reaches zero (all the data have been fetched), it sends out “done” signal, indicating the finish state of interference

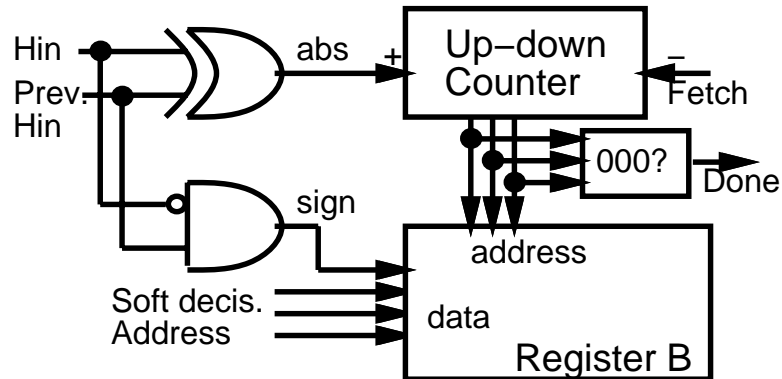


Figure 4.4 : The architecture of the recoder

cancellation.

Register file A has 12x8 register block, "read" decoder and "write" decoder and corresponding PLAs. It has the capability to be read and written at the same time (with different addresses). In the first step, it stores the input soft decision values. During the accumulation stage, it outputs the soft decisions to the ALU and stores the accumulation results.

Register file B has 4x8 register block, decoder and the corresponding PLA. It stores the address of non-zero element addresses in the differencing vector. During the accumulation stage, it outputs these addresses to fetch the proper cross-correlation constants in register C to the ALU.

Register file C has 8x8 register block the decoder and multiplexers. Due to the synchronous environment and the orthogonality of the Gold code 31, the cross-correlation matrix (**RA**) has the same elements in each row, so only eight registers are necessary to store the matrix. Before we load the soft decision values to the register file A, we could load the cross-correlation constants to this register. The write of this register file is controlled by the same PLA in register file A and the read is controlled

by the output of the register file B. We select the proper cross-correlation constants by determining if it is in the first stage and if the address is the same as that of register A (since we do not cancel the desired user's signal).

ALU is a 12-bit adder/subtractor logic block, which uses six 2-bit carry look-ahead addition to reduce the propagation delay time.

4.2.3 Cascade mode

Our chip implements a single stage of the conventional/differencing multistage detection algorithm. We can implement the multistage detection algorithm by cascading chips together. The number of chips that are cascaded depends on the number of the stages in the detector. If the chip is used as the first stage, we directly feed the inputs into the chip, which should have been supplied by the matched filter bank. But if the chip is not used as the first stage, it receives inputs from the chip of the previous stage. The flow of data between the chips is controlled by a hand shaking mechanism.

A three-stage differencing multistage detector is shown in Figure 4.5. 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. Thus, using differencing method will reduce the system delay dramatically.

4.3 Chip performance

The heart of the chip is the ALU, which performs the calculations in the nested loop and stores them back into registers. Hence, the speed of the ALU operating on the input and storing them back in the register file constitutes the critical path in the

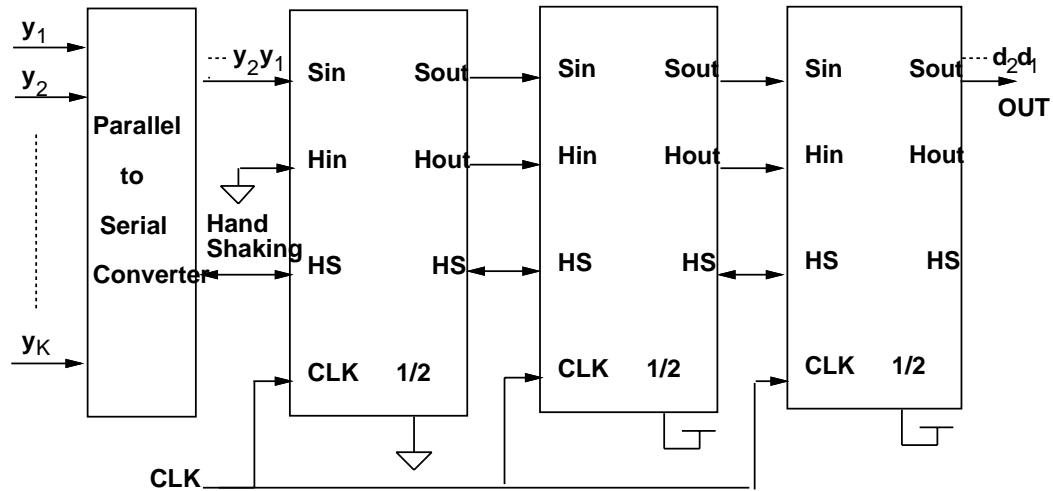


Figure 4.5 : Cascade mode of three-stage differencing multistage detector

circuit. To perform the worst case timing analysis, we should ensure that a change in the carry of one stage causes a change in the carry of the next stage, so that the carry bit ripples all the way to the top and produces a change in the output of the last stage.

Maximum Frequency

The SPICE analysis showed that the worst case could occur when all the inputs are high (carry is high) and all the carries change to low (due to all the inputs changing to low). i.e on a 1-to-0 transition of all the carries. The SPICE simulation for the worst case showed the maximum delay to be around 17ns. The result is shown in Figure 4.6. Hence, the theoretical maximum frequency obtained is 100MHz. But, the speed of operation is limited by the I/O pads to around 25MHz. Hence, maximum operating frequency for the chip is 25MHz.

Cascade mode test

From our hardware simulation, three-stage system delay with the differencing

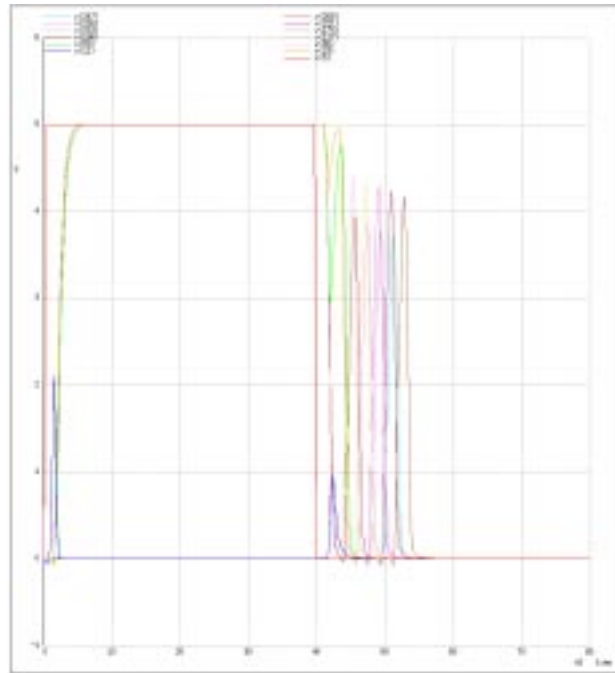


Figure 4.6 : The SPICE analysis of the ALU

algorithm is less than 100 cycles. Working at the 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 $12\mu s$. Further, system throughput is determined by the first stage because it is the slowest stage in the system. By our design, the system can reach a throughput up to 190kb/s with proper buffering. This rate meets 144kb/s requirement of the W-CDMA communication requirements.

4.4 Scalable ASIC design

The tiny chip implementation shows very high performance in the real-time communication. However, due to resource limitations, this chip can handle much less users than a commercial CDMA base station can. If we need to design a more powerful chip, it should follow the specifications in table 4.2.

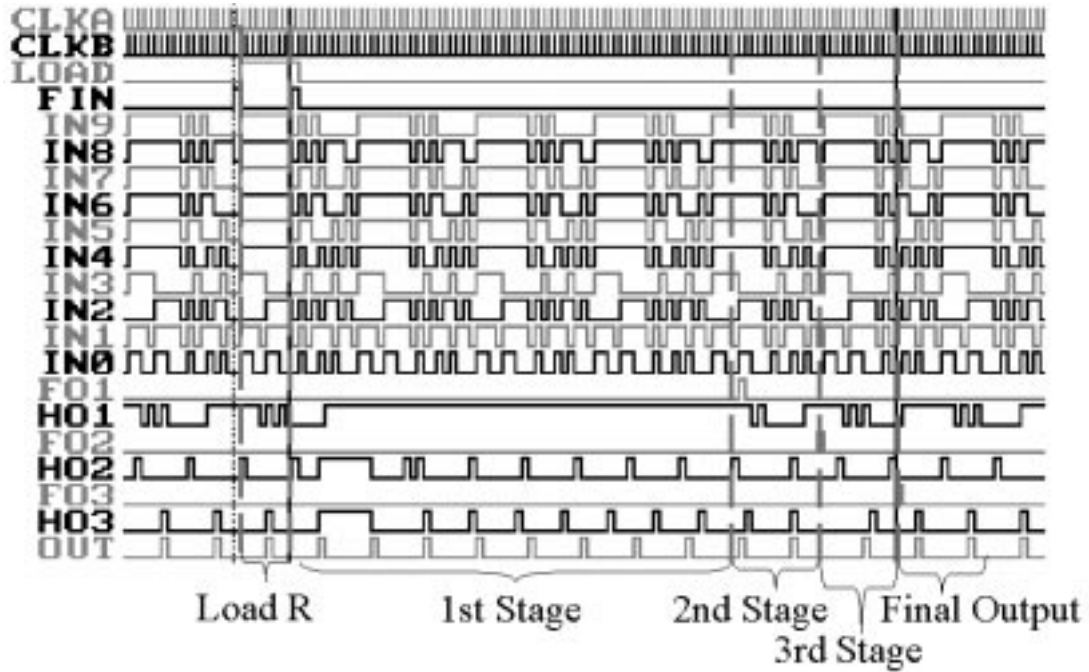


Figure 4.7 : Chip testing result

	Current Tiny-Chip	Chip in the future
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 4.2 : Scalable ASIC design for the multistage detector

For a larger ASIC design, high level behavior description and synthesis language such as VHDL or Verilog should be used. If we design a chip, which can handle 30 asynchronous users (upper limit for Gold code 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 2.7). We expand the data bus width to 16 bits in order to accommodate higher MAI. Total number of register cells are $900 \times 8 + 30 \times 16 \approx 8kbit$. Since a static register cell consists of 10 transistors, we can presume that total number of transistors would be around 100K. Current high density FPGA, such as Xilinx XC4000XL [45], has up to 500k gates, running at 96MHz clock rate. Its capacity and speed are good enough for W-CDMA base station detector design.

4.5 Comparison of DSP and ASIC implementations

Since we have implement the same algorithm – differencing multistage detector by both DSP and ASIC, it would be very interesting to compare these two different implementations in terms of cost, speed and design cycle. Table 4.3 summarizes the major features of these two implementations.

From Figure 3.5, 'C62 DSP can process eight users at 300kb/s/user, while ASIC works at 190kb/s/user. However, DSP requires much higher clock rate and higher complexity. VLIW architecture is a good general purpose DSP core, but not cost effective for one specific application. On the other hand, DSP implementation is more flexible and versatile. It allows developers to fine tune the algorithm easily, consequently the design cycle is short.

The next generation W-CDMA base station would use the combination of the general purpose DSP core and ASIC for the system-on-a-chip design. Current TI's

8 users	DSP (C6201)	ASIC (Tiny Chip)
Clock	200MHz	12.5MHz
Precision	16-bit	12-bit
Speed	300kb/s/user	190kb/s/user
Complexity	10M (0.25 μ m) transistors	6K (1.2 μ m) transistors
Design Cycle	short	long

Table 4.3 : Comparison between DSP implementation and ASIC implementation

'C54 family DSP has DSP core with a Viterbi decoder, dedicated to the communication algorithm. Such kind of software and hardware co-design allows DSP core to do higher level control and signal processing work and ASIC to implement highly developed algorithms, in order to achieve the best cost-effective design.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we have focused on the implementation issues for the multiuser detection algorithm in CDMA wireless communication systems. Particularly, we have implemented the differencing multistage detector by both DSPs and ASICs. The major contributions of this thesis is the following:

First, we invented the differencing multistage detection algorithm. Compared to the conventional single user like detector, the multistage detector shows a great improvement in the detection performance. We exploited 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. This technique shows a great deal of savings in contrast to the conventional multistage detector, shown by a 2X speedup in a three-stage detector. We have also investigated the convergence property and fixed-point prototyping issues to insure its feasibility.

Second, we have developed an optimized DSP implementation of this algorithm. We use multi-step optimization methods to make full usage of all eight functional units in TI's 'C6x DSP and meet the real-time requirement of 144kb/s/user in a 12-user system.

Finally, we designed an ASIC chip to implement the differencing multistage detec-

tor. The chip was fabricated by $1.2\mu m$ CMOS technology with a die size $2.0 \times 2.0 mm^2$. Three cascaded chips perform a three-stage multistage detection with a throughput of 190kb/s/user in an eight-user system. The architecture is scalable for a larger design.

5.2 Future work

In Section 2.2.2, we mentioned joint channel estimation and multiuser detection scheme. It both simplifies the receiver structure and achieves high performance. Therefore, it would be the research direction in the future. The future research will lead to the implementation of joint synchronization and detection receivers in DSPs and ASICs.

For the hardware implementation, the first stage is the bottleneck of the detector, which limits the throughput of the system. How to reduce the complexity of the first stage will be an interesting topic. Some viable schemes could be: lower precision for the first stage or higher speed and complexity for the first stage.

The other limitation is that current differencing multistage detection algorithm can only be adopted in short spreading-code W-CDMA systems. Using some generic interference cancellation methods, we should also be able to extend this algorithm to long spreading code system.

Finally, since we have already had a basic architecture for the differencing multistage detector, we could easily port this design to a larger ASIC. We should be able to use hardware description language VHDL or Verilog to design the chip and implement it on FPGAs.

Appendix A

Appendix

A.1 How to get Figure 1.3

The standard SNR_0 to get a bit error rate P_e is

$$SNR_0 = \frac{E_b}{N_0/2} = 20 \log(Q^{-1}(\frac{P_e}{0.5}))^2 \quad (\text{A.1})$$

The SNR_1 needed to maintain a bit error rate 10^{-3} is:

$$\frac{1}{SNR_1^{-1} + \alpha K} = 20 \log(Q^{-1}(\frac{P_e}{0.5}))^2 \quad (\text{A.2})$$

where α is the cross-correlation coefficient and K is the number of users. Here we assume all the users have the same power and they are normalized to 1.

Bibliography

- [1] J.G. Proakis, *Digital Communications*, McGraw-Hill, Inc., 1989.
- [2] R. Lupus and S. Verdú, "Near-far resistance of multiuser detectors in asynchronous channels," *IEEE Trans. Commun.*, vol. vol. 38, no. no. 4, pp. pp. 496–508, Apr. 1990.
- [3] S. Parkvall, "Long vs short spreading codes in cellular DS-CDMA," in *Proc. of 5th IEEE ISSSTA*, Sep. 1998, vol. 3, pp. pp. 681–685.
- [4] S. Moshavi, "Multi-user detection for DS-CDMA communications," *IEEE Communications Magazine*, pp. 124–136, Oct. 1996.
- [5] A. Duel-Hallen and et al, "Multiuser detection for CDMA systems," *IEEE Personal Commun.*, pp. 46–58, Apr. 1995.
- [6] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. vol. IT-32, no. no. 1, pp. pp. 85–96, 1986.
- [7] R. Lupas and S. Verdú, "Linear multiuser detectors for synchronous code-division multiple-access channels," *IEEE Trans. Inform. Theory*, Jan 1989.
- [8] M. Saquib and et al, "Decorrelating detectors for a dual rate synchronous DS/CDMA system," in *Proceeding of VTC'96*, 1996, pp. 377–81.
- [9] U. Madhow and M. L. Honig, "MMSE interference suppression for direct-sequence spread-spectrum CDMA," *IEEE Trans. Commun.*, vol. vol. 42, no. no. 12, pp. pp. 3178–88, 1994.
- [10] C. Sengupta and S. Das, "Mops count for channel parameter estimation and detection algorithms," Tech. Rep., ECE Dept. Rice Univ., 1998.
- [11] P. D. Alexander and L. K. Rasmussen, "On the windowed Cholesky factorization of the time-varying asynchronous CDMA channel," *IEEE Trans. Commun.*, vol. vol. 46, no. no. 6, pp. pp. 735–737, June 1998.
- [12] S.V. Uppla and J.D. Sahr, "Recursive structures and finite impulse response implementations linear multiuser detectors for an asynchronous CDMA system," *IEEE J. of Sel. Commun.*, vol. vol. 16, no. no. 9, pp. pp. 1736–46, 1998.

- [13] M. C. Reed and et al, "Iterative multiuser detection for CDMA with FEC: near-single-user performance," *IEEE Trans. Commun.*, vol. vol. 48, no. no. 12, pp. pp. 1693–99, 1998.
- [14] M.J. Juntti and J.O. Lilleberg, "Linear FIR multiuser detection for multiple data rate CDMA systems," in *Proceeding of VTC'97*, 1997, pp. 455–59.
- [15] H. V. Poor and S. Verdú, "Probability of error in MMSE multiuser detection," *IEEE Trans. Commun.*, vol. vol. 43, no. no. 3, pp. pp. 858–71, May 1997.
- [16] T. B. Giallorenzi and S. G. Wilson, "Suboptimum multiuser receivers for convolutionally coded asynchronous DS-CDMA systems," *IEEE Trans. Commun.*, vol. vol. 44, no. no. 9, pp. pp. 1183–96, Sep. 1996.
- [17] P.B. Rapajic and B.S. Vucetic, "Adaptive receiver structures for asynchronous CDMA systems," *IEEE J. of Sel. Commun.*, vol. vol. 12, no. no. 4, pp. pp. 685–97, May 1994.
- [18] M.L. Honig and W. Veerakachen, "Performance variability of linear multiuser detection for DS-CDMA," in *Proceeding of VTC'96*, Atlanta, GA, 1996, pp. 372–376.
- [19] Andrew L. C. Hui and Khaled Ben Letaief, "Successive interference cancellation for multiuser asynchronous DS/CDMA detectors in multipath fading links," *IEEE Trans. Commun.*, vol. vol. 46, no. no. 3, pp. pp. 384–91, Mar. 1998.
- [20] Y. Cho and J. H. Lee, "Analysis of an adaptive SIC for near-far resistant DS-CDMA," *IEEE Trans. Commun.*, vol. vol. 46, no. no. 11, pp. pp. 1429–32, Nov. 1998.
- [21] M.K. Varanasi and B. Aazhang, "Multistage detection in asynchronous Code-Division Multiple -Access communications," *IEEE Trans. Commun.*, vol. vol. 38, no. no. 4, pp. pp. 509–519, Apr. 1990.
- [22] M.K. Simon D. Divsalar and D. Raphaeli, "Improved parallel interference cancellation for CDMA," *IEEE Trans. Commun.*, vol. vol 46, no. no. 2, pp. pp. 258–268, Feb. 1998.
- [23] A. Duel-Hallen, "A family of multiuser decision-feedback detectors for asynchronous code-division multiple-access channels," *IEEE Trans. Commun.*, vol. vol. 43, no. no. 2/3/4, pp. pp. 421–34, Feb./Mar./Apr. 1995.
- [24] Michael Moher, "An iterative multiuser decoder for near-capacity communications," *IEEE Trans. Commun.*, vol. vol. 46, no. no. 7, pp. pp. 870–880, July 1998.

- [25] Laurie B. Nelson, "Iterative multiuser receivers for CDMA channels: An EM-based approach," *IEEE Trans. Commun.*, vol. vol. 44, no. no. 12, pp. pp. 1700–10, Dec. 1996.
- [26] A. Hafeez and W.E. Stark, "Combined decision-feedback multiuser detection/soft-decision decoding for CDMA channels," in *Proceeding of VTC'96*, 1996, pp. 382–86.
- [27] H. Elders-Boll and et al, "Implementation of linear multiuser detectors for asynchronous CDMA systems by linear interference cancellation algorithms," in *Proc. of VTC'98*, May 1998, vol. 3, pp. 3225–28.
- [28] A.C.K. Soong and W.A. Krzymien, "Performance of reference symbol assisted multi-stage successive interference canceling receiver with quadriphase spreading," in *Proceeding of VTC'97*, 1997, pp. 460–64.
- [29] K. Jamal and E. Dahlman, "Multi-stage serial interference cancellation for DS-CDMA," in *Proceeding of VTC'96*, 1997, pp. 671–75.
- [30] Wun-Chang Lin, "Differentially matched filter for a spread spectrum system," United States Patent 5,663,983, Sep. 2 1997.
- [31] Wun-Chang Lin Kuang-Chan Liu, "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.
- [32] G. Xu and J. Cavallaro, "Real-time implementation of the multistage algorithm for next-generation wideband CDMA systems," in *Proceeding of SPIE*, July 1999.
- [33] J.K. Hinderling and et al, "CDMA mobile station modem ASIC," *IEEE J. Solid-State Circuits*, vol. vol. 28, pp. pp. 253–260, Mar. 1993.
- [34] Inyup Kang and Alan N. Willson Jr., "Low-power Viterbi decoder for CDMA mobile terminals," *IEEE J. Solid-State Circuits*, vol. vol. 33, no. no. 3, pp. pp. 473–482, Mar. 1998.
- [35] Chaitali Sengupta, *Algorithms and Architectures for Channel Estimation in Wireless CDMA Communication systems*, Ph.D. thesis, Rice University, December 1998.
- [36] Z. Xie and et al, "Joint signal detection and parameter estimation in multiuser communications," *IEEE Trans. Commun.*, vol. vol. 41, no. no. 7, pp. pp. 1208–15, August 1993.
- [37] G.H.Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1989.

- [38] R. K. Morrow, "Accurate CDMA BER calculations with low computational complexity," *IEEE Trans. Commun.*, vol. vol. 46, no. no. 11, pp. pp. 1413–17, Nov. 1998.
- [39] C. Sengupta and et al, "Fixed point error analysis of multiuser detection and synchronization algorithms for CDMA communication systems," in *Proc. of ICASSP'98*, May 1998, vol. 6.
- [40] Suman Das and et al, "Hardware design issues for a mobile unit for next generation CDMA systems," in *Proc. of SPIE'98*, July 1998, vol. VIII, pp. pp. 476–487.
- [41] Seehyun Kim and et al, "Fixed-point optimization utility for C and C++ based digital signal processing programs," *IEEE Trans. Circuits and Systems*, vol. vol. 45, no. no. 11, pp. pp. 1455–64, Nov. 1998.
- [42] Z. Kostic and S. Seetharaman, "Digital signal processors in cellular radio communications," *IEEE Commun. Mag.*, pp. 22–35, Dec. 1997.
- [43] "TMS320C6201, Digital Signal Processor Data Sheet," Texas Instruments, Mar. 1998, <http://www.ti.com/>.
- [44] "TMS320C6201, Assembly Benchmarks," Texas Instruments, Mar. 1998, <http://www.ti.com/>.
- [45] *XC4000XL/XLA Series FPGAs*, <http://www.xilinx.com/products/>.
- [46] S. G. Glisic and P. A. Leppanen, Eds., *Code Division Multiple Access Communications*, Kluwer Academic Publishers, 1995.
- [47] S. Glisic and B. Vucetic, *Spread Spectrum CDMA Systems for Wireless Communications*, Artech House Publishers, 1997.
- [48] F. M. Gardner and J.D. Baker, *Simulation Techniques: Models of Communication Signals and Processes*, John Wiley & Sons, Inc, 1997.
- [49] V. K. Madisetti, *VLSI Digital Signal Processors: An Introduction to Rapid Prototyping and Design Synthesis*, Butterworth-Heinemann, 1995.
- [50] M. A. Bayoumi, Ed., *VLSI Design Methodologies for Digital Signal Processing Architectures*, Kluwer Academic Publishes, 1994.
- [51] E. Spedicato, Ed., *Computer Algorithms for Solving Linear Algebraic Equations*, Springer-Verlag, 1990.