

MP Core: Algorithm and Design Techniques for Efficient Channel Estimation in Wireless Applications

Yan Meng
Timothy Sherwood

Andrew P. Brown
Hua Lee
University of California, Santa Barbara
Santa Barbara, CA 93106-9560

Ronald A. Iltis
Ryan Kastner

yanmeng@engr.ucsb.edu; abrown00@umail.ucsb.edu
{iltis,hualee,kastner}@ece.ucsb.edu; sherwood@cs.ucsb.edu

ABSTRACT

Channel estimation and multiuser detection are enabling technologies for future generations of wireless applications. However, sophisticated algorithms are required for accurate channel estimation and multiuser detection, and real-time implementation of these algorithms is difficult. This paper presents architectural design methods for wireless channel estimation which can be leveraged to enable real-time multiuser detection. We redesign the matching pursuit (MP) channel estimation algorithm to reduce the complexity while maintaining the estimation accuracy. Furthermore, we develop a parameterized intellectual property (IP) core, which provides a hardware implementation of the MP algorithm. Experimental results demonstrate the effectiveness and efficiency of the new algorithm and IP core for channel estimation. The implementation of our MP core on a modern, high performance reconfigurable system is about 216 times faster than running the algorithm on a state of the art microprocessor. The MP core possesses the speed required for performing true multiuser detection, enabling future generations of wireless communication applications.

Categories and Subject Descriptors

B.5.0 [REGISTER-TRANSFER-LEVEL IMPLEMENTATION]: General; J.6 [Computer-Aided Engineering]: Computer-Aided Design

General Terms

Algorithms, Design, Performance, Experimentation

Keywords

Channel estimation, matching pursuit algorithm, design space exploration

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

Copyright 2005 ACM 1-59593-058-2/05/0006 ...\$5.00.

1. INTRODUCTION

While the vision of ubiquitous connectivity is quickly becoming a reality due in large part to the rapid development and deployment of wireless networks, many important challenges remain. State of the art wireless devices are optimized for dealing with a few low bandwidth users in a relatively unobstructed environment. When many users attempt to access the network at once, or when there is significant distortion due to multiple transmission paths, these systems either slow to a crawl or even simply cease to function [17]. Exacerbating these problems is the fact that users demand not only more extensive connectivity, but also increased bandwidth and additional features such as position information. To address these problems, intelligent wireless systems must be developed that can mitigate and even exploit the existence of multiple transmission paths.

If, for example, a user is trying to connect to an access point located in an office down the hall, there will typically be many transmission paths, including a highly attenuated direct path and other multiple-reflection paths. Associated with each path is a delay and attenuation, and thus the received signal is corrupted due to destructive interference. Not only does unmitigated multipath interference result in decreased data rate, but other applications such as accurate radiolocation become almost impossible. Rayleigh fading [13] due to multipath propagation will cause catastrophic failure in received signal strength-based radiolocation. For accurate radiolocation, time-of-arrival (TOA)-based methods are much more promising, but accurate estimation of the direct path arrival time is required [10]. Thus, efficient implementation of an algorithm for accurate estimation of dynamic multipath channels is required. Given estimates of the channel parameters, signal corruption due to multipath propagation can be easily reversed, and the signals due to multiple paths can be combined coherently for increased noise immunity, resistance to signal loss due to shadowing, and overall improvements in data rate and bit/frame error rates.

While there have been many theoretically-sound approaches proposed for multipath channel estimation and multiuser detection [2, 8, 10, 17], these approaches have not yet been adopted by hardware designers because of the complexity of the algorithms involved and the cost associated with realizing them in an actual implementation. In this paper we present a parameterized design for a small, high throughput,

channel estimation engine that can effectively handle a large number of multiple transmission paths. The result of our research is a synthesizable IP-core that can be quickly tuned to the requirements of its application and then instantiated in any number of wireless devices. We begin with the matching pursuit (MP) algorithm [3, 10] for DS-CDMA signals, which is able to achieve accurate channel estimation with reasonable complexity, and which can be combined with the GSIC algorithm for efficient multiuser detection [10]. By redesigning the MP algorithm, we are able to achieve a substantial increase in efficiency with zero decrease in estimation accuracy. In fact, the key to our design is a cross-cutting approach that has resulted in novel optimizations at every level, from the theory and algorithms to the arithmetic and placement.

In this paper we describe our design and quantify the tradeoffs in terms of channel estimation accuracy and the performance and area of the implementation. We explore the possibilities of extracting parallelism from the MP formulation, the effect of fixed and floating point on both area and error, and benefit seen from deep pipelining and specialized multiply units on a reconfigurable device. Based on the design space exploration, our final architecture is mapped onto a Virtex-II XC2V3000 FPGA, resulting in a speedup of over 216 times compared with the execution time on a high performance desktop machine. Our synthesizable MP core meets the needs for high-bandwidth, reliable communication and radiolocation for diverse applications in public safety (e.g. search and rescue), environmental monitoring, ubiquitous computing, and homeland security. Furthermore, it can also be employed to increase the capacity and coverage of wireless networks in environments which are rich with transmission paths such as in large buildings and urban cityscapes.

The rest of the paper is organized as follows. In section 2, we discuss a model for multipath propagation, present a new formulation of the MP algorithm, and compare it with other channel estimation algorithms. In section 3, different design trade-offs are discussed for the design space exploration of the parameterized MP core implementation. We discuss related work in section 4 and draw conclusions in section 5.

2. MATCHING PURSUIT ALGORITHM

2.1 Multipath Channel Propagation

Wireless communication channels typically contain multiple paths due to scattering effects, and thus the received signal is composed of many delayed and attenuated versions of the transmitted signal. For outdoor communications, the scatterers may be buildings, mountains, etc., while for indoor communications, the scatterers may be walls, furniture, etc. Path lengths may vary greatly. In this paper, the multipath propagation delays are assumed to be less than the symbol duration, a reasonable assumption in most cases.

The multipath channel with continuous-valued delays can be represented by a tapped-delay-line (TDL) filter with discrete-valued delays iT_s , for $i = 0, 1, \dots, N_s - 1$, where $1/T_s$ is the Nyquist sampling rate (twice the chip rate), and N_s is the number of samples per symbol duration [17]. Associated with each TDL path i is a complex-valued channel coefficient f_i . The f_i are given by the sum of sampled sinc(.) functions centered at the true delays τ [5, 10]. A *sparse* channel is one in which $N_f \ll N_s$ channel coefficients have

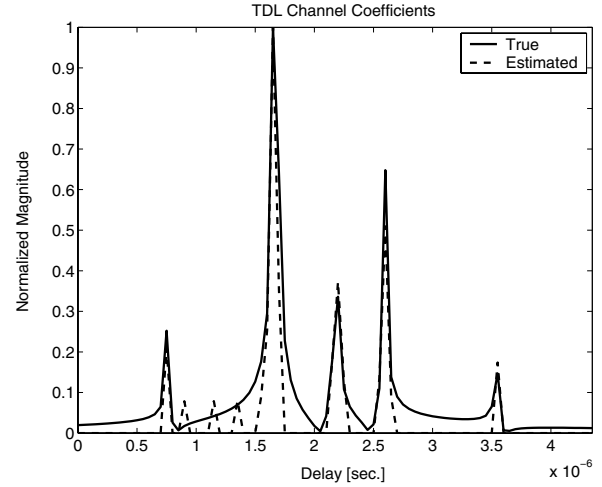


Figure 1: Multipath channel estimation with MP.

non-negligible magnitude. The TDL representation of an example 5-path channel is shown in Figure 1 (solid line).

The received signal after RF-to-baseband down-conversion and A/D sampling is denoted by

$$r = Sf + n \in \mathcal{C}^{MN_s \times 1}, \quad (1)$$

where M is the number of training symbols, n is the sampled additive white Gaussian noise vector, $f = [f_0, f_1, \dots, f_{N_s-1}]^T \in \mathcal{C}^{N_s \times 1}$ is the channel coefficient vector, and $S \in \mathcal{R}^{MN_s \times N_s}$ is the characteristic signal matrix. \mathcal{R} and \mathcal{C} represent the real and complex numbers, respectively, and $(\cdot)^T$ denotes the transpose operation. The i th column S_i of S is the received signal due to path i if $f_i = 1$, and in general $f_i S_i$ is the received signal due to path i . S is given in [10] and is known *a priori*, since it depends only on the DS-CDMA spreading sequence and the transmit and receive filters. Referring to the received signal model (1), the multipath channel estimation problem is that of computing an estimate \hat{f} of f , given S and the received signal vector r containing noise n .

Sparse ML	MP	Fast MP
$O(MN_s C_{N_f}^{N_s} Q^{2(N_s - N_f)})$	$O(2N_f M N_s^2)$	$O(2M N_s^2)$

Table 1: Complexity comparison for sparse multipath channel estimation algorithms. M = number of training symbols, N_s = number of samples per symbol, N_f = number of non-zero channel coefficients, $2/Q$ = channel coefficient precision.

2.2 Redesigning the MP Algorithm

The Matching Pursuit (MP) channel estimation algorithm [3, 10] algorithm provides a low complexity approximation to the Maximum Likelihood (ML) [17, 8] solution for sparse channels, i.e., under the constraint that only N_f elements in \hat{f} are non-zero: $|\hat{f}| = N_f$. The exact ML solution under the sparse channel constraint is given by

$$\hat{f} = \arg \min_{f \in A_{N_f}} \{ \|r - Sf\|^2 \}, \quad (2)$$

where $A_{N_f} = \{f : |f| = N_f\}$. Note that if the channel is not sparse, $N_f = N_s$ and estimates are computed for all elements in f . Since the channel estimation cost function minimized in (2) is non-convex, an exhaustive search is required. The complexity (in terms of the number of scalar multiplications) of the optimal ML algorithm is shown in Table 1, where the binomial coefficient $C_{N_f}^{N_s} = (N_s!)/(N_f!(N_s - N_f)!)$ and $2/Q$ is the precision of the channel coefficient estimates. Clearly, real-time implementation of ML channel estimation is infeasible. By contrast, the MP algorithm [10] is highly efficient.

The algorithm implemented in the Matching Pursuit IP core has been redesigned for a speed improvement of N_f times, with zero reduction in channel estimation accuracy. The new fast MP algorithm is obtained by posing the ML estimation problem in terms of sufficient statistics, as follows.

$$-\|r - Sf\|^2 \propto 2\text{Re}\left\{(V^0)^H f\right\} - f^H A f \quad (3)$$

is a sufficient statistic for signal parameter estimation and data symbol detection [17], where $V^0 = S^T r \in \mathcal{C}^{N_s \times 1}$ and $A = S^T S \in \mathbb{R}^{N_s \times N_s}$. S is known a priori, as mentioned in Section 2.1, and therefore S and A are pre-computed once for all time and stored in memory. The computation of V^0 can be parallelized as N_s vector inner products (correlations) $V_i^0 = S_i^T r$. Since the columns S_i of S are generated as filtered circular-shifted versions of the transmitted DS-CDMA spreading sequence, the computation of V^0 is equivalent to matched filtering the received signal r .

MP maximizes (3) iteratively, one channel coefficient \hat{f}_{q_j} at a time, using a greedy approach in which q_j and \hat{f}_{q_j} are selected such that the increase in (3) at each stage j is the largest possible. That is, the multipath signal components are estimated via successive interference cancellation. The algorithm is summarized in Figure 2(c). Note that the “hat” symbol on f is omitted for convenience. To eliminate the need for division operations, the vector a , with $a_k = 1/A(k, k)$ and $A(k, k)$ denoting the k th diagonal element in A , is pre-computed once for all time and stored in memory. The storage of A and a in memory corresponds to the only increase in memory requirements for the reformulation of the MP algorithm. The increase is insignificant for large M (training sequence length).

After each multipath successive interference cancellation stage, V^j is updated at the start of the next stage j as

$$V^j \leftarrow V^{j-1} - \hat{f}_{q_{j-1}} A_{q_{j-1}}. \quad (4)$$

Since the estimation of \hat{f} via (3) depends only on V^0 and A , with A fixed, effectively the sufficient statistic is updated to reflect cancellation of the signal due to path q_{j-1} . Thus the matched filter output vector V^0 is updated to V^j , compared with the procedure in [10] where an intermediate canceled received signal r^j is formed and matched filtering is repeated. The result is a speedup on the order of N_f times, as shown in Table 1.

The algorithm terminates after stage $j = N_f$. In practice, N_f can be determined on the fly based on $|\hat{f}_{q_j}|$ and/or the SNR (ratio of energy per symbol to noise energy per symbol duration). For the example in Figure 1, $N_f = 15$, $M = 1$, $N_c = 44$ binary DS-CDMA chips, and the MP channel estimate is shown (dotted line) for an SNR of 20 dB.

3. DESIGN SPACE EXPLORATION FOR THE PARAMETERIZED MP CORE

MP, due to its inherent parallelism, is an ideal candidate for efficient implementation on modern reconfigurable platforms. Since on-chip resources are limited, in this section we will study trade-offs of performance and area in implementing the parameterized MP core. To this end, we will explore the design space from the following three perspectives.

1. Parameterization of the MP core.
2. Data representation.
3. Data distribution.

3.1 Parameterization of the MP Core

A good parameterization of MP provides trade-offs between important system metrics, such as estimation accuracy, latency, area, and power/energy consumption. The MP algorithm is inherently parameterizable in terms of the number of training symbols M , the number N_f of nonzero estimated channel coefficients, the number of samples N_s per symbol, and the length N_c and type of the spreading sequence.

The utilization of hardware resources on the target platform provides additional dimensions for parameterization of the MP core. MP involves matrix-vector multiplication, which can be decomposed into parallel vector-vector multiplications. The multiplication/accumulation in vector-vector operations can be further parallelized. Moreover, the channel coefficients of the uncanceled multipath signals can be estimated in parallel. Parallel computing can improve the latency of the platform, but this does come at a cost: more computation resources are required, which increases the cost of the target device. By varying the number of signal estimation resources used in the architecture, we can trade off latency for number of resources.

3.2 Data Representation

The second design trade-off can be performed by optimizing the binary representation of the data in the MP algorithm. Specifically, the number of bits used to represent each data value can be varied, and there is a choice between fixed point and floating point representations. Floating point provides large dynamic range and very high precision, but it can be costly. Fixed-point, on the contrary, represents the data less precisely, but it can save resources and have much better performance.

Bits	(S,E,F)	Adder		Multiplier	
		Performance (ns)	Area (slices)	Performance (ns)	Area (slices)
16	(1,6,9)	59.99	224	39.84	284
32	(1,8,23)	70.45	475	57.59	565
64	(1,11,52)	89.67	1054	69.84	2021
128	(1,32,95)	117.13	2200	100.91	5868

Table 2: Performance and area for floating-point functional units. (S,E,F) indicates the number of sign, exponent, and fractional bits, respectively.

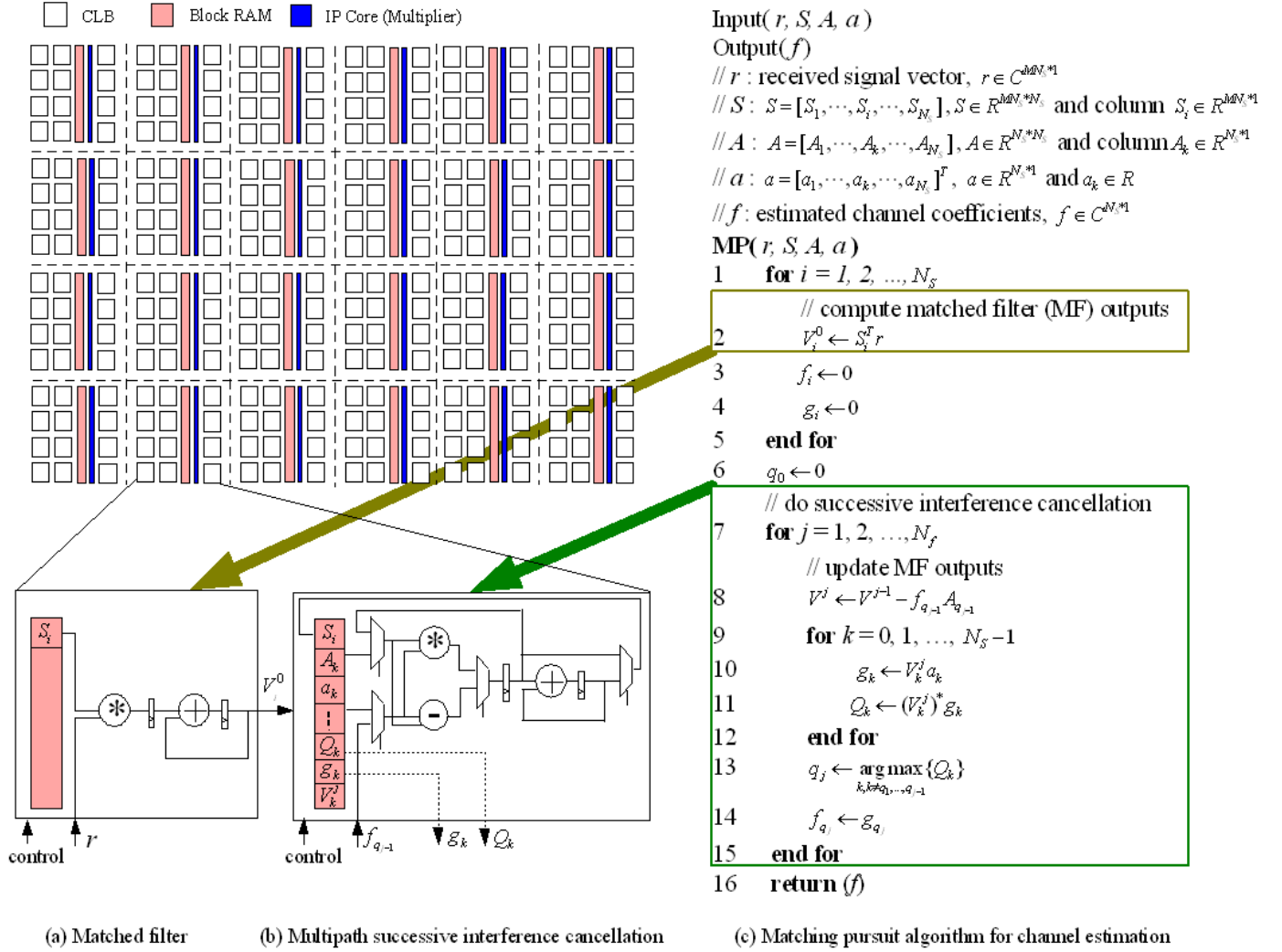


Figure 2: Mapping of the MP algorithm on a modern FPGA. The matched filter (a) and multipath successive interference cancellation (b) are distributed across the FPGA to parallelize the MP algorithm (c).

In MP, adders and multipliers are the basic functional units. Their estimated performance and area shown in Table 2 and Figure 3 can be used to provide rules of thumb for choosing the right data representation scheme in the parameterized MP core. From Table 2 and Figure 3, we can see that compared with the fixed point representation (Figure 3), floating point functional units (Table 2) consume much more hardware resources and have much longer execution time¹. Since the amount of hardware resources MP needs for multiplication and addition operations is very large, employing floating point data representation will require tight resource sharing, which correspondingly degrades the system performance. Fixed point representation restricts the accuracy of the signals in the digital domain compared with its floating point counterpart. However, it requires integer functional units, which are more efficient in both area and performance (Figure 3). Thus, fixed-point

¹Addition is one of the most computationally expensive arithmetic operations in floating point operations [1], as the results in Table 2 demonstrate.

representation is employed in designing the parameterized MP core.

The other data representation consideration involves the trade-off between the number of fixed-point bits and the channel estimation accuracy. To explore the design space in this dimension, we conducted bit width analysis [6, 15]. Figure 4 shows the results for channel estimation average squared error (ASE) vs. number of fixed-point bits for SNRs of -10, 0, 10, 20, 30 dB, where SNR is defined as the ratio of the desired signal energy to the noise signal energy, both measured over one symbol duration. The results are averaged over three different multipath channels, with 30 ensemble runs (different noise realizations) per channel. From Figure 4, it is clear that eight bits is sufficient for accurate multipath channel estimation with optimal dynamic range scaling [6] throughout the implementation.

3.3 Data Distribution Schemes

Another important trade-off to consider for speeding up the MP core is the data distribution schemes. MP processes

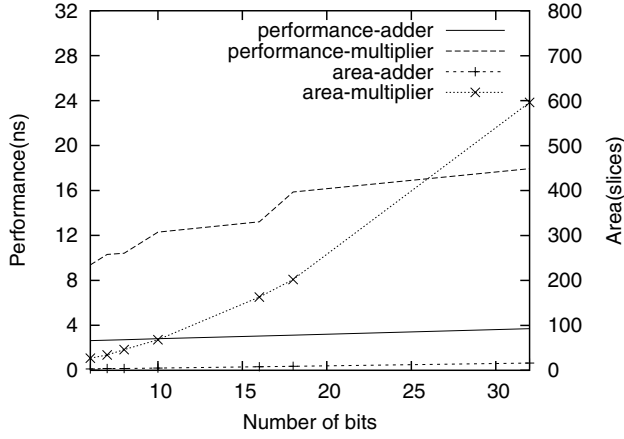


Figure 3: Performance and area for fixed-point functional units.

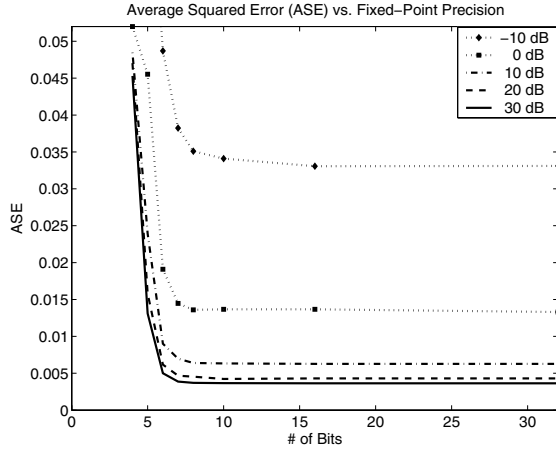


Figure 4: Channel estimation accuracy vs. number of bits for fixed point representation.

a quite large amount of data, depending on the number of symbols, the number of chips per symbol, and the number of samples per chip. In such data dominated designs, poor data distribution can eliminate all benefits gained through parallelization, due to large data transfer times. Thus, it is necessary to perform a careful distribution of the data onto the target platform to achieve good latency.

Figure 2 (a) and (b) depict how the MP data S, A, a , are distributed in one section of a modern reconfigurable device, which has distributed memory, and dedicated IP cores (e.g. multipliers). The top-left figure shows how the RAMs and multipliers are equally distributed across the columns of the chip. All the blobs are identical. The bottom figure shows the configuration of one of the blobs, which includes a block RAM, a multiplier, and surrounding CLBs and programmable interconnections. From the MP algorithm (Figure 2 (c)), we can see that all the correlations (Line 2) in the matched filters can run independently. So the matrix S can be evenly distributed on the block RAMs, with each column S_i stored in one of the block RAMs for the discrete

Bits	Multiplier	BRAM	Slices	Performance (ns)
6	88	89	7806	4896.4
7	88	89	8342	4896.4
8	88	89	8969	4896.4
10	88	89	10134	4896.4
16	88	89	13630	8460.24

Table 3: Experimental results for the MP core.

path delay i . The stored S_i are provided as the operands for multiplications. This parallelization of the matched filters results in a speedup factor of $O(N_s)$. Similarly, in the multipath successive interference cancellation steps (Line 7 through Line 15), a and A can be disseminated equally in the block RAMs, where a_k and A_k are saved in the same block RAM i with S_i . The block RAM i is also used as registers to save intermediate values of Q_k , g_k , and V_k^j , which are also independent between different block RAMs. Due to resource limitations, the multiplications within the same blob share a multiplier in both steps of matched filtering and interference cancellation. By distributing those MP data that are independent onto block RAMs, the memory operations of multiple ports and operations on different block RAMs can be performed simultaneously. Thus, the system can gain more performance improvement.

In general, a major goal in the implementation of the MP core is the use of different techniques to optimize the core toward fast channel estimation. Based on inherent properties of the MP algorithm, the design space parameters are summarized as: (1) hardware resources, (2) parameterized parallelism of the MP algorithm, (3) fixed point architecture, and (4) distributed data storage schemes. Furthermore, the dedicated resources on the reconfigurable device are used to maximize the efficiency of the device.

3.4 Putting It All Together—Experimental results for the parameterized MP core

We present several experimental results for implementing the MP core on a Xilinx Virtex-II XC2V3000 FPGA [18] to better illustrate the general trade-offs involved in our study, specifically in terms of area/performance. Table 3 shows the results for different numbers of bits per data value. From the table, we can see that the data are distributed across 89 block RAMs and 88 multipliers to support parallel execution, and when increasing the number of bits, the area (number of slices) increases accordingly, while the performance decreases. We also compared the execution time of the MP core with a high performance desktop computer, which has a 2.17GHz 3000+ AMD Athlon XP processor and about 1GB DDR PC3200 RAMs. The 8-bit MP core runs about 216 times faster, and the 16-bit MP core runs about 125 times faster. Here it is important to note that even though MP runs quickly on a high performance microprocessor, the achieved speed falls short for the high data rate required by 3G/4G wireless systems. In contrast, the proposed MP core meets the 3G/4G speed requirements, justifying its applicability for implementation in future generations of wireless communications systems.

We would like to mention that another MP core was also

designed, which fully exploits the parallelism of the MP algorithm and is deeply pipelined. Yet, due to the large resource requirement, it does not entirely fit into the largest currently available FPGA. As fabrication technology improves and more transistors are integrated into a silicon chip, it will be possible to map the fully-parallelized MP core onto a single FPGA chip, enabling very high data rate processing.

4. RELATED WORK

Much research has been conducted on the topics of wireless channel estimation, design space exploration, and IP reuse. Thus, space constraints will limit our discussion to a small set of representative contributions which are closely related to our work.

Channel estimation for wireless applications has attracted considerable attention recently, and a variety of algorithms have been developed [2, 3, 8, 10, 14, 17]. In [14], an approximate ML-based algorithm was targeted for implementation in DSP hardware. In the ML algorithm redesign, a matrix inversion step was approximated via gradient descent for improved efficiency. We choose the MP-based algorithm [10] because of its efficiency and accuracy, and we redesign the algorithm for improved efficiency based on a sufficient statistics interpretation.

Design space has been explored from different perspectives through various techniques to meet the design objectives [12, 16]. In this paper, we focus on the bit-width analysis [4, 6, 15] and data distribution [7]. BitWise [15] determines the minimum number of bits by propagating static information in the program data-flow graph, and Bitsize [6] decides the number of bits through sensitivity analysis of outputs. Through careful bit width analysis, hardware costs and energy consumption can be substantially reduced. Huang et al. [7] proposed a methodology for HLS to distribute data across memory logic blocks for reducing data communications.

IP reuse is a design methodology for bridging the gap between available chip complexity and design productivity. Quite a few methods and techniques [9, 11] have been proposed for IP creation, assembly and testing. Metacores [11] is similar to our work; it creates parameterized cores for Viterbi decoding and IIR filters.

5. CONCLUSIONS

In this paper, an MP algorithm has been redesigned for efficient channel estimation. Through exploring the design space of the MP parameters, data representation, and data distribution, a highly efficient parameterized MP core was developed and mapped on a Xilinx Virtex-II FPGA. The resulting core runs over 216 times faster than a high performance desktop machine. With this level of efficiency, in the near future the parameterized MP core could enable high data rate multiuser detection, delivering on decades of promises from communications theory and revolutionizing the state of the art in applied wireless technologies.

6. ACKNOWLEDGMENT

This work was supported by National Science Foundation Grant #0411321.

7. REFERENCES

- [1] P. Belanovic and M. Leeser. A library of parameterized floating-point modules and their use. In *FPL 2002*, 2002.
- [2] A. D. Blackowiak and S. D. Rajan. Multi-path estimates via optimizing highly oscillatory cost functions. In *IEEE J. Oceanic Eng.*, volume 30, July 1998.
- [3] S. F. Cotter and B. D. Rao. Sparse channel estimation via matching pursuit with application to equalization. *IEEE Trans. Communications*, 50:374–377, Mar. 2002.
- [4] S. Mahlke et al. Bit-width cognizant architecture synthesis of custom hardware accelerators. In *IEEE Trans. on Computer-Aided Design*, Nov. 2001.
- [5] J. J. Fuchs. Multipath time-delay detection and estimation. *IEEE Trans. Signal Processing*, 47:237–243, 1999.
- [6] A. A. Gaffar, O. Mencer, W. Luk, and P. Y. K. Cheung. Unifying bit-width optimisation for fixed-point and floating-point designs. In *IEEE Trans. on Computer-Aided Design*, Nov. 2001.
- [7] C. Huang, S. Ravi, A. Raghunathan, and N. K. Jha. High-level synthesis of distributed logic-memory architectures. In *Proc. Int. Conf. Computer-Aided Design*, Nov. 2002.
- [8] R. A. Iltis and S. Kim. Geometric derivation of expectation-maximization and generalized successive interference cancellation algorithms with CDMA channel estimation. *IEEE Trans. Signal Processing*, 51(5):1367–1377, May 2003.
- [9] M. F. Jacome and H. P. Peixoto. A survey of digital design reuse. In *IEEE Design and Test of Computers*, May 2001.
- [10] S. Kim and R. A. Iltis. A matching pursuit/GSIC-based algorithm for DS-CDMA sparse channel estimation. *IEEE Signal Processing Letters*, 11:12–15, Jan. 2004.
- [11] S. Meguerdichian, F. Koushanfer, A. Mogre, D. Petranovic, and M. Potkonjak. Metacores: design and optimization techniques. In *DAC.*, Las Vegas, Nevada, June 2001.
- [12] G. D. Micheli. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill, 1994.
- [13] J. Proakis. *Digital Communications*. McGraw-Hill, New York, NY, 1995.
- [14] S. Rajagopal, S. Bhashyam, and J. R. Cavallaro. Real-time algorithms and architectures for multiuser channel estimation and detection in wireless base-station receivers. In *IEEE Tran. on Wireless Comm.*, number 3, pages 374–377, July 2002.
- [15] M. Stephenson, J. Babb, and S. Amarasinghe. Bitwidth analysis with application to silicon compilation. In *IEEE Trans. on Computer-Aided Design*, Nov. 2001.
- [16] L. Vanzago, B. Bhattacharya, J. Cambonie, and L. Lavagno. Design space exploration for a wireless protocol on a reconfigurable platform. In *DATE'03*, Munich, Germany, Mar. 2003.
- [17] S. Verdú. *Multiuser detection*. Cambridge University Press, New York, Oct. 1998.
- [18] Xilinx, INC. *Virtex-II 1.5V Field Programmable Gated Arrays Datasheet*, Oct. 2001.