Hermitian Optimization and Scalable VLSI Architecture for Circulant Approximated MIMO Equalizer in CDMA Downlink

Yuanbin Guo, Dennis McCain
Nokia Research Center
Irving, Tx 75039
Email: Yuanbin.Guo, Dennis.McCain@nokia.com

Joseph R. Cavallaro
Dept. of Electrical and Computer Engineering
Rice University
Houston, Tx, 77005
Email: cavallar@rice.edu

***Abstract*— In this paper, we propose a parallel and pipelined VLSI architecture for a circulant approximated equalizer for the MIMO-CDMA systems. The FFT-based tap solver reduces the Direct-Matrix-Inverse of the size $(NF \times NF)$ to the inverse of $\mathcal{O}(N)$ sub-matrices of the size $(N \times N)$. Hermitian optimization and tree pruning is proposed to reduce the number and complexity of the FFTs. A divide-and-conquer method partitions the 4×4 sub-matrices into 2×2 sub-matrices and simplifies the inverse of sub-matrices. Generic VLSI architecture is derived to eliminate the redundancies in the complex operations. Multiple level parallelism and pipelining is investigated with a Catapult C High-Level-Synthesis (HLS) methodology. This leads to efficient VLSI architectures with $3\times$ further complexity reduction. The scalable VLSI architectures are prototyped with the Xilinx FPGAs and achieve area/time efficiency.**

I. Introduction

The growing demands for broadband multimedia services, ubiquitous networking via mobile devices push the development of advanced modem technology. Multiple Input Multiple Output (MIMO) technology [1] [2] using multiple antennas at both the transmitter and receiver has recently emerged as one of the most significant technical breakthroughs in modern communications. On the other hand, UMTS and CDMA2000 extensions optimized for data services lead to the standardization of multi-code CDMA systems such as the High-Speed-Downlink-Packet-Access (HSDPA) and its equivalent 1X EV-DV (Evolution Data and Voice) [3]. Recently, the MIMO technology has been proposed in the CDMA downlink systems to achieve much higher data rate than the current 3^{rd} generation cellular systems.

The original MIMO technology is known as D-BLAST [1] and a more realistic strategy as V-BLAST [2] by nulling and cancelling with reasonable tradeoff between complexity and performance. However, the original MIMO spatial multiplexing was proposed for narrow band and flat-fading channels. In a multipath fading channel, the orthogonality of the spreading codes is destroyed and Multiple-Access-Interference (MAI) along with Inter-Symbol-Interference (ISI) are introduced. With a very short spreading gain, the conventional Rake receiver could not provide acceptable performance. Linear-Minimum-Mean-Square-Error (LMMSE) chip equalizer is promising to restore the orthogonality of the spreading code, so as to suppress both the ISI and MAI [4]. However, it requires to inverse a large correlation matrix with $O((NF)^3)$ complexity, where N is the number of Rx antenna and F is the channel length. This is very expensive for hardware implementation. To avoid the Direct-Matrix-Inverse (DMI), adaptive stochastic gradient algorithms such as LMS have been proposed. However, they suffer from stability problems because the convergence depends on the choice of a good step size. On the other hand, non-adaptive block-based algorithms are proposed, e.g., a Conjugate Gradient-based tap solver in [5] with a complexity at the order of $O((NF)^2)$. However, it is shown that this algorithm still has a high complexity for real-time implementation.

To further reduce the complexity, an FFT-based fast algorithm is proposed in [7] by approximating the block Toeplitz structure of the correlation matrix using a block circulant matrix to avoid the DMI. Although this FFT-based algorithm avoids the DMI of the original correlation matrix with the dimension of $(NF \times NF)$, the matrix inverse of some smaller sub-matrices with size of $(N \times N)$ is inevitable for MIMO receiver. For a MIMO receiver with high antenna configuration, the complexity increases dramatically with the number of antennas. The fact that the receiver must be embedded into a portable device makes the design of low complexity and low cost products critical for widespread commercial deployment. It is necessary to determine which range of possible architectures is most suitable for VLSI implementation [6].

In this paper, a reduced complexity MIMO receiver architecture for the FFT-based chip level equalizer is proposed. Hermitian optimization is proposed by utilizing the structures of the correlation coefficients and the FFT algorithm to further reduce the complexity. A reduced-state FFT module is proposed to avoid duplicate computation of the symmetric coefficients and the zero coefficients. The number and complexity of conventional FFT design modules is reduced. The Hermitian feature is then applied to reduce the complexity of the inverse of the sub-matrices. Of particular interest is the non-trivial inverse of many 4×4 sub-matrices. We apply a divide-and-conquer method to partition the 4×4 sub-matrices into four 2×2 sub-matrices. The 4×4 matrix inverse is then dramatically simplified by exploring the commonality. Generic VLSI design architecture is derived from the special design blocks to eliminate the redundancies in the complex operations. The simplified model facilitates the design of efficient parallel VLSI modules such as "*Complex-Hermitian-Multiplication*", "*Hermitian Inverse*" and "*Diagonal Transform*". This leads to efficient architectures with $3\times$ further complexity reduction and more parallel and pipelined VLSI schematic, which is verified in a Xilinx FPGA prototyping platform using a Catapult C based HLS design methodology [8].

II. System Model and Chip Equalizer

A. System Model

The system model of the MIMO multi-code CDMA downlink using M Tx antennas and N Rx antennas is described as follows. Multiple spreading codes are assigned to a single user to achieve high data rate in the multi-code CDMA downlink. First, the high data rate symbols are demultiplexed into KM lower rate substreams, where K is the number of spreading codes used in the system for data transmission. The substreams are divided into M groups, where each substream in the group is spread with a spreading code of spreading gain G. Each group is then combined and scrambled with a long

scrambling code and transmitted through the m^{th} Tx antenna. The chip level signal at the m^{th} transmit antenna is given by

$$d_m(i) = \sum_{k=1}^{K} s_m^k(j) c_m^k(i) + s_m^P(j) c_m^P(i) \quad (1)$$

where j is the symbol index, i is chip index and k is the index of the composite spreading code. $s_m^k(j)$ is the j^{th} symbol of the k^{th} code at the m^{th} substream. In the following, we focus on the j^{th} symbol index and omit the index for simplicity. $c_m^k(i) = c_k(i)c_m^{(s)}(i)$ is the composite spreading code sequence for the k^{th} code at the m^{th} substream where $c_k(i)$ is the user specific Hadamard spreading code and $c_m^{(s)}(i)$ is the antenna specific scrambling long code. $s_m^P(j)$ denotes the pilot symbols at the m^{th} antenna. $c_m^P(i) = c^P(i)c_m^{(s)}(i)$ is the composite spreading code for pilot symbols at the m^{th} antenna. The received chip level signal at the n^{th} Rx antenna is given by

$$r_n(i) = \sum_{m=1}^{M} \sum_{l=0}^{L_{m,n}} h_{m,n}(l) d_m(i - \tau_l) + z(i) \quad (2)$$

where $h_{m,n}(l)$ and $L_{m,n}$ are the l^{th} path channel coefficient and the delay spread between the m^{th} Tx antenna and the n^{th} Rx antenna, respectively. $z_n(i)$ is the additive Gaussian noise at the n^{th} receive antenna.

By packing the received chips from all the receive antennas in a vector $\mathbf{r}(i) = [r_1(i), \cdots, r_n(i), \cdots, r_N(i)]^T$ and collecting the $L_F = 2F + 1$ consecutive chips with center at the i^{th} chip from all the N Rx antennas, we form a signal vector as $\mathbf{r}_A = [\mathbf{r}(i + F]^T, \cdots, \mathbf{r}(i)^T, \cdots, \mathbf{r}(i - F)^T]^T$. In the vector form, the received signal is given by

$$\mathbf{r}_A(i) = \sum_{m=1}^{M} \mathbf{H}_m(i) \mathbf{d}_m(i) \quad (3)$$

where $\mathbf{H}_m(i)$ is the channel matrix with a block Toeplitz structure. The transmitted chip vector for the m^{th} transmit antenna is given by $\mathbf{d}_m(i) = [d_m(i + F), \cdots, d_m(i), \cdots, d_m(i - F - L)]^T$.

B. LMMSE Tap Solver with Circulant Approximation

Linear MMSE based chip equalizer estimates the transmitted chip samples by a set of linear FIR filter coefficients as $\hat{\mathbf{d}}_m(i) = \hat{\mathbf{w}}_m^H(i)\mathbf{r}_A(i)$. It is well known that the LMMSE chip equalizer coefficients are given by minimizing the MSE between the transmitted and recovered chip samples as

$$\begin{aligned} \hat{\mathbf{w}}_m^{opt}(i) &= \arg \min_{\hat{\mathbf{w}}_m(i)} \mathrm{E}[||\mathbf{d}_m(i) - \hat{\mathbf{w}}_m^H(i)\mathbf{r}_A(i)||^2] \\ &= \sigma_d^2(i) \hat{\mathbf{R}}_{rr}(i)^{-1} \hat{\mathbf{h}}_m(i) \end{aligned} \quad (4)$$

where $\sigma_d^2(i)$ is the transmitted chip power. $\hat{\mathbf{R}}_{rr}(i)$ and $\hat{\mathbf{h}}_m(i)$ are the covariance estimation and channel estimation, respectively. Here the covariance matrix is estimated by the time-average with ergodicity assumption as

$$\hat{\mathbf{R}}_{rr}(i) = \mathrm{E}[\mathbf{r}_A(i)\mathbf{r}_A^H(i)] = \frac{1}{N_B} \sum_{i=0}^{N_B - 1} \mathbf{r}_A(i)\ \mathbf{r}_A^H(i) \quad (5)$$

where N_B is the length for the time average. The channel coefficients are estimated as $\hat{\mathbf{h}}_m(i) = \mathrm{E}[\mathbf{r}_A(i)\mathbf{d}_m^H(i)]$ using the pilot symbols. In the HSDPA standard, about 10 % of the total transmit power is dedicated to the Common Pilot Channel (CPICH). This will provide good channel estimation. By assuming that the channel is stationary over the observation window length, we can have a block based operation by omitting the chip index in $\hat{\mathbf{R}}_{rr}(i)$, $\hat{\mathbf{h}}_m(i)$ and $\hat{\mathbf{w}}_m(i)$.

Using the stationarity of the channel and the convolution property, it is shown that the correlation matrix $\mathbf{R}_{rr}$ is a banded block Toeplitz matrix, which is approximated by a block-circulant matrix [7] after we add two corner matrices as

$$\mathbf{C}_{rr} = \mathbf{R}_{rr} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{C}_L^H \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{C}_L & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

where

$$\mathbf{C}_L = \begin{pmatrix} \mathbf{E}^H[L] & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{E}^H[1] & \cdots & \mathbf{E}^H[L] \end{pmatrix}.$$

Here $\mathbf{E}[l]$ is a $N \times N$ block matrix constructed with the correlation coefficients. Using the extension of the diagonalization theorem, the block-circulant matrix can be decomposed as

$$\mathbf{C}_{rr} = \Big(\mathbf{D}^H \otimes \mathbf{I}\Big)\Big(\sum_{i=0}^{L_F - 1} \mathbf{W}^i \otimes \mathbf{E}[i]\Big)\Big(\mathbf{D}^H \otimes \mathbf{I}\Big) \quad (6)$$

where $\mathbf{W} = \mathrm{diag}(1, \mathrm{W}_{\mathrm{L_F}}^{-1} \cdots \mathrm{W}_{\mathrm{L_F}}^{-(\mathrm{L_F}-1)})$ and $W_{L_F} = e^{j(2\pi/L_F)}$is the phase factor coefficient for the DFT computation. $\otimes$ denotes the *Kronecker* product and $\mathbf{D}$ is the DFT matrix. Finally the MIMO equalizer taps are computed from

$$\hat{\mathbf{w}}_m^{opt} \approx (\mathbf{D}^H \otimes \mathbf{I}) \cdot \mathbf{F}^{-1} \cdot (\mathbf{D} \otimes \mathbf{I}) \hat{\mathbf{h}}_m. \quad (7)$$

$\mathbf{F} = \mathrm{diag}(\mathbf{F}_0,\ \mathbf{F}_1, \cdots, \mathbf{F}_{\mathrm{L_F}})$ is a block-diagonal matrix with elements taken from the element-wise FFT of the first column of a circular matrix. For an $(M \times N)$ MIMO system, this reduces the inverse of a $(NL_F \times NL_F)$ matrix to the inverse of sub-block matrices with size $(N \times N)$.

III. Scalable Pipelined VLSI Architecture

To achieve a real-time implementation, either DSP processors or VLSI architectures can be applied. The limited hardware resource and power supply in mobile handsets makes the hardware design more challenging, especially for the MIMO system. Many optimizations are needed to reduce the redundant computation and make it suitable for real-time implementation. We emphasize the interaction between architecture, system partitioning and pipelining with these objectives: 1). Propose further optimization schemes to reduce the computation complexity for efficient VLSI implementation; 2). Design parallel and pipelined architecture for the critical computation blocks.

A. HLS Architecture Scheduling

Field Programmable Gate Array (FPGA) can behave like a number of different ASICs. This makes FPGA a good platform to build, verify and prototype System-on-Chip (SoC) designs quickly. We applied an efficient Catapult C based High-Level-Synthesis (HLS) design methodology to investigate various pipelined architectures and different levels of parallelism. Register-Transfer-Level (RTL) design is generated directly from C/C++ code and imported to the HDL Designer tool for high-level integration. Seamless verification and software hardware co-design is achieved. Catapult C provides architecture scheduling for both block-mode and throughput mode [8] to generate efficient RTL on different resource/timing requirements. Configurable parallelism is achieved by assigning the number of Functional Units (FU) according to area/timing constraints. The best solution would be the smallest design meeting the real-time requirements.

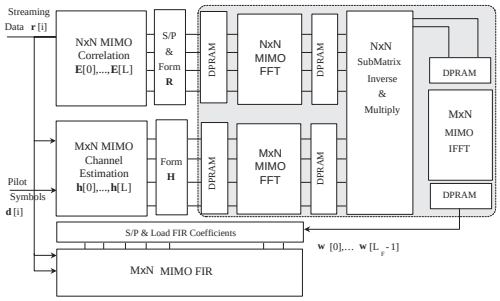

Fig. 1. Top level partitioning of the FFT-based MIMO equalizer architecture.

B. System-level Partitioning

With a timing and data dependency analysis, the top level design blocks for the MIMO equalizer are shown in Fig. 1. The system-level pipelining is designed for better modularity. The overall equalizer receiver includes the tasks in the following procedure:

1) Compute the independent correlation elements $[\mathbf{E}[0] \ \cdots \ \mathbf{E}[L]]$ and form the first block column of circulant $C_{rr}^{(1)}$ by adding the corner elements as $C_{rr}^{(1)} = [\mathbf{E}[0] \ \cdots \ \mathbf{E}[L] \ \mathbf{0} \ \cdots \ \mathbf{0} \ \mathbf{E}^H[L] \ \cdots \ \mathbf{E}^H[1]]^T$. Each element is an $(N \times N)$ sub block matrix.
2) Take the element-wise FFT of $\mathbf{C}_{rr}^{(1)}$, where the element vectors $\mathbf{F}_{n1,n2} = \mathrm{FFT}\{\mathbf{E}_{\mathrm{n1,n2}}^{(c)}\}$ and $\mathbf{E}_{n1,n2}^{(c)}(i) = \mathbf{C}_{rr}^{(1)}[(n_1 - i - 1) * N + n_2 - 1]$, for $i \in [0, L_F]$ and $n_1, n_2 \in [1, N]$.
3) For $m \in [1, M]$, compute the dimension-wise FFT of the channel estimation as $\mathbf{\Phi}_m = (\mathbf{D} \otimes \mathbf{I})\hat{\mathbf{h}}_m = \mathrm{FFT}([0, \cdots, 0, h_{m,n}(L), \cdots, h_{m,n}(0), 0, \cdots, 0])$.
4) Compute the inverse of the $(N \times N)$ sub matrix $\mathbf{F}[i]$, where $\mathbf{F}[i]^{-1} = (\mathbf{F}_{n1,n2}[i])^{-1} = \mathrm{diag}(\mathbf{F}[0]^{-1}, \cdots, \mathbf{F}[L_F - 1]^{-1})$.
5) Compute the matrix multiplication of the sub-matrices inverse with the FFT output of channel estimation coefficients $\mathbf{\Psi}_m = \mathbf{F}^{-1}\mathbf{\Phi}_m$.
6) Compute the dimension-wise IFFT of the matrix multiplication results $\hat{\mathbf{w}}_m^{opt} \approx (\mathbf{D}^H \otimes \mathbf{I})\mathbf{\Psi}_m$.

A correlation estimation block takes the multiple input samples for each chip to compute the correlation coefficients of the first column of $\mathbf{R}_{rr}$. It is made circulant by adding corner to form the matrix $[\mathbf{E}[0], \cdots, \mathbf{E}[L], \mathbf{0}, \cdots, \mathbf{0}, \mathbf{E}[L]^H, \cdots, \mathbf{E}[1]^H]$. The complete coefficients are then written to DPRAMs and the $(N \times N)$ element-wise FFT module computes $[\mathbf{F}[0], \cdots, \mathbf{F}[L_F]] = \mathrm{FFT}[\mathbf{E}[0], \cdots, \mathbf{E}[L], \mathbf{0}, \cdots, \mathbf{0}, \mathbf{E}[L]^H, \cdots, \mathbf{E}[1]^H]$.

Another parallel data path is for the channel estimation and the $(M \times N)$ dimension-wise FFTs on the channel coefficient vectors as in $(\mathbf{D} \otimes \mathbf{I})\hat{\mathbf{h}}_m$. A sub-matrix inverse and multiplication block takes the FFT coefficients of both channels and correlations from DPRAMs and carries out the computation as in $\mathbf{F}^{-1}$. Finally an $(M \times N)$ dimension-wise IFFT module generates the results for the equalizer taps $\hat{\mathbf{w}}_m^{opt}$ and sends them to the $(M \times N)$ MIMO FIR block for filtering. To reflect the correct timing, the correlation and channel estimation modules at the front-end will work in a throughput mode on the streaming input samples. The FFT-inverse-IFFT modules in the dotted line block construct the post-processing of the tap solver. They are suitable to work in a block mode using dual-point RAM blocks to communicate the data. The MIMO FIR filtering will also work in throughput mode on the buffered streaming input data.

C. Hermitian Optimization and Reduced-state FFT

From the circulant feature of the correlation matrix, we can reduce the complexity of the FFT computation with the following Lemma

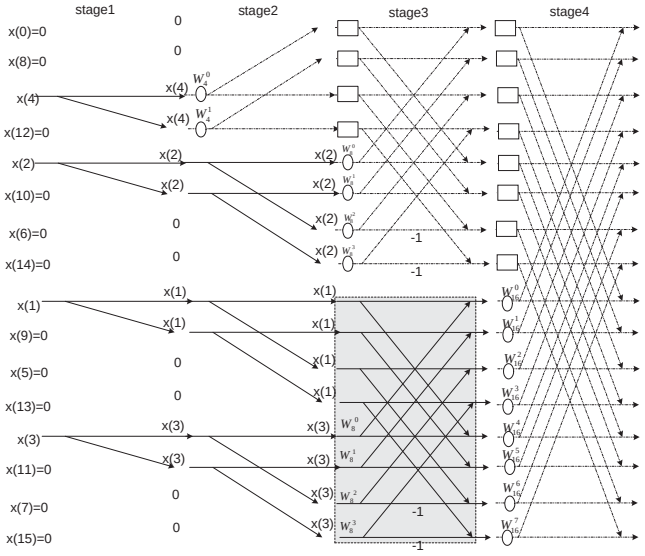

Fig. 2. Reduced-state FFT butterfly tree.

for the MIMO receiver.

Lemma 1 (Hermitian): $\mathbf{F}_{i,j} = \mathrm{conj}(\mathbf{F}_{\mathrm{j,i}})$. Thus the computation of $\mathbf{F}_{j,i}$ is redundant for $j < i$.

Lemma 2 (Hermitian Complexity): Because the imaginary part of $\mathbf{F}_{i,i}$ equals to 0, the computation of $\mathbf{F}_{i,i}$ can be reduced to only L/L_F of the full DFT. The computations related to $\mathbf{F}_{i,i}$ also reduce to real computation, saving 50% of complexity.

Because the FFT algorithm applies the features of the rotation coefficients, the application of the Hermitian feature in Lemma 2 is not straightforward. Thus, we derived the hardware-oriented optimization for the Reduced-State FFT (RS-FFT) with pruning operations based on the standard Decimation-In-Time (DIT) FFT algorithm. We differentiate the different types of butterfly units based on the feature of the output coefficients and prune unnecessary computation branches in the butterfly tree. This is shown in Fig. 2.

IV. Hermitian Matrix Inverse Architectures

In this section, we utilize the Hermitian feature and focus on the optimization of the matrix inverse and multiplication module following the element-wise FFT modules in the block tap solver. Although the FFT-based tap solver avoids the direct matrix inverse of the original correlation matrix with the dimension of $(NF \times NF)$, the inverse of the diagonal matrix $\mathbf{F}$ is inevitable. For a MIMO receiver with high receive dimension, the matrix inverse and multiplication in $\mathbf{F}^{-1}\hat{\mathbf{h}}_m$ is not trivial. Because of the diagonal feature of $\mathbf{F}$ matrix, the inverse of $\mathbf{F}$ can be divided into the inverse of L_F sub-matrices of size $(N \times N)$ as in

$$\mathbf{F}^{-1} = \mathrm{diag}\big(\mathbf{F}[0]^{-1}, \mathbf{F}[1]^{-1}, \cdots, \mathbf{F}[L_F - 1]^{-1}\big). \tag{8}$$

Gaussian elimination or Cholesky decomposition can be applied to inverse these matrices with $\mathcal{O}(N^3)$ complex operations. However, it requires arithmetic square root operations that are preferable to be avoided in hardware due to their complexity. Considering the fact that it is unlikely to have more than four Rx antennas in a mobile terminal, we consider the two special cases individually, i.e., 2 and 4 Rx antennas, with the Hermitian optimization. We propose complexity reduction schemes and efficient architectures suitable for VLSI implementation based on the exploration of block partitioning.

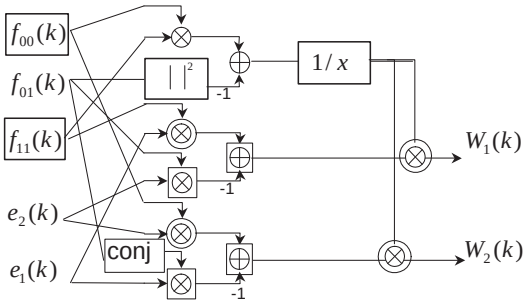

Fig. 3. The data path of merged 2×2 inverse and multiplication.

The commonality of the partitioned block matrix inverse is extracted to design generic RTL modules for reusable modularity. We then build the 4×4 receiver by reusing the 2×2 block partitioning.

A. Dual-antenna MIMO Receivers

From equation (7), a straightforward partitioning is at the matrix inversion for $\mathbf{F}$ and then the matrix multiplication of $\mathbf{F}^{-1}(\mathbf{D} \otimes \mathbf{I})\mathbf{h}_m$ and dimension-wise FFT of the channel coefficients. In this partitioning, we would first compute the inverse of the entire sub-block matrix in $\mathbf{F}$ and then carry out a matrix multiplication. However, this partitioning involves two separate loop structures. Since the two steps have same loop structure, it is more desirable to merge the two steps and reduce the overhead. If the inverse of a 2×2 submatrix is given by

$$\mathbf{F}[k]^{-1} = \begin{pmatrix} f_{00}(k) & f_{01}(k) \\ f_{10}(k) & f_{11}(k) \end{pmatrix}^{-1}$$

and $\mathbf{\Gamma}[k] = [e_1(k)\ e_2(k)]$ is the k^{th} elements of the dimension-wise FFT coefficients $(\mathbf{D} \otimes \mathbf{I})\mathbf{h}_m = [\mathbf{\Gamma}[0]\ \mathbf{\Gamma}[1]\ \cdots\ \mathbf{\Gamma}[L_F - 1]]$, then a merged matrix inverse and multiplication is given by $\mathbf{W} = \mathbf{F}^{-1} \cdot (\mathbf{D} \otimes \mathbf{I})\mathbf{h}_m = [\mathbf{F}[0]^{-1}\mathbf{\Gamma}[0]^T,\ \mathbf{F}[1]^{-1}\mathbf{\Gamma}[1]^T,\ \cdots,\ \mathbf{F}[L_F - 1]^{-1}\mathbf{\Gamma}[L_F - 1]^T]$ with the k^{th} element of the matrix $\mathbf{W}$ as

$$\mathbf{W}[k] = \frac{1}{f_{00}(k)\ \cdot\ f_{11}(k) - |f_{01}(k)|^2} \cdot \begin{pmatrix} f_{11}(k) \circ e_1(k) - f_{01}(k) * e_2(k) \\ -f_{10}(k) \circ e_2(k) - f_{01}(k) * e_1(k) \end{pmatrix}. \quad (9)$$

With the *Hermitian* features of $\mathbf{F}_{00}$ and $\mathbf{F}_{11}$, we can reduce the number of real operations. In the equatoin, "$a \cdot b$" means "$real \times real$" and "$a \circ b$" means "$real \times complex$" while "$a * b$" means "$complex \times complex$" multiplications. The complex division is replaced by a real division. From this, we derived the simplified data path with the Hermitian optimization as in Fig. 3. In this figure, $f_{00}(k)$ and $f_{11}(k)$ are real numbers. The single multiplier means a real multiplication. The multiplier with a circle means the "$real \times complex$" multiplication and the multiplier with a rectangle is a "$complex \times complex$" multiplication. The data path is significantly simplified, which facilitates the scaling in fixed-point implementation and increases the numerical stability. Notice that the storage for the interface from element-wise FFTs is also reduced. We save four distributed DPRAMs for the following real and imaginary parts $\Im(f_{00}), \Im(f_{11}), \Re(f_{10}), \Im(f_{10})$.

B. Receiver with 4 Rx Antennas

This includes the $1 \times 4,\ 2 \times 4,\ 4 \times 4$ SIMO and MIMO scenarios. Note that the receiver diversity by over-sampling also has the same mathematic format. So this may also be the case of two receive antennas with an over-sampling factor of 2. The principle operation

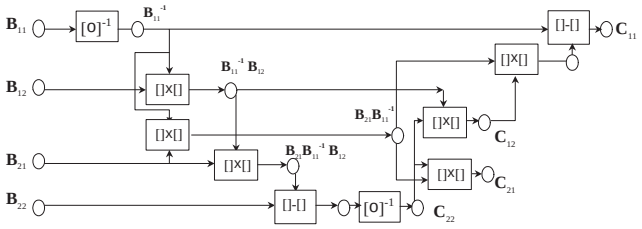

Fig. 4. The data dependency path of the partitioned 4×4 matrix inverse.

of interest is the inverse of the 4×4 matrices, so it is necessary to determine which range of possible matrix architectures is most suitable to this application. In addition to minimizing the circuit area used, the design needs to work within a short time budget. We need to derive efficient computing architecture for this part to save the area and time resources.

We partition the 4×4 sub matrices and its inverse in $\mathbf{F}[i]$ into four 2×2 block sub matrices as

$$\mathbf{F}(i) = \begin{pmatrix} \mathbf{B}_{11}(i) & \mathbf{B}_{12}(i) \\ \mathbf{B}_{21}(i) & \mathbf{B}_{22}(i) \end{pmatrix} \Rightarrow \mathbf{F}(i)^{-1} = \begin{pmatrix} \mathbf{C}_{11}(i) & \mathbf{C}_{12}(i) \\ \mathbf{C}_{21}(i) & \mathbf{C}_{22}(i) \end{pmatrix}$$

Then we apply an partitioned inverse of the 4×4 matrix from the inverse of 2×2 sub-matrices. It can be shown that the subblocks are given by the following equations:

$$\begin{cases} \mathbf{C}_{22}(i) = [\mathbf{B}_{22}(i) - \mathbf{B}_{21}(i)\mathbf{B}_{11}(i)^{-1}\mathbf{B}_{12}(i)]^{-1} \\ \mathbf{C}_{12}(i) = -\mathbf{B}_{11}(i)^{-1}\mathbf{B}_{12}(i)\mathbf{C}_{22}(i) \\ \mathbf{C}_{21}(i) = -\mathbf{C}_{22}(i)\mathbf{B}_{21}(i)\mathbf{B}_{11}(i)^{-1} \\ \mathbf{C}_{11}(i) = \mathbf{B}_{11}(i)^{-1} - \mathbf{C}_{12}(i)\mathbf{B}_{21}(i)\mathbf{B}_{11}(i)^{-1} \end{cases} \quad (10)$$

Without looking into the data dependency, a straightforward computation will have 8 complex matrix multiplications, 2 complex matrix inverses and 2 complex matrix subtractions, all of the size 2×2. But this is not very efficient. By examining the data dependency, we will find some duplicate operations in the data path. Now we utilize the Hermitian feature of the $\mathbf{F}$ matrix to derive more parallel and optimized computing architecture. Since the inverse of a Hermitian matrix is Hermitian, this leads to the data path by removing the duplicate computation blocks that has the Hermitian relationship.

However, the straightforward treatment still does not lead to the most efficient computing architecture. The data path is still constructed with a very long dependency path. To fully extract the commonality and regulate the design blocks in VLSI, we define the following special operators on the 2×2 matrices for the different type of complex number computations. These special operators will be mapped to VLSI Processing Units (PU) to deal with the special features of the Hermitian matrix. High-level modularity is achieved by extracting the commonality among the data path.

Define 1 (Pseudo-Power): $pPow(a, b) = \Re(a) \cdot \Re(b) + \Im(a) \cdot \Im(b)$ is defined as the *pseudo-power* function of two complex numbers and $\Re(a, b) = \Re(a) \cdot \Re(b) - \Im(a) \cdot \Im(b)$ is defined as the real part of a complex multiplication.

Define 2 (Complex-Hermitian-Multiplication): For a general 2×2 matrix $\mathbf{A}$ and a Hermitian 2×2 matrix $\mathbf{B} = \mathbf{B}^H$, we define the operator *Complex-Hermitian-Multiplication (CHM)* as

$$M(\mathbf{A}, \mathbf{B}) = \mathbf{AB} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{21}^* \\ b_{21} & b_{22} \end{pmatrix}. \quad (11)$$

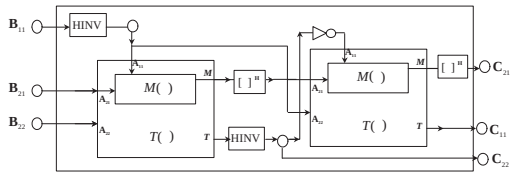

Fig. 5. Simplified partitioned 4×4 inverse with Hermitian optimization.

Define 3 (Hermitian Inverse): For a 2×2 Hermitian matrix $\mathbf{B} = \mathbf{B}^H$, define the *Hermitian Inverse(HInv)* operator as

$$\mathrm{HInv}(\mathbf{B}) = \frac{1}{\mathrm{b}_{11}\ \mathrm{b}_{22} - |\mathrm{b}_{21}|^2} \begin{pmatrix} b_{22} & -b_{21}^* \\ -b_{21} & b_{11} \end{pmatrix}. \quad (12)$$

Define 4 (Diagonal Transform): Given the 4×4 Hermitian $\mathbf{A}$ which is divided into four subblocks as

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{21}^H \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} = \mathbf{A}^H,$$

the *Diagonal Transform (DT)* of $\mathbf{A}$ is defined as,

$$T(\mathbf{A}_{11}, \mathbf{A}_{21}, \mathbf{A}_{22}) = \mathbf{A}_{22} - M(\mathbf{A}_{21}, \mathbf{A}_{11})\mathbf{A}_{21}^H. \quad (13)$$

With these definitions, we regulate the inverse of the 4×4 Hermitian matrix $\mathbf{F} = \mathbf{F}^H$ into simplified operations on 2×2 matrices. After some manipulation, the partitioned subblock computation equations can be mapped to the following procedure using the defined operators.

$$\begin{cases} \mathbf{B}_{inv} = \mathrm{HInv}(\mathbf{B}_{11}) = \mathbf{B}_{inv}^H; \\ \mathbf{D} = M(\mathbf{B}_{21}, \mathbf{B}_{inv}); \\ \mathbf{C}_{22} = \mathrm{HInv}(T(\mathbf{B}_{inv}, \mathbf{B}_{21}, \mathbf{B}_{22})); \\ \mathbf{C}_{12} = -M(\mathbf{D}^H, \mathbf{C}_{22}); \\ \mathbf{C}_{11} = \mathbf{B}_{inv} + \mathbf{D}^H\mathbf{C}_{22}\mathbf{D} = T(-\mathbf{C}_{22}, \mathbf{D}^H, \mathbf{B}_{inv}). \end{cases}$$

Finally this leads to the much simplified hardware mapping using the generic Processing Units in Fig.5. From the figure, it is clear that the overall computation complexity is 2 HInv operations, 2 DTs, 1 extra CHM block. Because the sign inverter and the Hermitian formatter $[\ ]^H$ has no hardware resource at all, the computation complexity is determined by the three generic blocks. The data path of the computation shows the timing relationship between different design modules.

C. Parallel Architecture Modules

This regulated block diagram facilitates the design of efficient parallel VLSI modules. To extract the commonality and reduce the redundancy, we need to explore the timing relationship of the basic computations involved in generic operations. Because the operation *M* is also embedded in the *T* transform, we design the interface in a way that the duplicate computations are removed and the efficient computing architecture is reused. The grouping of computations and the smart usage of interim registers will eliminate the redundancy and give simple and generic interface to the design modules. Finally the simplified parallel $M(\mathbf{A}, \mathbf{B})$ RTL module can be design as Fig. 6, with the input of both real and imaginary parts of $\mathbf{A}$ as $\{a_{11}(r/i), a_{12}(r/i), a_{21}(r/i), a_{22}(r/i)\}$ and only the necessary elements of the *Hermitian* matrix $\mathbf{B}$ as in $\{b_{11}(r), b_{21}(r/i), b_{22}(r)\}$. The output ports include $\{tmp_1, tmp_2, tmp_3, tmp_5, tmp_6, tmp_7\}$. There is no need to output tmp_4 and tmp_8. Moreover, although all these numbers have complex values, the products with the input values do not need to be complex multiplications. We also only need to compute d_{21} and d_{22} to get the $\mathbf{G}$ elements. The redundant computations in $\{tmp_4, tmp_8, d_{11}, d_{12}\}$ are eliminated from the $M(\mathbf{A}, \mathbf{B})$ operation. Built from the simplified $M(\mathbf{A}, \mathbf{B})$ module, the data path RTL module of the transform $T(\mathbf{A}_{11}, \mathbf{A}_{21}, \mathbf{A}_{22})$ of the 4×4 Hermitian matrix is given by Fig. 7, with the simplified Functional Components $\{pPow(a, b), \Re(a, b)\}$ as defined. The output ports of the $T(\mathbf{A}_{11}, \mathbf{A}_{21}, \mathbf{A}_{22})$ include the independent elements $\{t_{11}, t_{21}, t_{22}\}$.

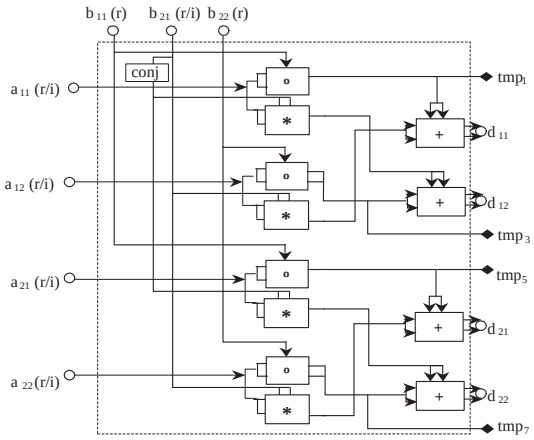

Fig. 6. Parallel VLSI RTL layout of the $M(\mathbf{A}, \mathbf{B})$ processing unit.

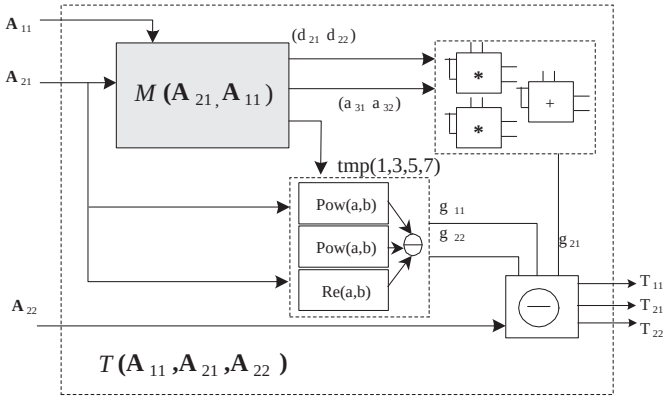

Fig. 7. The VLSI RTL layout of the $T(\mathbf{A}_{11}, \mathbf{A}_{21}, \mathbf{A}_{22})$ block.

V. Experimental FPGA Implementation

The performance is evaluated in an HSDPA simulation chain for different MIMO antenna configurations. The readers are referred to [7] for the algorithmic Bit-Error-Rate performance. Based on the above algorithmic and architectural optimizations, we have designed the VLSI architecture and prototyped the RTL design on the *Nallatech* FPGA platform. The chip rate is in accordance with the WCDMA chip rate at $3.84MHz$. We applied a clock rate of $38.4MHz$ for the *Xilinx Virtex-II V6000* FPGA. The correlation window is set to 10 chips for all 4 receive antennas. The FFT size is 32-points. In the following, we give the specification of the major design blocks: the throughput-mode correlation calculation, the multiple FFT/IFFT modules and the L_F inverse of 4×4 submatrices. We utilize a Catapult C based design methodology to study many area/time tradeoffs of the VLSI architecture design. For example, for the 16-FFT/IFFT modules, at one extreme, we can design a fully parallel and pipelined architecture with parallel butterfly-units and complex

TABLE I

THE AREA/TIME SPECIFICATION OF THE MAJOR FPGA DESIGN MODULES.

Architecture	latency	CLB	ASICMult
Correlator	1 chip	22399	80
16-FFT32	$44\mu s$	2530	4
32 MatInvMult(4×4)	$38\mu s$	4526	6

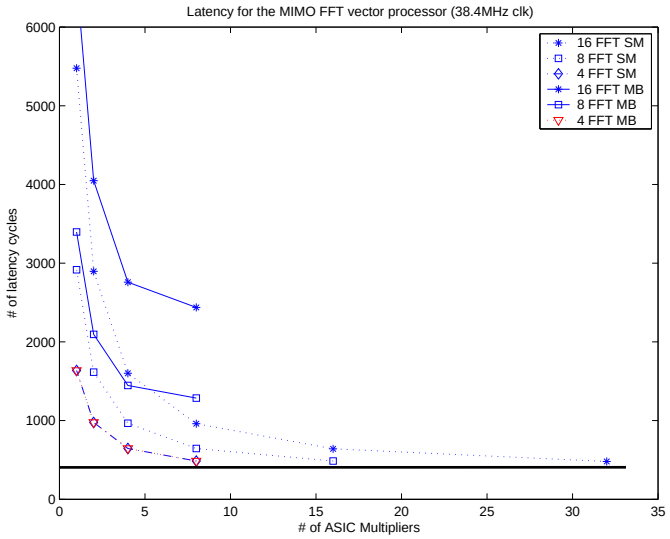

Fig. 8. Latency vs. # of multipliers for the merged MIMO-FFT module.

multipliers laid out in the pattern of butterfly-tree. Since it is economically desirable to reduce the area, this is not practical. We design the merged multiple input multiple output FFT modules to utilize the commonality in control logic and phase coefficient loading. Overall, we can utilize only 4 multipliers to achieve area/time efficient design for this module. For the L_F inverse of 4×4 Hermitian matrix, the latency is 38 μs with 6 multipliers. This benefits from the aforementioned architectural optimization. Moreover, to achieve power-saving, scalability using different functional units are also explored extensively for different type of architectures, such as shown in Fig. 8 for the MIMO-FFT modules with different latency. Although such detail is out of the scope of focus in this paper due to the limited space, we demonstrate the architectural scalability using the Catapult C scheduling methodology.

VI. CONCLUSION

In this paper, we propose a MIMO receiver architecture for a circulant LMMSE chip equalizer for the CDMA downlink systems. Hermitian optimization and partitioned sub-matrix inverse is proposed to construct parallel architecture for the 4×4 MIMO receiver using the commonality. The much simplified parallel and pipelined RTL is design using Catapult C HLS flow and verified in an FPGA prototyping platform.

ACKNOWLEDGMENT

The authors would like to thank Dr. Charlie Zhang and Dr. Behnaam Aazhang and the anonymous reviewers for their instructive feedback to improve the paper. Dr. Cavallaro is supported in part by NSF under grants ANI-9979465 and EIA-0224458 and EIA-0321266.

REFERENCES

[1] G. J. Foschini,*Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas*, Bell Labs Tech. J., pp. 41-59, 1996.

[2] G. D. Golden, J. G. Foschini, R. A. Valenzuela and P. W. Wolniansky, *Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture*, Electron. Lett., vol. 35, pp.14-15, Jan. 1999.

[3] A. Wiesel, L. Garca, J. Vidal, A. Pags and Javier R. Fonollosa, *Turbo linear dispersion space time coding for MIMO HSDPA systems*, 12^{th} IST Summit on Mobile and Wireless Communications, Aveiro, Portugal, June 15-18, 2003.

[4] K. Hooli, M. Juntti, M. J. Heikkila, P. Komulainen, M. Latva-aho and J. Lilleberg, *Chip-level channel equalization in WCDMA downlink*, EURASIP Journal on Applied Signal Processing, pp. 757-770, Aug.2002.

[5] P. Radosavljevic, J. R. Cavallaro and A. D. Baynast, *Implementation of channel equalization for MIMO systems in WCDMA downlink*, in proceeding of VTC Fall 2004, Vol. 3, pp. 1735- 1739, Los Angeles, CA, Sept. 2004.

[6] Y. Guo, J. Zhang, D. McCain, J. R. Cavallaro, *Scalable FPGA architectures for LMMSE-based SIMO chip equalizer in HSDPA downlink*, 37^{th} IEEE Asilomar Conference on Signals, Systems and Computers, Vol. 2, pp. 2171-2175, Monterey, CA, Nov. 9-12, 2003.

[7] Y. Guo, J. Zhang, D. McCain and J. R. Cavallaro, *Efficient MIMO equalization for downlink multi-code CDMA: complexity optimization and comparative study*, in IEEE Globecom'04, Vol. 4, pp. 2513 - 2519, Dallas, TX, Nov. 28th- December-2nd, 2004.

[8] Y. Guo, G. Xu, D. McCain, J. R. Cavallaro, *Rapid scheduling of efficient VLSI architectures for next-generation HSDPA wire-less system using Precision-C synthesizer*, Proceding of IEEE Intl. Workshop on Rapid System Prototyping'03, San Diego, CA, pp. 179-185, June 2003.