

Title	Efficient Equalization Hardware Architecture for SC-FDMA Systems without Cyclic Prefix
Author(s)	Ferdian, Rian; Anwar, Khoirul; Adiono, Trio
Citation	2012 International Symposium on Communications and Information Technologies (ISCIT): 936-941
Issue Date	2012-10
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/10897
Rights	This is the author's version of the work. Copyright (C) 2012 IEEE. 2012 International Symposium on Communications and Information Technologies (ISCIT), 2012, 936-941. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Description	

Efficient Equalization Hardware Architecture for SC-FDMA Systems without Cyclic Prefix

Rian Ferdian*, Khoiril Anwar**, and Trio Adiono*

*School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB),
Jl. Ganesha 10, Bandung, Indonesia 40132

E-mail: rian.ferdian@students.ee.itb.ac.id, tadiono@stei.itb.ac.id

**School of Information Science, Japan Advanced Institute of Science and Technology (JAIST)
Asahidai 1-1, Nomi, Ishikawa, Japan 923-1211, E-mail: anwar-k@jaist.ac.jp

Abstract—Single carrier frequency domain multiple access (SC-FDMA) system achieves better spectral efficiency when cyclic prefix (CP) is not transmitted. However, the chained turbo equalization (CHATUE) algorithm, an equalizer required to both equalize the multipath fading effect and cancel the inter-block interference of SC-FDMA without CP, requires high computational complexity due to the non-circulant structure of the past and the future interference matrices. This paper proposes efficient hardware architecture based on *systolic architecture* for practical implementation. The main idea is to minimize the number of required processing elements by utilizing efficient resources sharing method while exploiting concurrency of the processing. A new computation method using *masking matrices* is introduced to obtain interference matrix from its corresponding circulant channel matrix. The results with fixed point computation show that the computational complexity can be significantly reduced up to 96% for practical implementation without significant degradation in bit-error-rate (BER) performances.

I. INTRODUCTION

The need for a wireless transmission system with more efficient spectrum rises along with the significant growth in the number of wireless communication users. Transmission system without cyclic prefix (CP) or guard interval (GI) is one of the solutions to achieve a better efficiency in spectrum. However, the absence of CP or GI causes additional interference, i.e., inter-block interference (IBI) from past and future blocks, beside the inter-symbol interference (ISI) due to the multipath fading effect.

Chained turbo equalization (CHATUE) algorithm offers a low computational complexity method to equalize the both ISI and IBI in block transmission systems without CP or GI [1]. The detail analysis and advantages over the standard block transmission systems with CP as well as its impact due to the Doppler effect has also been investigated in [2]. Furthermore, for practical broadband application, CHATUE algorithm is applied to SC-FDMA systems in [3] and [4]. In order to minimize the computational complexity, CHATUE algorithm utilizes matrix \mathbf{J} to form a circulant structure of the (current) channel matrix. However, the introduction of matrix \mathbf{J} is still

This research has been supported in part by Chubu Electric Co. Ltd., (Kinki) Mobile Radio Center (MRC), Inc., Japan and in part by Ministry of Telecommunications and Informatics, Indonesia. This research is conducted when the first author was a visiting research student in JAIST, Ishikawa, Japan.

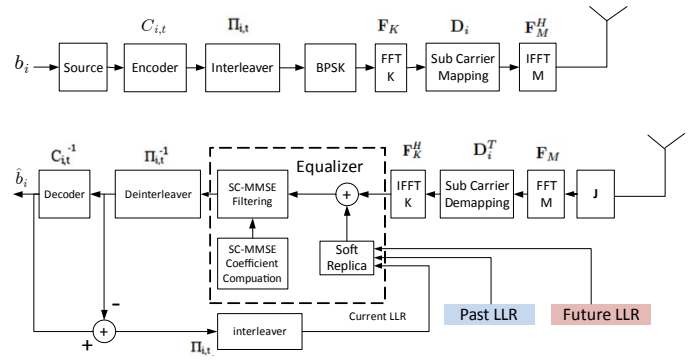


Fig. 1. Transceiver structure for CHATUE-SC-FDMA systems

limited *only* to the current channel matrix, while the past and the future matrix, referred to as *interference matrices*, are still with non-circulant structure which cause the computation complexity is still high.

Ref. [4] proposes a new method of approximation for the covariance matrix inversion \mathbf{X}^{-1} , however, the non-circular structure of the past and the future matrices still make the computational complexity high in the calculation of matrix \mathbf{X} itself. The computation of matrix \mathbf{X} requires frequency domain covariances of the matrices, where the fast Fourier transform (FFT) may be required. In addition, non-diagonal matrix-and-matrix multiplications are required to obtain the covariance values.

In this paper, we propose a semi-optimal CHATUE-SC-FDMA algorithm, as shown in Fig. 1, with reduced computational complexity and optimal hardware resource sharing. The proposed algorithm can obtain the frequency response of the past and the future interference matrix without FFT operations. We design CHATUE-SC-FDMA systems based on systolic architecture [5] to significantly reduce the computational complexity in hardware implementation. The optimal resource sharing and *sinc* function approximation for systolic coefficient are utilized to suppress the number of required processing elements. Performances of the proposed hardware architecture are assessed in fixed-point simulations in terms of average bit-error-rate (BER).

II. SYSTEM MODEL

In this paper, we assume CHATUE-SC-FDMA systems without doped-accumulator (DA) [3] with a block diagram as shown in Fig. 1. At the transmitter, the information bits for the i -th user at t -th block are encoded by $C_{i,t}$, interleaved by $\Pi_{i,t}$, and modulated through K -points FFT (\mathbf{F}_K), sub-carrier mapped and M -point inverse FFT (\mathbf{F}_M^H) to produce vector signal $\mathbf{s}_{i,t}$, $\mathbf{s}'_{i,t-1}$, and $\mathbf{s}''_{i,t+1}$. The notations $(\bullet)'$ and $(\bullet)''$ indicate the past and the future blocks relative to current block, respectively.

The SC-FDMA blocks is transmitted without CP over the multi-path block Rayleigh fading channel.¹ At the receiver, the received signal is affected by three channel matrices and four interference matrices. For the current block, the computation involves the current channel matrix ($\mathbf{H}_{i,t}$), the past interference matrix from past block ($\mathbf{H}'_{i,t-1}$), and future interference matrix from future block ($\mathbf{H}''_{i,t+1}$). The details structure of the channel and interference matrices are discussed in [2].

We assume a perfect user sub-carrier mapping such that the interference from other users is negligible and hence, the received signal for the i -th user can be expressed as

$$\begin{aligned} \mathbf{r}_{i,t} = & \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}_{i,t} \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \mathbf{s}_{i,t} \\ & + \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}'_{i,t-1} \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \mathbf{s}'_{i,t-1} \\ & + \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}''_{i,t+1} \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \mathbf{s}''_{i,t+1} + \mathbf{n}, \end{aligned} \quad (1)$$

where \mathbf{D}_i represents the sub-carrier mapping matrix, \mathbf{D}_i^T represents sub-carrier de-mapping matrix with $(\bullet)^T$ denotes a matrix transpose operation and \mathbf{n} is the additive white Gaussian noise vector with variance of σ_n^2 .

In this paper, \mathbf{D}_i is assumed to be the same over the current, the past and the future blocks for each user. $\mathbf{s}_{i,t}$ is Binary phase shift keying (BPSK) modulated block with in total of 4 users and $M = 512$.² The total number of path is 20 and 64 with equal average power. The encoder $C_{i,t}$ is a very simple memory 1 convolutional code with generator polynomial of $G = [3, 2]_8$.³ A fixed point model is used to observe the bit-width required by each variable in the CHATUE-SC-FDMA computation. The results are used as the baseline to define the hardware's bit specification.

III. OPTIMIZED CHATUE-SC-FDMA ALGORITHM

The computations of CHATUE-SC-FDMA algorithm can be divided into three parts: soft cancellation, SC-MMSE coefficients computation, and SC-MMSE filtering. As noted in Section I, the circulant structure of interference matrices is not achieved even with matrix \mathbf{J} multiplication. As the solution, we introduce the use of *column* and *row masking matrix* \mathbf{M}_C and \mathbf{M}_R , respectively, to retrieve interference matrix from its

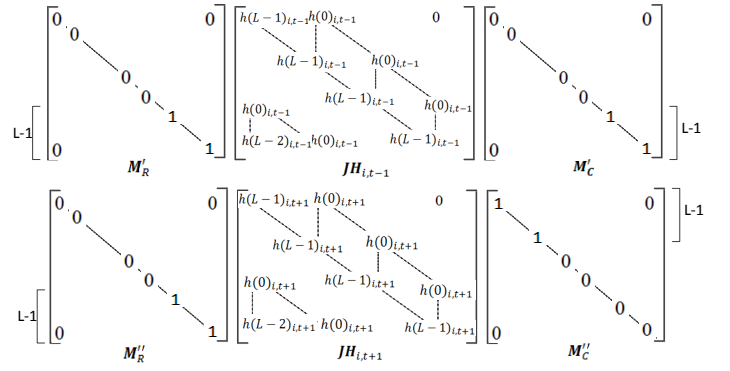


Fig. 2. Structure of column and row masking matrices

corresponding channel matrix. The mathematical derivation for past interference matrix is described as

$$\mathbf{J} \mathbf{H}'_{i,t-1} = \mathbf{M}'_R \mathbf{J} \mathbf{H}_{i,t-1} \mathbf{M}'_C, \quad (2)$$

and for the future interference matrix as

$$\mathbf{J} \mathbf{H}''_{i,t+1} = \mathbf{M}''_R \mathbf{J} \mathbf{H}_{i,t+1} \mathbf{M}''_C. \quad (3)$$

The structure of column and row masking matrices are shown in Fig. 2, where L is the number of path in the channel. From the figure it can be noted that \mathbf{M}'_R , \mathbf{M}'_C and \mathbf{M}''_R are having the same structure.

A. Soft Cancellation

Soft cancellation takes a major part in ISI and IBI components removal. The equalizer creates replica of the received signal based on *a priori* information provided by the decoder as well from its neighbouring decoders in the form of log-likelihood ratio (LLR). In total, there are three LLR are involved in the soft cancellation computation, *a posteriori* LLR $L'_{p,c_{i,t-1}}$ from the past decoder, *a posteriori* LLR $L''_{p,c_{i,t+1}}$ from future decoder, and *extrinsic* LLR $L_{e,c_{i,t}}$ from the current decoder. The details of soft-estimate $\hat{\mathbf{s}}_{i,t}$ for CHATUE-SC-FDMA have been described in [4].

By using the channel matrices (from the channel estimator), the soft replica $\hat{\mathbf{r}}_{i,t}$ of the received signal $\mathbf{r}_{i,t}$ is obtained as

$$\begin{aligned} \hat{\mathbf{r}}_{i,t} = & \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}_{i,t} \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \hat{\mathbf{s}}_{i,t} + \\ & \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}'_{i,t-1} \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \hat{\mathbf{s}}'_{i,t-1} + \\ & \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}''_{i,t+1} \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \hat{\mathbf{s}}''_{i,t+1}. \end{aligned} \quad (4)$$

Since $\mathbf{J} \mathbf{H}_{i,t}$ is circulant, the matrix $\ddot{\mathbf{H}}_{i,t}$, the frequency response or current channel matrix,

$$\ddot{\mathbf{H}}_{i,t} = \mathbf{F}_M \mathbf{J} \mathbf{H}_{i,t} \mathbf{F}_M^H. \quad (5)$$

is a diagonal matrix. Computational complexity for this matrix is as simple as FFT vector problem for $h_t^{[1]}$, where $h_t^{[1]}$ is the first column vector of the circulant current channel matrix $\mathbf{J} \mathbf{H}_{i,t}$ [7].

Because the channel frequency response of the past and future interference matrices do not reduce the computation

¹The channel gains remain the same in a block.

²Extension to higher order modulations such as QPSK or 64-QAM is straightforward.

³The decoder is Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm with log-max operation (the exponent is taken into account) [6].

complexity, we apply the masking matrices technique in (2) and (3) to compute the past and future parts in (4). For the past block, we can modify as

$$\begin{aligned} & \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}'_{i,t-1} \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \\ &= \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{M}'_R \mathbf{J} \mathbf{H}'_{i,t-1} \mathbf{M}'_C \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \\ &= \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{F}_M \mathbf{M}'_R \mathbf{F}_M^H \mathbf{F}_M \mathbf{J} \mathbf{H}'_{i,t-1} \mathbf{F}_M^H \mathbf{F}_M \mathbf{M}'_C \mathbf{F}_M^H \mathbf{D}_i \mathbf{F}_K \\ &= \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}'_{i,t-1} \mathbf{M}'_C \mathbf{D}_i \mathbf{F}_K, \end{aligned} \quad (6)$$

where \mathbf{M}'_C and \mathbf{M}'_R are the response frequency of masking matrix as

$$\mathbf{M}'_R = \mathbf{F}_M \mathbf{M}'_R \mathbf{F}_M^H, \quad (7)$$

and

$$\mathbf{M}'_C = \mathbf{F}_M \mathbf{M}'_C \mathbf{F}_M^H, \quad (8)$$

respectively.⁴

Eqs. (7) and (8) can be considered as filter operator by observing the structure of masking matrices from Fig. 2. Furthermore, it can also be noted that \mathbf{M}'_R , \mathbf{M}'_C , and \mathbf{M}'_R act as high-pass filter, while \mathbf{M}'_C as low-pass filter. The diagonal and real property of the masking matrices provides a circulant and hermitian structure of its frequency response. Finally, (4) for soft replica computation can be re-expressed as

$$\begin{aligned} \hat{\mathbf{r}}_{i,t} &= \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{H}_{i,t} \mathbf{D}_i \mathbf{F}_K \hat{\mathbf{s}}_{i,t} + \\ & \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}_{i,t-1} \mathbf{M}'_C \mathbf{D}_i \mathbf{F}_K \hat{\mathbf{s}}'_{i,t-1} + \\ & \mathbf{F}_K^H \mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}_{i,t+1} \mathbf{M}'_C \mathbf{D}_i \mathbf{F}_K \hat{\mathbf{s}}''_{i,t+1}. \end{aligned} \quad (9)$$

Here, the matrices $\mathbf{H}_{i,t-1}$ and $\mathbf{H}_{i,t+1}$ can be get from equalizer past or equalizer future when computing their current block. Hence, we obtain the benefit that the M points FFT computation can be avoided for past and future part computation by replacing them with two filter operation from the masking matrix. Finally, the soft cancellation is performed by subtracting the received data with the its soft replica as

$$\tilde{\mathbf{r}}_{i,t} = \mathbf{r}_{i,t} - \hat{\mathbf{r}}_{i,t}, \quad (10)$$

where residue $\tilde{\mathbf{r}}_{i,t}$ is further minimized using SC-MMSE filter.

B. SC-MMSE Filters Coefficients

The SC-MMSE filter should update the coefficients according its data input, where coefficients is

$$\begin{aligned} X &\approx \mathbf{\Lambda}_{i,t} \mathbf{\Phi}_{i,t} \mathbf{\Phi}_{i,t}^H + \mathbf{\Lambda}'_{i,t-1} \frac{1}{K} \text{tr}(\mathbf{\Phi}'_{i,t-1} \mathbf{\Phi}'_{i,t-1}) \mathbf{I}_K \\ &+ \mathbf{\Lambda}''_{i,t+1} \frac{1}{K} \text{tr}(\mathbf{\Phi}''_{i,t+1} \mathbf{\Phi}''_{i,t+1}) \mathbf{I}_K \\ &+ \sigma_i^2 \frac{1}{K} \text{tr}(\mathbf{D}_i^T \mathbf{J} \mathbf{J}^T \mathbf{D}_i) \mathbf{I}_K. \end{aligned} \quad (11)$$

where, σ_i is the noise variance for the i -th user as

$$\sigma_i^2 = \frac{K}{M} \sigma_n^2. \quad (12)$$

The covariance matrix of modulation level is defined as

$$\mathbf{\Lambda}_{i,t} = \text{diag}\{(1 - |\hat{\mathbf{s}}_{i,t}|^2) \mathbf{I}\}, \quad (13)$$

⁴Modification for future block is not shown here due to the space limitation.

and the equivalent frequency domain channel matrix is

$$\mathbf{\Phi}_{i,t} = \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}_{i,t} \mathbf{F}_M^H \mathbf{D}_i. \quad (14)$$

To minimize the computational complexity for the covariance of interference matrices, similarly we apply (2) and (3) to $\mathbf{\Phi}'_{i,t-1}$ and $\mathbf{\Phi}''_{i,t+1}$ as

$$\begin{aligned} \mathbf{\Phi}'_{i,t-1} &= \mathbf{D}_i^T \mathbf{F}_M \mathbf{J} \mathbf{H}_{i,t} \mathbf{F}_M^H \mathbf{D}_i \\ &= \mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}_{i,t} \mathbf{M}'_C \mathbf{D}_i. \end{aligned} \quad (15)$$

By carefully observing (15), the covariance matrix computation for the past and the future interference matrix still requires heavy matrix-matrix multiplications. However, we found that only the diagonal parts of the covariance matrix that makes effect to matrix \mathbf{X} due to trace operator. Here, we exploit the cyclic shifts property of trace operation to minimize the required computation by modifying

$$\begin{aligned} & \text{tr}(\mathbf{\Phi}'_{i,t-1} \mathbf{\Phi}'_{i,t-1}) \\ &= \text{tr}(\mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}_{i,t-1} \mathbf{M}'_C \mathbf{D}_i \mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}_{i,t-1} \mathbf{M}'_C \mathbf{D}_i) \\ &= \text{tr}(\mathbf{M}'_C \mathbf{D}_i \mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}_{i,t-1} \mathbf{M}'_C \mathbf{D}_i \mathbf{D}_i^T \mathbf{M}'_R \mathbf{H}_{i,t-1}) \\ &= \text{tr}(\mathbf{P} \mathbf{H}_{i,t-1} \mathbf{P} \mathbf{H}_{i,t-1}) \\ &= \text{tr}(\text{diag}(\mathbf{P} \mathbf{H}_{i,t-1} \mathbf{P}) \mathbf{H}_{i,t-1}), \end{aligned} \quad (16)$$

where

$$\mathbf{P} = \mathbf{M}'_C \mathbf{D}_i \mathbf{D}_i^T \mathbf{M}'_R. \quad (17)$$

Using \mathbf{P} , let's define

$$\mathbf{W} = \text{diag}(\mathbf{P} \mathbf{H}_{i,t-1} \mathbf{P}). \quad (18)$$

Matrix \mathbf{P} can be pre-computed, the element of which is a constant that varies depends on the sub-carrier mapping matrix, because matrix \mathbf{P} inherits Hermitian and circulant structure from matrix \mathbf{M}'_C and \mathbf{M}'_R . Consequently, diagonal elements of matrix \mathbf{W} can be derived as

$$\begin{aligned} \mathbf{W}(k, k) &= \sum_{i=1}^M \mathbf{P}(i, k) \mathbf{P}(k, i) \mathbf{H}_{i,t-1}(i, i) \\ &= \sum_{i=1}^M |\mathbf{P}(i, k)|^2 \mathbf{H}_{i,t-1}(i, i) \\ &= \sum_{i=1}^M |\mathbf{P}([i-K+1] \bmod M, 1)|^2 \mathbf{H}_{i,t-1}(i, i) \end{aligned} \quad (19)$$

Eq. (19) shows that the computation for diagonal matrix \mathbf{W} can be performed by circular convolution of vector \mathbf{p} which is a vector taken from the first column of matrix \mathbf{P} .

C. SC-MMSE Filtering

The final output of SC-MMSE filter is [2]

$$\mathbf{z}_{i,t} = (\mathbf{I}_K + \mathbf{\Gamma}_{i,t} \mathbf{S}_{i,t})^{-1} [\mathbf{\Gamma}_{i,t} \hat{\mathbf{s}}_{i,t} + \mathbf{F}_K^H \mathbf{\Phi}_{i,t}^H \mathbf{X}^{-1} \mathbf{F}_K \tilde{\mathbf{r}}_{i,t}] \quad (20)$$

with $\mathbf{\Gamma}_{i,t}$

$$\mathbf{\Gamma}_{i,t} = \text{diag}[\mathbf{F}_K^H \mathbf{\Phi}_{i,t}^H \mathbf{X}^{-1} \mathbf{\Phi}_{i,t} \mathbf{F}_K], \quad (21)$$

and matrix $\mathbf{S}_{i,t}$ of

$$\mathbf{S}_{i,t} = \text{diag}|\hat{\mathbf{s}}_{i,t}|^2 \times \mathbf{I}_K, \quad (22)$$

where \mathbf{I}_K is the $K \times K$ size identity matrix.

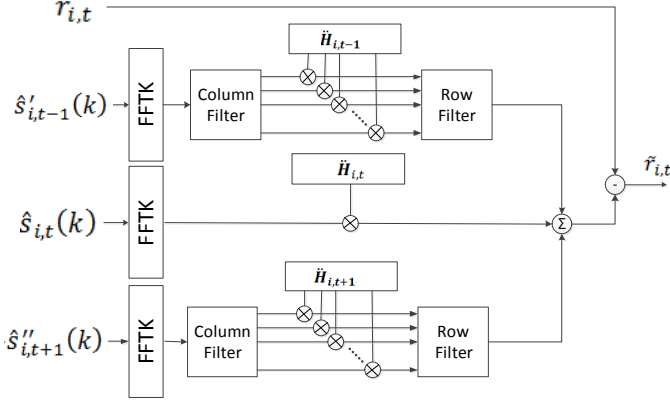


Fig. 3. Soft cancellation block diagram

IV. HARDWARE ARCHITECTURE

The proposed algorithm presented in section III contributes very efficient computation where only diagonal and circulant matrix are involved. Furthermore, the circulant matrices can be further simplified by performing cyclic convolution, while the diagonal matrices can be implemented with point wise multiplications.

A. Hardware Architecture for Soft Cancellation

Based on the results in (9) and (10), the block diagram for soft cancellation can be designed as in Fig. 3. The channels frequency response is assumed unchanged in each iteration, and hence it can be pre-computed. The computation of the current block consists only the diagonal matrix multiplication, which can be implemented with a single multiplier, as the FFTK's output data in series. On the other hand, the computations for the past and the future parts consist of two cyclic convolution from row/column filters and one diagonal matrix multiplication from the channel matrices.

A column filter is a combination of sub-carrier mapping matrix multiplication with a column masking frequency response matrix, denoted as $\ddot{M}'_C \mathbf{D}_i$ or $\ddot{M}''_C \mathbf{D}_i$. Sub-carrier mapping matrix affect the matrix's size reduction to $K \times M$, which means the column filter can be done in K time iterations as one iteration for one column. The structure of column filter matrix for user 1 is shown in Fig. 4(a).⁵ The flow of column filter matrix can be mapped into systolic architecture where the data flow is shown by the systolic space and time diagram. Fig. 4(b) shows the systolic architecture type applied which broadcast inputs, move results and weights stay.

A row filter is a result of row masking matrix multiplication with the sub-carrier de-mapping matrix, $\mathbf{D}_i^T \ddot{M}'_R$ or $\mathbf{D}_i^T \ddot{M}''_R$. These matrices have $M \times K$ size as shown Fig. 5(a). For total of K iterations, in each iteration a complete row computation should performed. Thus, we select the systolic type with fan-in results, move inputs, and weight stays, where the space and

⁵Note that user 1 in this figure is just for an example to show the systolic space time representation of column filter. The same assumption applies for row filter.

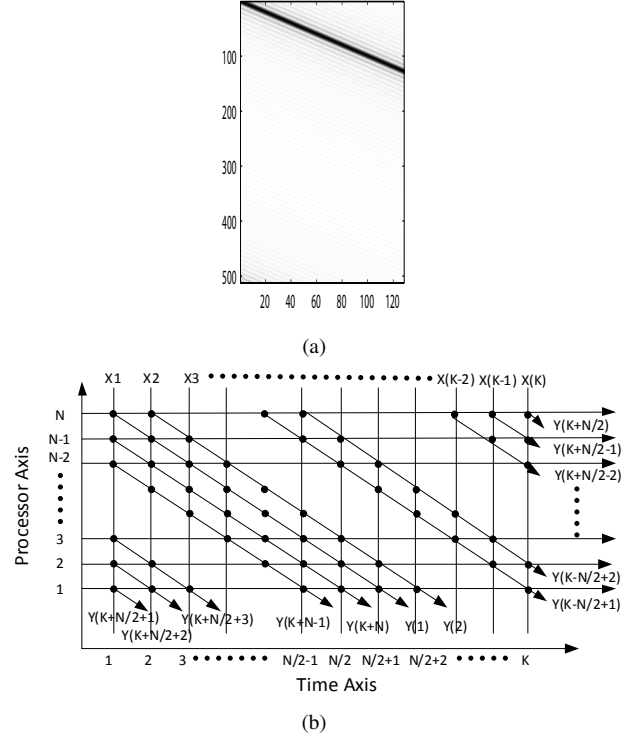


Fig. 4. (a) Matrix $\mathbf{D}_i^T \ddot{M}'_R$ or $\mathbf{D}_i^T \ddot{M}''_C$ for user 1, (b) The systolic space time representation for column filter

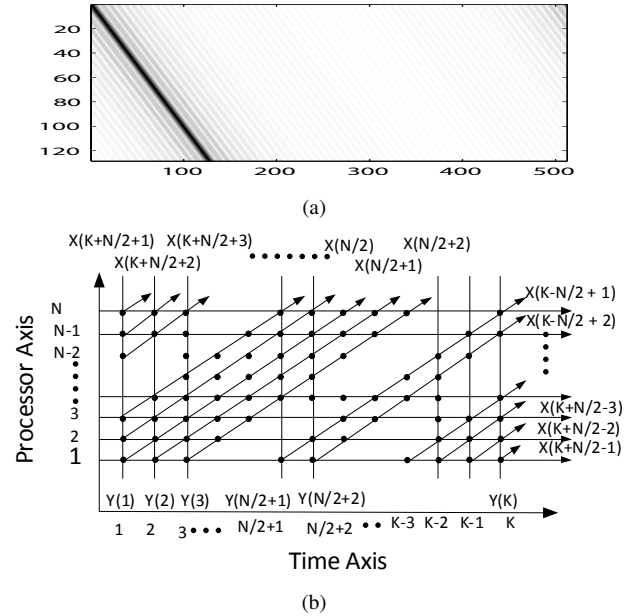


Fig. 5. (a) Matrix $\mathbf{D}_i^T \ddot{M}'_R$ or $\mathbf{D}_i^T \ddot{M}''_C$ for user 1, (b) The systolic space time representation for row filter

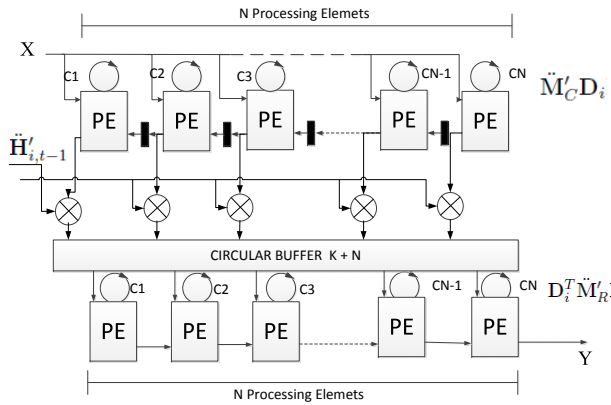


Fig. 6. Systolic architecture for row and column filter

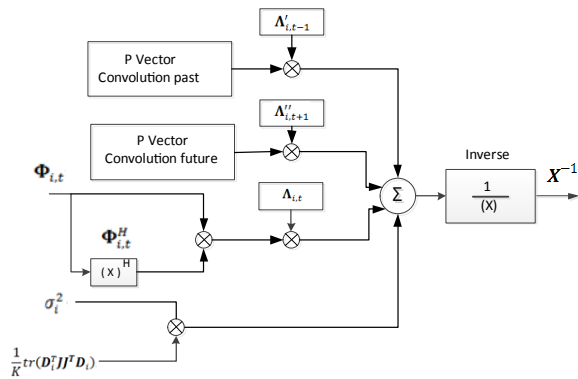


Fig. 7. Block Diagram to compute \mathbf{X}^{-1}

time diagram is shown in Fig. 5(b). While the parallel computation for input data is processed for each row computation, the input data should be stored in registers. Those registers is implemented as a circular buffer due to the merit of circulant masking matrix frequency response. The buffer has size of $K + N$, where N is the number of filter's coefficients.

Finally, the hardware architecture for the past/future blocks computation can be designed as in Fig. 6. One processing element (PE) is implemented as: one register for filter coefficient, one adder, and one multiplier. In details, Section V investigates the effect of reducing the PE.

B. Hardware Architecture for matrix \mathbf{X} computation

As shown in (11), matrix \mathbf{X} comprises the noise, frequency domain components of interferences from the past and the future blocks and the frequency domain components of the current block [4]. The computation of its inversion is shown in the block diagram of Fig. 7. It should be noted here that since the computation for noise term can be considered as multiplication of two scalars, we use only a single multiplier, while the variable $\text{tr}(\mathbf{D}_i^T \mathbf{J} \mathbf{J}^T \mathbf{D}_i)$ can be pre-computed as a constant. Similarly, the diagonal matrix-by-matrix multiplication for the current channel covariance computation can also be implemented using a single multiplier because the output data are in series.

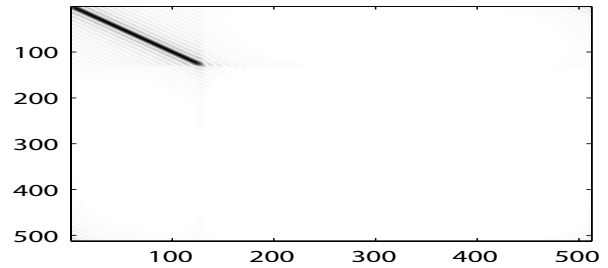


Fig. 8. The shape of matrix \mathbf{P} for user 1

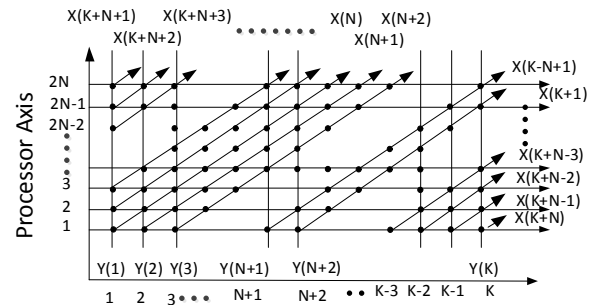


Fig. 9. Space and time representation for vector \mathbf{p} convolutions

The past and future covariance computations are implemented based on (16), where one circular convolution and one diagonal matrix multiplication are needed. The circular convolution can be performed in K iterations because matrix \mathbf{P} has K non-zero columns and K non-zero rows as plotted in Fig. 8.

We apply the same systolic architecture type as for row filter to this vector \mathbf{p} circular convolution, which is fan-in results, move inputs, and weights stay, as shown in Fig. 9. However, the number of processing element required for this circular convolution is $2N$, because matrix \mathbf{P} is resulted from the column filter with row filter, as N is the number of processing elements in those masking filters.

The diagonal matrix multiplication for the past/future covariance computation can be realized with a single multiplier, since the output data from circular convolution are in series. The trace operator is implemented with one accumulator and $\log(K)$ right shifter for $1/K$ operation. It results in hardware architecture for the past/future covariance matrix computation as shown in Fig. 10.

V. SIMULATION RESULTS

To investigate the effect of bit-width reduction, we conduct fixed point simulations to find the minimum required bit-width for hardware implementation. We assume the channel is 64-path block Rayleigh fading channels (the channel gains do not change within a block) with average equal power. The performance is presented in Fig. 11 in terms of average BER (over the fading channel realizations) vs. average energy bit per noise, E_b/N_0 (dB). From the figure, we found that the minimum bit-width is 9 bits since the BER degradation is

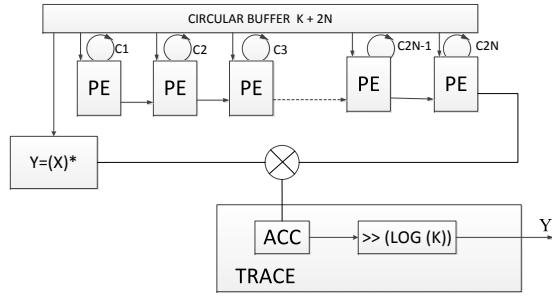


Fig. 10. Systolic architecture for matrix X computation

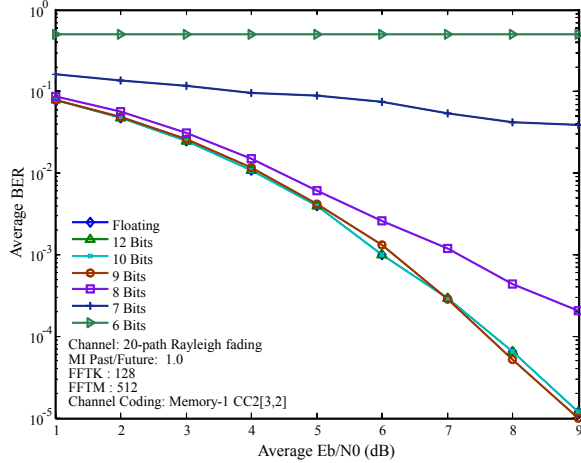


Fig. 11. Bit Error Rate Performance with difference fixed point setting

large when the bit-width is set less than 8 bits, which finally fails at bit-width of 6 bits.

In Fig. 12, we evaluate the effect of coefficient reductions since the row and column filters, including the vector \mathbf{p} contains elements of *sinc* function. The *sinc* has center energy in the middle, but very small at the beginning and at the end. Thus, the reduction of coefficients number may not effect significantly as long the center part of the *sinc* is kept. Our results of this investigation is shown in Fig. 12, where the BER degraded slightly with 32 or even with 16 coefficients.⁶ In this section, we may conclude that the row and column filters can be implemented with simply 16 processing elements, and 32 processing element for matrix \mathbf{W} with degradation in BER performance of only 1 dB (in maximum) compare with the computation using M numbers processing elements.

VI. CONCLUSIONS

In this paper, we have proposed an efficient hardware architecture so that the CHATUE-SC-FDMA algorithm can be implemented with very simple computation by applying the proposed masking matrices technique, which can be mapped

⁶In Fig. 12, BER is performed over 64-path block Rayleigh fading channels so that the maximum performance is better than BER with floating point in Fig. 11, which was performed over 20-path block Rayleigh fading channels. However, this difference does not cancel the conclusion about effect of number of coefficients.

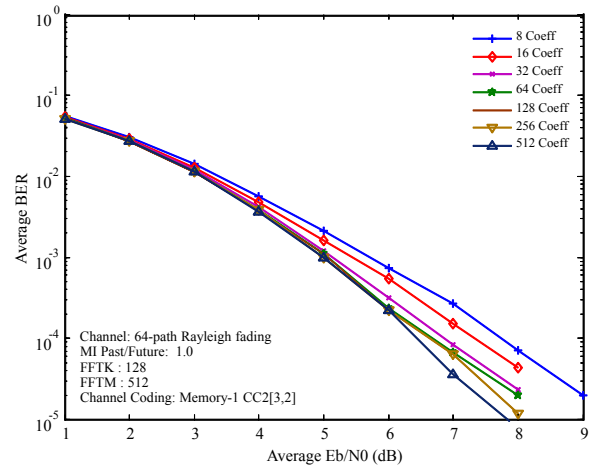


Fig. 12. BER performance for reducing masking filters coefficients

directly to the systolic architecture. Heavy computational complexity such as FFT for matrix-by-matrix multiplications can be avoided by row filter, column filter, and vector \mathbf{p} convolution. Furthermore, the proposing architecture has also been evaluated for fixed point model, which show that 9 bits quantization is enough to achieve as good performance as floating point model. Finally, we conclude that required computation for CHATUE-SC-FDMA technique can be reduced up to 96% (with acceptable degradation, less than 1 dB) which is very significant for spectral efficient SC-FDMA system without CP.

VII. ACKNOWLEDGEMENT

The authors are thankful to Prof. Tad Matsumoto for valuable supports and interesting discussions when conducting this research in JAIST.

REFERENCES

- [1] K. Anwar, H. Zhou, and T. Matsumoto, "Chained turbo equalization for block transmission without guard interval," in *IEEE VTC-Spring 2010*, Taiwan, May 2010.
- [2] K. Anwar and T. Matsumoto, "Low complexity time-concatenated turbo equalization for block transmission without guard interval: Part 1–The concept," *Wireless Pers. Commun., Springer*, 2012, DOI: 10.1007/s11277-012-0563-0 (Online 24 March 2012).
- [3] K. Anwar, H. Zhou, and T. Matsumoto, "Chained turbo equalization for SC-FDMA systems without cyclic prefix," in *IEEE Globecom 2010 Workshop on Broadband Single Carrier and Frequency Domain Communications*, Miami, Dec. 2010.
- [4] H. Zhou, K. Anwar, and T. Matsumoto, "Low complexity time-concatenated turbo equalization for block transmission without guard interval: Part 2–Application to SC-FDMA," *Wireless Pers. Commun., Springer*, Sept. 2011, DOI: 10.1007/s11277-011-0409-1.
- [5] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley-Interscience, 1999, page 192–200.
- [6] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Info. Theory*, vol. IT-20(2), pp. 284–287, March 1974.
- [7] A. Bottcher and S. M. Grudsky, *Spectral Properties of Banded Toeplitz Matrices*. Siam, 2005, page 31–33.