A Robust Channel Estimator for High-Mobility STBC-OFDM Systems

Hsiao-Yun Chen, *Associate Member, IEEE*, Meng-Lin Ku, Shyh-Jye Jou, *Senior Member, IEEE*, and Chia-Chi Huang

***Abstract*—In this paper, a robust channel estimator for high-mobility space-time block code-orthogonal frequency division multiplexing (STBC-OFDM) systems is proposed and applied in IEEE 802.16e systems. A high-performance two-stage channel estimation method is adopted. The proposed architecture reduces computational complexity effectively and improves 85.2% of the hardware implementation. The performances of the proposed design have been demonstrated through the simulation of an STBC-OFDM system with two transmit antennas and one receive antenna. At the vehicle speed of 120 and 240 km/hr for quadrature phase shift keying (QPSK) modulation, the proposed design can achieve the bit-error rate (BER) of about 10^{-4} and 10^{-3} without using channel coding. Moreover, it has significant performance improvement as compared with interpolation-based channel estimation methods. The proposed channel estimator implemented in 90 nm CMOS technology can support up to 29.03 Mbps (uncoded) downlink data transmission. The design only requires 859.6 K gates and dissipates 43.71 mW at 83.3 MHz operating frequency with 1 V power supply.**

***Index Terms*—Channel estimator, space-time block code, orthogonal frequency division multiplexing, IEEE 802.16e.**

I. Introduction

In recent years, space-time block code (STBC)-orthogonal frequency division multiplexing (OFDM) techniques (STBC-OFDM) have been shown to be very promising [1]–[3]. With multiple transmit antennas, STBC can provide transmit diversity gain to improve system performance in wireless communications, especially when receive diversity is too expensive to deploy. STBC-OFDM systems have been adopted in IEEE 802.16e which is an extension of IEEE 802.16-2004 for supporting the mobility of wireless metropolitan area network (WMAN) [4], [5]. However, for STBC decoding, STBC-OFDM systems require accurate channel state information (CSI), which is particularly difficult to obtain in mobile wireless channels. Therefore, high quality channel estimation with acceptable hardware complexity is a crucial challenge for realizing a successful STBC-OFDM system.

Manuscript received November 26, 2008; revised April 07, 2009. First published December 22, 2009; current version published April 09, 2010. This work was supported by the UMC, MediaTek Inc., CIC and the National Science Council of Taiwan, under Grant NSC96-2220-E-9-4. This paper was recommended by Associate Editor C.-Y. Chi

H. Y. Chen and S. J. Jou are with Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: jerryjou@mail.nctu.edu.tw).

M. L. Ku and C. C. Huang are with Department of Communications Engineering, National Chiao Tung University, Hsinchu 300, Taiwan.

Digital Object Identifier 10.1109/TCSI.2009.2027629

Various channel estimation methods have been proposed for OFDM systems. Among these methods, discrete Fourier transform (DFT)-based channel estimation methods using either minimum mean square error (MMSE) criterion or maximum likelihood (ML) criterion have been studied for OFDM systems with preamble symbols [6]–[8]. Since no information on channel statistics or operating signal-to-noise ratio (SNR) is required in the ML scheme, the ML scheme is simpler to implement than the MMSE scheme [6]–[8]. Furthermore, when the number of pilots is sufficient, the two schemes have comparable performances [8]. For this reason, the decision-feedback (DF) DFT-based channel estimation method is adopted to use the decided data as pilots to track channel variations for providing sufficient tracking information. Recently, Ku and Huang [9], [10] presented a DF DFT-based method derived from ML criterion and Newton's method. Moreover, they concluded that a refined two-stage channel estimation method [10] is more robust than the classical DF DFT-based method to apply in fast time-varying channels. Thus, the two-stage channel estimation method with an initialization stage and a tracking stage is adopted in this paper. Nevertheless, the two-stage channel estimation method has high computational complexity and is difficult to realize in hardware directly; hence, a novel architecture and an implementation method shall be proposed to reduce the hardware complexity.

In this paper, a robust channel estimator for high-mobility STBC-OFDM systems is proposed and implemented in IEEE 802.16e baseband receiver. The channel estimator designed in 90 nm CMOS technology can support up to 29.03 Mbps (uncoded) downlink data transmission. This design has about 859.6 K gates and dissipates 43.71 mW at 83.3 MHz operating frequency. As compared with interpolation-based channel estimation methods which are commonly adopted in the channel estimator designs [11], [12], our proposed channel estimator has significant performance improvements, especially when it is applied in fast and selective fading channels. The proposed channel estimator includes the following features:

- implementation of a robust channel estimator applied in an STBC-OFDM system with two transmit antennas and one receive antenna;
- adoption of a high-performance two-stage channel estimation method for providing precise CSI in high-mobility wireless channels;
- provision of an efficient channel estimator architecture for low-complexity hardware implementation while keeping the high performance.

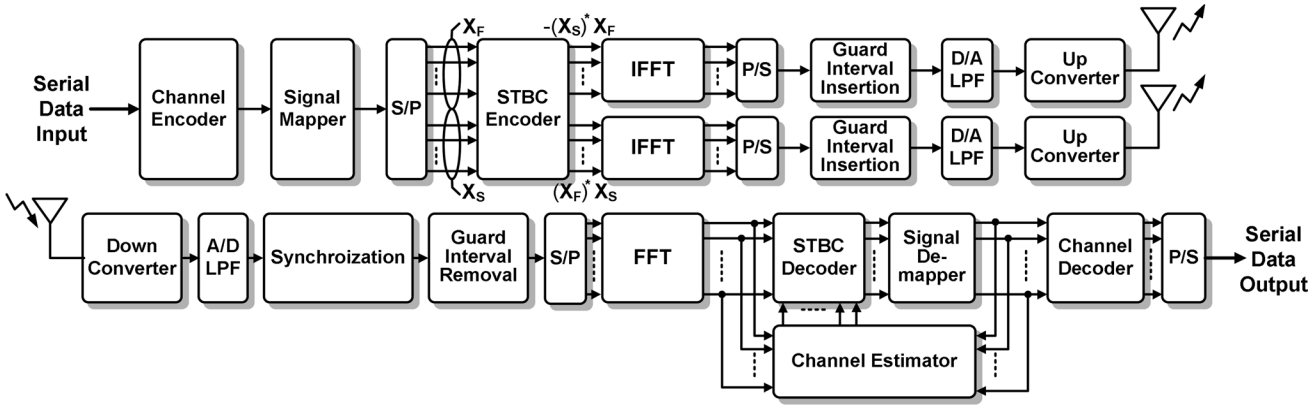

Fig. 1. Proposed STBC-OFDM system with two transmit antennas and one receive antenna.

TABLE I
MAJOR PARAMETERS OF THE PROPOSED STBC-OFDM SYSTEM

Parameters		Values
RF Frequency		2.5 GHz
System channel bandwidth (BW)		10 MHz
Sampling frequency (F_S)		11.9 MHz
FFT size (N)		1024
Subcarrier spacing (Δf)		11.6 kHz
Useful symbol time (T_b)		86.0 μs
Guard time (T_g)		10.8 μs
OFDM symbol duration (T_S)		96.8 μs
Number of OFDM data symbols in a frame		40
DL PUSC	Number of pilot subcarriers (N_{Pilot})	120
	Number of data subcarriers (N_{Data})	720

This paper is organized as follows. Section II describes the system architecture. Section III briefly reviews the two-stage channel estimation method. Section IV presents the proposed channel estimator. Then, the simulations and results are provided in Section V. Finally, Section VI is the conclusions.

Notation: By convention, boldface letters are used for sets, vectors, and matrices. The superscript $(\cdot)^*$ stands for complex conjugate. The notation $\mathrm{sign}(\cdot)$ takes the sign of $(\cdot)$. The notations $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ stand for the real part and the imaginary part of $(\cdot)$. The notation $\{\ldots\}$ denotes the contain elements of a set or a vector.

II. SYSTEM ARCHITECTURE

The orthogonal frequency division multiple access (OFDMA) specification of IEEE 802.16e that supports the multi-antenna technology is adopted in this paper. In downlink (DL) transmission, the subcarrier allocation of partial usage of subchannels (PUSC) is supported in this proposed system. The major parameters of the proposed STBC-OFDM system are summarized in Table I. The quadrature phase shift keying (QPSK) and 16 quadrature amplitude modulation (16QAM) are supported for data subcarriers, while binary phase shift keying (BPSK) is adopted for pilot subcarriers and preamble symbols. Each frame is composed of one preamble symbol and 40 OFDM data symbols. The cyclic prefix (CP) length is 128 sampling periods, i.e., 1/8 of the useful symbol time.

The proposed STBC-OFDM system with two transmit antennas and one receive antenna is shown in Fig. 1. In the transmitter, Alamouti's STBC encoding method [1] is used to encode two transmitted symbols, $\mathbf{X}_F = \{X_F[k] : 0 \le k \le N-1\}$ and $\mathbf{X}_S = \{X_S[k] : 0 \le k \le N-1\}$, within a time slot which is the duration of two OFDM symbols, where k is the subcarrier index, and N is the total number of subcarriers. The N-point inverse fast Fourier transform (IFFT) unit is used in each arm to transform the frequency domain OFDM symbols into time domain. The CP with time duration T_g is then inserted as a guard interval to combat inter-symbol interference (ISI). Finally, a complete OFDM symbol with symbol duration $T_b + T_g$ is converted into an analog signal by a digital-to-analog (D/A) converter, filtered by a low-pass filter (LPF), up converted to RF band, and transmitted in air.

The receiver architecture consists mainly of a channel estimator along with other blocks. After an RF signal has been received from an antenna, it is down converted to the equivalent baseband, low-pass filtered, and digitized by an analog-to-digital (A/D) converter. Both timing and carrier frequency synchronization are assumed to be ideal in this case. The channel is assumed to be quasi-static within any two successive OFDM symbol durations. Hence, without loss of generality, the signal processing of the received data is focused on each time slot, and the symbol time index is omitted hereafter except otherwise stated. The channel frequency response between the first transmit antenna and the receive antenna is denoted as $H^{(1)}[k]$, and the other one is denoted as $H^{(2)}[k]$. Within a time slot, after the received signals have passed through the guard interval removal and the N-point fast Fourier transform (FFT), the two successive received OFDM symbols, $R[1,k]$ and $R[2,k]$, are given by

$$R[1,k] = H^{(1)}[k]X_F[k] + H^{(2)}[k]X_S[k] + Z[1,k] \quad (1)$$

$$R[2,k] = -H^{(1)}[k](X_S[k])^* + H^{(2)}[k](X_F[k])^* + Z[2,k] \quad (2)$$

for $k \in \mathbf{Q} \cup \mathbf{J}$, where $\mathbf{Q}$ and $\mathbf{J}$ denote the sets of data and pilot subcarrier indices, respectively, and $Z[1,k]$ and $Z[2,k]$ are the uncorrelated additive white Gaussian noise (AWGN) with zero-mean and variance $(\sigma_Z)^2$.

III. Two-Stage Channel Estimation Method

Most mobile wireless channels are characterized by channel impulse response (CIR) consisting of a few dominant paths. These path delays usually change slowly in time, but the path gains may vary relatively fast. In this section, the refined two-stage channel estimation method [10] will be briefly reviewed. An initialization stage uses a multipath interference cancellation (MPIC)-based decorrelation method to identify the significant paths of CIR in the beginning of each frame. However, the CIR estimated by the preamble can not be directly applied in the following data bursts since the receiver is mobile. Thus, a tracking stage is then used to track the path gains with known CIR positions. The details are described as follows.

A. Initialization Stage

The MPIC-based decorrelation estimates CIR path-by-path and cancels out the known multipath interference. The channel estimation for each transceiver antenna pair can be independently performed because the preambles transmitted from different antennas do not interfere with each other. First, two parameters N_P and $\mathbf{W}_B$ are defined as a presumptive path number of a channel and an observation window set, respectively. Second, the cyclic cross-correlation $C_{\mathrm{RP}}[\tau]$ between the received and transmitted preambles as well as the normalized cyclic auto-correlation $C_{\mathrm{PP}}[\tau]$ of the transmitted preamble are calculated. The indexes l and κ which stand for a path counting variable and the number of the legal paths found by the MPIC-based decorrelation are initialized to zero. Third, the process is started by picking only one path whose time delay τ_l yields the largest value in $C_{\mathrm{RP}}[\tau_l]$, for $\tau_l \in \mathbf{W}_B$. If the path delay τ_l is larger than the length of CP, this path is treated as an illegal path and discarded by setting $C_{\mathrm{RP}}[\tau_l] = 0$. Otherwise, this path is recorded as the κ-th legal path with a time delay $\tau_\kappa = \tau_l$ and a complex path gain $\mu_\kappa = C_{\mathrm{RP}}[\tau_l]$. Then, the interference associated with this legal path is canceled from $C_{\mathrm{RP}}[\tau]$ to obtain a refined cross-correlation function

$$C_{\mathrm{RP}}[\tau] = C_{\mathrm{RP}}[\tau] - \mu_\kappa C_{\mathrm{PP}}[|\tau - \tau_\kappa|], \quad \tau \in \mathbf{W}_B \backslash \{\tau_0, \ldots, \tau_{l-1}\}. \tag{3}$$

Meanwhile, κ is increased by one. The value of l is also increased by one at the end of each iteration, and the iterative process is continued until l reaches the presumed value of N_P.

B. Tracking Stage

After the initialization stage, we can obtain the information of the path numbers $\kappa^{(i)}$, the multipath delays $\tau_l^{(i)}$, the multipath complex gains $\mu_l^{(i)}$, for $l \in \{0, \ldots, \kappa^{(i)} - 1\}$, and the corresponding channel frequency responses, where i is corresponding to the i-th transmit antenna. Under the assumption that the multipath delays do not change over the duration of a frame, the DF DFT-based channel estimation method can be equivalently expressed in Newton's method as [9]

$$\boldsymbol{\delta}_\nu[k] = \mathbf{M}_{\nu-1}[k] - \frac{1}{\hat{C}_\nu[k]}(\hat{\mathbf{X}}_\nu[k])^* \mathbf{R}[k] \tag{4}$$

$$\mathbf{q}_\nu^{(i)} = \mathbf{F}_{\mathrm{IDFT}}^{(i)} \boldsymbol{\Delta}_\nu^{(i)} \tag{5}$$

$$\mathbf{M}_\nu^{(i)} = \mathbf{M}_{\nu-1}^{(i)} - \mathbf{F}_{\mathrm{DFT}}^{(i)} [\mathbf{E}^{(i)}]^{-1} \mathbf{q}_\nu^{(i)}. \tag{6}$$

According to [9], the vector $\boldsymbol{\delta}_\nu[k] = \{\Delta_\nu^{(i)}[k] : i = 1, 2\}$ calculates the difference between the previous estimated channel frequency response vector $\mathbf{M}_{\nu-1}[k] = \{M_{\nu-1}^{(i)}[k] : i = 1, 2\}$ and the least-square (LS) estimation vector in (4), where ν is the iteration index. The matrix $\hat{\mathbf{X}}_\nu[k]$ is the re-encoded STBC matrix with decided symbols, $\hat{X}_F[k]$ and $\hat{X}_S[k]$, as its entries. The decided symbols are obtained by applying the previous estimated channel frequency responses to decode the received signal vector $\mathbf{R}[k] = \{R[t, k] : t = 1, 2\}$, where t is the symbol index within a time slot. The value $\hat{C}_\nu[k] = |\hat{X}_F[k]|^2 + |\hat{X}_S[k]|^2$ is the energy normalization factor. The inverse DFT (IDFT) matrix $\mathbf{F}_{\mathrm{IDFT}}^{(i)}$ multiplying by the vector $\boldsymbol{\Delta}_\nu^{(i)} = \{\Delta_\nu^{(i)}[k] : k \in \boldsymbol{\Theta}\}$ in (5) is to form the gradient vector $\mathbf{q}_\nu^{(i)}$ in Newton's method as shown in (6), where $\boldsymbol{\Theta}$ is a subset of $\mathbf{Q}$. In addition, the weighting matrix $[\mathbf{E}^{(i)}]^{-1}$ in (6) is in fact the inverse of the Hessian matrix in Newton's method [9]. The (l, l')-th entry of $\mathbf{E}^{(i)}$ is given by

$$(\mathbf{E}^{(i)})_{l,l'} = \sum_{k \in \boldsymbol{\Theta}} \cos\left(\frac{2\pi k\left(\tau_l^{(i)} - \tau_{l'}^{(i)}\right)}{N}\right) + j \sum_{k \in \boldsymbol{\Theta}} \sin\left(\frac{2\pi k\left(\tau_l^{(i)} - \tau_{l'}^{(i)}\right)}{N}\right). \tag{7}$$

In the previous studies [6], [7], the pilots as well as the decided data symbols are simultaneously adopted to perform channel estimation at each tracking iteration. From the viewpoint of optimization, since the pilots inserted in each OFDM symbol are much more reliable than the decided data symbols, they should play a dominant role in providing a global search direction at the first tracking iteration [10]. Thus, the first iteration of the channel tracking is modified as

$$\mathbf{M}_1^{(i)} = \mathbf{M}_0^{(i)} - \gamma \cdot \mathbf{F}_{\mathrm{DFT}}^{(i)} \mathbf{q}_1^{(i)} \tag{8}$$

where the gradient vector $\mathbf{q}_1^{(i)}$ is calculated according to (4)–(5) by using the pilot subcarrier set $\mathbf{J}$ instead of the set $\boldsymbol{\Theta}$, and the value γ is an experimental constant of step size to have the best performance.

It is demonstrated in [10] that the two-stage channel estimation method has better performance than the classical DF DFT-based method, the STBC-based MMSE method, and the Kalman filtering method for estimating channels in high mobility, and its computational complexity is quite the same with these methods. However, the high complexity problem still needs to be solved for hardware implementation. Hence, we propose a modified two-stage channel estimation method and its architecture for hardware design.

IV. Proposed Channel Estimator

The overall block diagram of the proposed channel estimator is shown in Fig. 2. The initialization stage is decomposed to a preamble match, an IFFT, a straight MPIC (SMPIC)-based decorrelator, and an FFT. The tracking stage is decomposed to an STBC decoder, a demapper, an LS estimator, an IFFT, a path

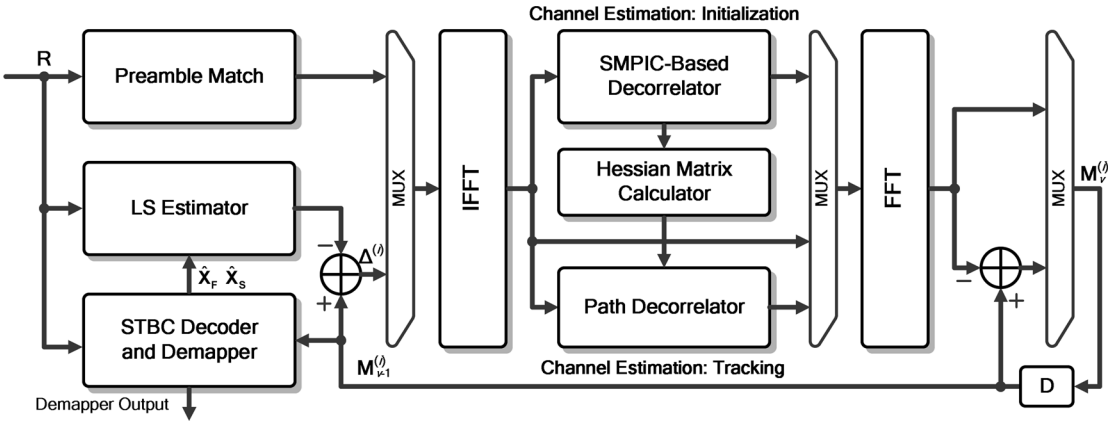

Fig. 2. Overall block diagram of the proposed channel estimator.

decorrelator, a Hessian matrix calculator and an FFT. Moreover, the IFFT and FFT are shared between the initialization stage and the tracking stage. These key blocks are described in the following subsections.

A. Initialization Stage: Preamble Match

In the initialization stage, the preamble match is used to estimate the preliminary channel frequency responses $M^{(i)}[k]$ for $i = 1, 2$ by matching the received signal with the preamble $P^{(i)}[k]$ transmitted from the i-th antenna. Since the preambles transmitted from different antennas do not interfere with each other, $M^{(i)}[k]$ can be independently performed by

$$M^{(i)}[k] = \frac{R[k] \cdot (P^{(i)}[k])^*}{|P^{(i)}[k]|^2} = H^{(i)}[k] + \frac{Z[k] \cdot (P^{(i)}[k])^*}{|P^{(i)}[k]|^2} \quad (9)$$

where $R[k]$ in the initialization stage is the first received OFDM symbol of a frame, and $Z[k]$ is AWGN. The values of preamble subcarriers are known patterns; besides, the subcarriers are modulated by BPSK and boosted as a constant power to increase the reliability. Thus, the absolute values of preamble subcarriers normalized by the power can be pre-computed to a real constant α. Furthermore, the value α can be quantized to a canonic sign digit (CSD) code [13] with two nonzero digits $(2^{\mathrm{a}} - 2^{\mathrm{b}})$ for the purpose of using only shifters and adders instead of multiplier implementation to reduce the design complexity. The $M^{(i)}[k]$ can be reformulated as

$$M^{(i)}[k] = \mathrm{sign}(P^{(i)}[k]) \cdot R[k] \cdot |P^{(i)}[k]|/|P^{(i)}[k]|^2 = \mathrm{sign}(P^{(i)}[k]) \cdot R[k] \cdot \alpha. \quad (10)$$

The preamble match design consists of subtractors and multiplexers controlled by the sign of $P^{(i)}[k]$ for forming the operations of (10) as shown in Fig. 3.

B. Initialization Stage: SMPIC-Based Decorrelator

After IFFT operation, the CSIs are obtained in time domain $m^{(i)}[\tau]$. The MPIC-based decorrelation method estimates CIR path-by-path. It picks the maximum path of $m^{(i)}[\tau]$ for $\tau \in \mathbf{W}_B$ and cancels the maximum path interference to other paths. If the set $\mathbf{W}_B$ has N_B paths, the process must iterate N_P times for finding the maximum path of these N_B paths and canceling the maximum path interference to other (N_B-1) paths. This method requires too many execution cycles and is unsuitable to directly implement in the proposed channel estimator.

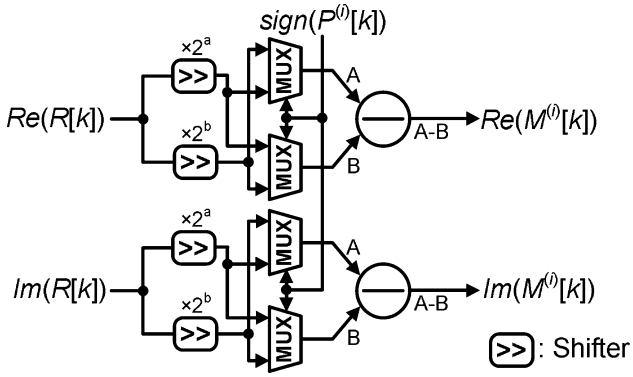

Fig. 3. Basic design unit of the preamble match.

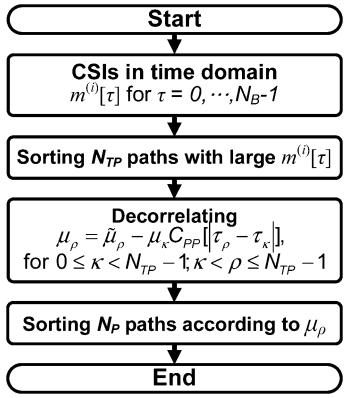

Fig. 4. Flowchart of the proposed SMPIC-based decorrelation.

In order to reduce the execution cycles, we propose an SMPIC-based decorrelation method to identify N_P significant paths in a straightforward method, and the flowchart is shown in Fig. 4. First, the proposed scheme sorts the N_B paths to find the first N_{TP} paths with large $m^{(i)}[\tau]$. Second, the decorrelation is carried out from the largest to the smallest one of these N_{TP} paths to cancel the path interference. Finally, the decorrelated N_{TP} paths are sorted again to pick up the first N_P

paths. For using a sorting network of fixed I/O size N_P to sort an arbitrarily larger data set, the number of N_{TP} is defined to be $\beta \cdot N_P$, and β is an integer which is searched to optimize the computational complexity and guarantee the acceptable performance. Here, the output SNR at the STBC decoder is used as a gauge of the system performance to determine the value of β and defined as

$$\text{output SNR} = \sum_{\ell=0}^{N_S-1} \mathrm{SNR}_\ell \cdot n_\ell / \sum_{\ell=0}^{N_S-1} n_\ell \tag{11}$$

$$\mathrm{SNR}_\ell = 10 \log_{10} \left(\frac{\phi_\ell^2}{2\sigma_\ell^2} \right) \tag{12}$$

$$\phi_\ell = \frac{1}{n_\ell} \sum_{j=0}^{n_\ell-1} |\hat{X}_j| \tag{13}$$

$$\sigma_\ell^2 = \frac{1}{n_\ell - 1} \sum_{j=0}^{n_\ell-1} (|\hat{X}_j|^2 - |\phi_\ell|^2) \tag{14}$$

where N_S is the number of symbols in a constellation, n_ℓ is the number of data belonging to the ℓ-th symbol after being sliced, and $\hat{X}_j$ is the desired data after STBC decoding.

If a sorting algorithm such as merge sorting is used, the computational complexity of the original MPIC-based decorrelation method requires $\mathrm{O}(N_P N_B \log_2 N_B)$ comparisons, $\mathrm{O}(N_P N_B)$ complex multiplications and $\mathrm{O}(N_P N_B)$ complex subtractions because it must repeat N_P times of sorting and decorrelation of N_B paths. However, the complexity of the SMPIC-based decorrelation method only requires $\mathrm{O}(N_B \log_2 N_B)$ comparisons, $\mathrm{O}(N_{\mathrm{TP}}^2)$ complex multiplications and $\mathrm{O}(N_{\mathrm{TP}}^2)$ complex subtractions. Thus, the requirement of execution cycles can be effectively reduced by about $\mathrm{O}(N_P)$ times.

For this channel estimator, N_B is defined to be 128 which is the CP length, and N_P is presumed to be eight. Fig. 5 shows the curves of the output SNR in QPSK modulation versus the value of β. These curves are simulated at the vehicle speed of 120 km/hr with different $\mathrm{E_b/N_0}$ which is defined as a ratio of received bit energy to the power spectral density of noise. The value of β is decided to be four where the curves of the output SNR get into saturation. Hence, the value of N_{TP} is 32. As compared with the original method, the performance loss due to the quantization of β is smaller than 0.5 dB when the bit error rate (BER) is at 10^{-4}.

The architecture of the SMPIC-based decorrelator requires a very efficient partial sorting network and a decorrelator. We propose a merge sorting network with programmable and partial sorting capability (MSNP^2) and a triangular decorrelator (TD).

1) MSNP^2*:* In order to avoid the high complexity of parallel sorting network, a fixed I/O size sorting network and a set of memory module are used to accommodate the number of sorting elements [14]–[16]. Here, the architecture of the MSNP^2 with a memory bank, a sorting control unit and an 8-item sorter is shown in Fig. 6. The 8-item sorter is the Batcher's sorting network with I/O size of eight. The Batcher's sorting network is widely used because of its inherent parallelism and short latency [17]. Fig. 7 shows the 8-item sorter, and the basic unit is a 2×2 comparator which is used to perform data comparison and exchange. The memory bank which is primarily used to save the path power values is organized into eight independent memory modules denoted as $\mathrm{MD_1}$–$\mathrm{MD_8}$.

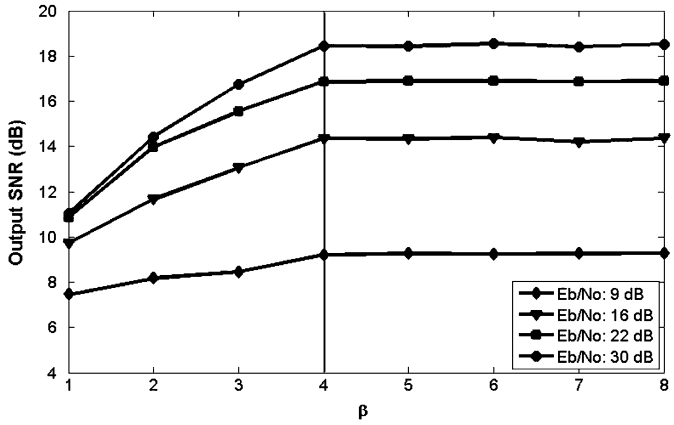

Fig. 5. Output SNR versus the value of β.

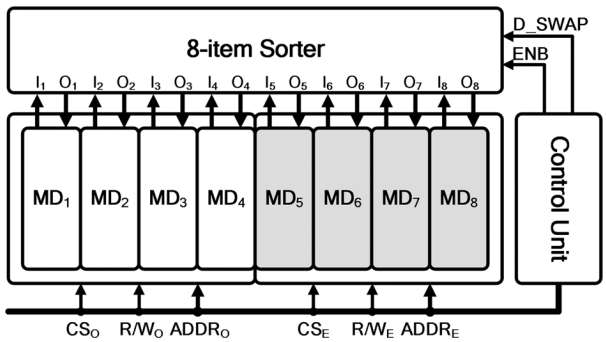

Fig. 6. Block diagram of the MSNP^2.

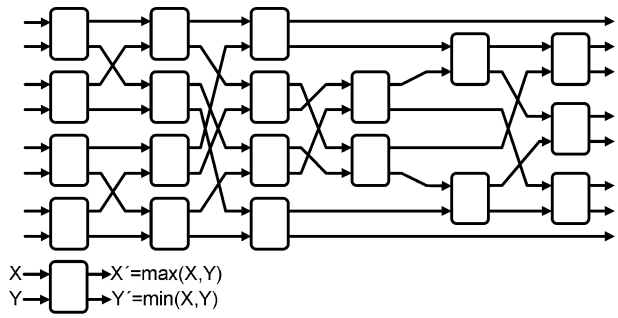

Fig. 7. Batcher classic sorting network with I/O size of eight.

Since the maximum sorting item is 128, the sorting data (path power values) are arranged with 32 rows which the row definition is used in the sorting sequence, and each row contains four sorting data. The odd rows are loaded into $\mathrm{MD_1}$–$\mathrm{MD_4}$, and the even rows are loaded into $\mathrm{MD_5}$–$\mathrm{MD_8}$. Based on the sorting sequence, the sorting control unit takes two rows of data to the 8-item sorter for sorting in each cycle; then, the outputs of the sorter which are divided into two clusters in descending order are written back to the memory bank and replaced the original two rows. The L-item merge-sorting sequence can be divided into three cycles: 1) the first local sorting cycle; 2) the cross sorting cycle; and 3) the second local sorting cycle. At the first local sorting cycle, L-item data are divided into two $L/2$-item data clusters to do $L/2$-item merge-sorting, respectively. Then,

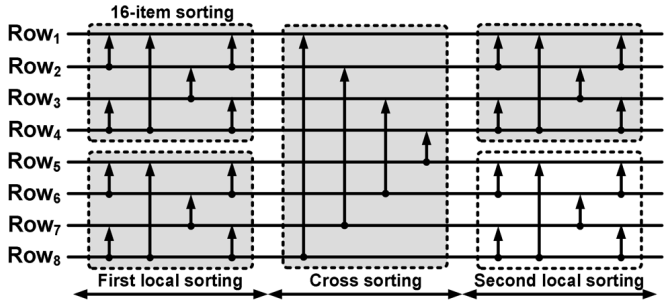

Fig. 8. 32-item merge sorting sequence.

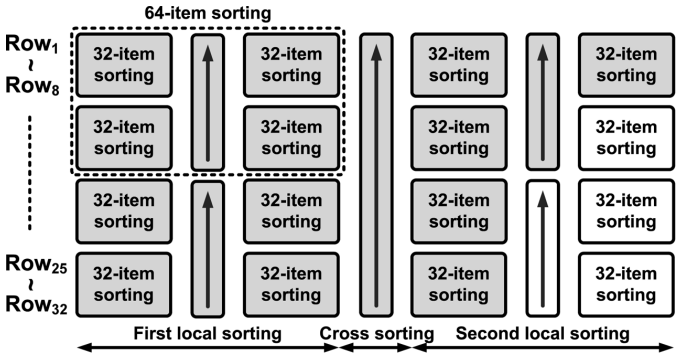

Fig. 9. 128-item merge sorting sequence.

L-item data will be arranged in two $L/2$-item clusters in descending order. At the cross sorting cycle, the data in the up cluster are compared and exchanged with the data in the down cluster. After cross sorting, the data in the up cluster are larger than that in the down cluster. At the second local sorting, the two clusters are sorted separately again in descending order. Finally, the sorted results are saved in the memory bank and arranged in the row order. Fig. 8 shows the 32-item merge sorting sequence represented by the directed arrows in the line representation, and each arrow represents an operation of the 8-item sorter. The merge sorting is used two times in the SMPIC-based decorrelator. The first time is to sort the 128-item data to find the first 32-item data and denoted as 128-32-item sorting. The second time is to sort the 32-item data to find the first 8-item data and denoted as 32-8-item sorting. The 32-item sorting sequence is used to be a basic control sequence, and the 128-item sorting sequence can be extended by the 32-item sorting sequence and constructed as the line representation shown in Fig. 9. For saving execution time and power, the 128-32-item sorting only executes the grey part of Fig. 9, and the 32-8-item sorting executes the grey part of Fig. 8.

2) TD: The TD consists of a decorrelated control unit, a decorrelated unit, and a memory bank shared with the sorting process. The TD is executed after the first 128-32-item sorting. There are 31 iterations of the TD process, and the process starts at the first legal path which is the maximum sorted path. If κ denotes the iteration number, for $0 \leq \kappa < 31$, the process of the κ-th iteration is to cancel the interferences associated with the κ-th legal path gain. The process can be expressed as

$$\mu_\rho = \tilde{\mu}_\rho - \mu_\kappa C_{\mathrm{PP}}[\lfloor \tau_\rho - \tau_\kappa \rceil], \kappa < \rho \leq 31 \tag{15}$$

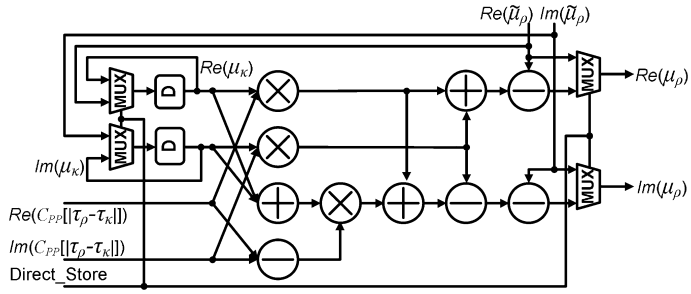

Fig. 10. Design of the decorrelated unit.

where μ_κ is the κ-th legal path gain, $\tilde{\mu}_\rho$ is the ρ-th sorted path gain, μ_ρ is the ρ-th decorrelated sorted path gain, τ_κ is the κ-th legal path delay, τ_ρ is the ρ-th sorted path delay, and $C_{\mathrm{PP}}[\tau]$ is the normalized cyclic auto-correlation of the preamble. The process of the κ-th iteration executes $(31-\kappa)$ times of decorrelating calculation. At the κ-th iteration, the process does not decorrelate the interference to the first $(\kappa-1)$ legal paths since the interference value is much smaller than their path gains and does not influence the accuracy of the decision in the significant path positions. Moreover, the small path gain offset can be revised in the tracking process without loss of the performance. In this way, the TD process can effectively save about half of execution cycles and power consumption. After the κ-th iteration processing, the $(\kappa+1)$-th legal path is acquired to execute the next iteration. Because the preamble is a known pattern, the value of $C_{\mathrm{PP}}[\tau]$ with different τ can be calculated and stored in ROM in advance. After 31 iterations have been completed, the 32 legal paths are obtained and then sorted again to find eight significant paths. Therefore, following the process of the 32-item decorrelation, the decorrelated control unit is designed to control the access flow of the sorted paths and the decorrelated paths. Fig. 10 shows the design of the decorrelated unit. Within the execution of the κ-th iteration, μ_κ must be used (31-κ) times; hence, μ_κ is read from memory at the beginning and saved in the local registers to reduce memory access until the iteration is finished.

After the SMPIC-based decorrelator, the significant paths have been identified and are then transformed to channel frequency responses by FFT for using as the reference in the tracking stage.

C. Tracking Stage: STBC Decoder and Demapper

In the tracking stage, from (4), the LS estimator is used to calculate the LS estimations followed by calculating the vector $\boldsymbol{\delta}_\nu[k] = \{\Delta_\nu^{(i)}[k] : i = 1, 2\}$ that can be expressed as

$$\Delta_\nu^{(1)}[k] = M_{\nu-1}^{(1)}[k] - \frac{1}{\hat{C}_\nu[k]}((\hat{X}_F[k])^* R[1,k] - \hat{X}_S[k]R[2,k]) \tag{16}$$

$$\Delta_\nu^{(2)}[k] = M_{\nu-1}^{(2)}[k] - \frac{1}{\hat{C}_\nu[k]}((\hat{X}_S[k])^* R[1,k] + \hat{X}_F[k]R[2,k]). \tag{17}$$

Before the LS estimation calculation, the decided symbols $\hat{X}_{\mathrm{F}}[k]$ and $\hat{X}_S[k]$ must be determined first. Based on the latest estimated channel frequency responses, the STBC decoder and

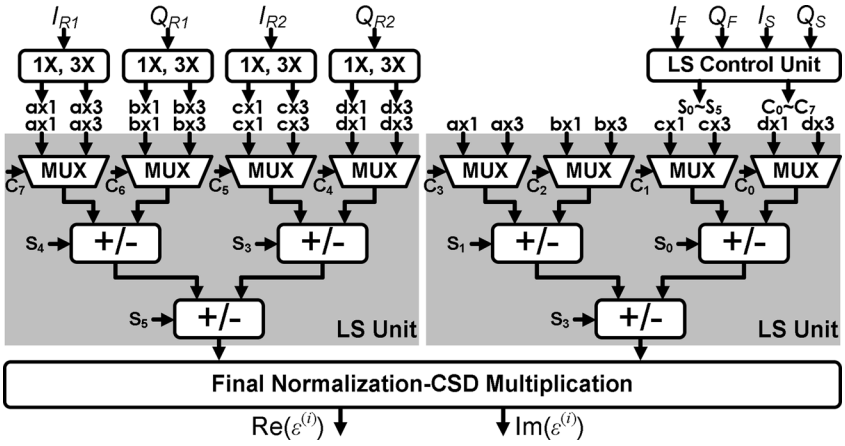

Fig. 11. Design of the LS estimator.

the symbol demapper are used to decode these two received symbols and can be formulated as

$$\hat{X}_F[k] = \mathrm{De}\left\{\frac{(M_{\nu-1}^{(1)}[k])^* R[1,k] + M_{\nu-1}^{(2)}[k](R[2,k])^*}{(|M_{\nu-1}^{(1)}[k]|^2 + |M_{\nu-1}^{(2)}[k]|^2)}\right\} \tag{18}$$

$$\hat{X}_S[k] = \mathrm{De}\left\{\frac{(M_{\nu-1}^{(2)}[k])^* R[1,k] - M_{\nu-1}^{(1)}[k](R[2,k])^*}{(|M_{\nu-1}^{(1)}[k]|^2 + |M_{\nu-1}^{(2)}[k]|^2)}\right\} \tag{19}$$

where $\mathrm{De}\{.\}$ is the demapper process. The hardware design of a divider is very costly; therefore, a demapping dichotomy method with two stages [18] is adopted to avoid the divider implementation. Also, a complex multiplier can be reduced from four multipliers and two adders to three multipliers and five adders. Hence, the design of the STBC decoder only requires 12 multipliers and 24 adders.

D. Tracking Stage: LS Estimator

After the decided symbols have been determined, the LS estimations, $\varepsilon_\nu^{(i)}[k]$ for $i = 1, 2$, are calculated by the LS estimator. Both $\{I_F, Q_F\}$ and $\{I_S, Q_S\}$ denote the I and Q coordinate values of $\hat{X}_\mathrm{F}[k]$ and $\hat{X}_S[k]$, respectively. Both $\{I_{R1}, Q_{R1}\}$ and $\{I_{R2}, Q_{R2}\}$ denote the real part and the imaginary part of $R[1,k]$ and $R[2,k]$, respectively. The value of constellation normalization and the value of $1/\hat{C}_\nu[k]$ both have a limited constant set; thus, these multiplications can be merged to one multiplication of λ, and the value of λ has also a limited constant set. The LS estimations can be expressed as

$$\begin{aligned}\varepsilon_\nu^{(1)}[k] &= ((\hat{X}_F[k])^* R[1,k] - \hat{X}_S[k]R[2,k])/\hat{C}_\nu[k] \\ &= \lambda \cdot [(I_{R1}\cdot I_F + Q_{R1}\cdot Q_F - I_{R2}\cdot I_S + Q_{R2}\cdot Q_S) \\ &\quad + j(Q_{R1}\cdot I_F - I_{R1}\cdot Q_F - Q_{R2}\cdot I_S - I_{R2}\cdot Q_S)]\end{aligned} \tag{20}$$

$$\begin{aligned}&\varepsilon_\nu^{(2)}[k] \\ &= ((\hat{X}_S[k])^* R[1,k] + \hat{X}_F[k]R[2,k])/\hat{C}_\nu[k] \\ &= \lambda \cdot [(I_{R1}\cdot I_S + Q_{R1}\cdot Q_S + I_{R2}\cdot I_F - Q_{R2}\cdot Q_F) \\ &\quad + j(Q_{R1}\cdot I_S - I_{R1}\cdot Q_S + Q_{R2}\cdot I_F + I_{R2}\cdot Q_F)].\end{aligned} \tag{21}$$

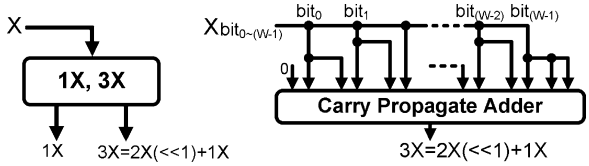

Fig. 12. Design of the coordinate precalculator.

Fig. 11 shows the design of the LS estimator which is composed of coordinate precalculators, LS units, an LS control unit and a final normalization. The coordinate precalculators are designed to generate the partial products of $R[t,k]$ multiplied by the coordinate values. The coordinate precalculators support the modulations of BPSK, QPSK, and 16QAM, and the multiples should be $\{1\mathrm{x}, 3\mathrm{x}\}$. Fig. 12 shows a coordinate precalculator implemented by carry propagate adder (CPA). An LS unit including multiplexers and adders is used to generate the LS estimation results without normalization. The LS control unit is based on the values of I_F, Q_F, I_S, and Q_S to generate the control signals for selecting the outputs of the coordinate precalculators, controlling the adders to add or subtract and choosing the results multiplied by the corresponding value of λ. Since the value of λ has a limited constant set, all possible values can be applied by CSD coding and then searched their common subexpressions to implement CSD multiplications to avoid the usage of dividers. Finally, the result is outputted after the final normalization.

Fig. 13 shows the pilots which are transmitted in the cluster structures over different time slot. In IEEE 802.16e, each cluster contains 14 subcarriers, and there are 60 clusters in an OFDM symbol with 1024 subcarriers. Each cluster has two pilot subcarriers, and the pilots are modulated by BPSK. If a pilot is transmitted on one pilot subcarrier from one antenna, the other antenna will not transmit a pilot on the same subcarrier to avoid the inter-antenna interference. The dimension of $\mathbf{J}$ is N_J. According to this allocation, if the pilot subcarrier index is $k \in \mathbf{J} = \{J_0, J_1, \ldots, J_{N_\mathrm{J}-1}\}$, the LS estimations at the first iteration can be expressed as follows:

$$\varepsilon_1^{(1)}[k] = \begin{cases} \pm\sigma \cdot R[2,k], \text{when } k = J_{2k'} \\ \pm\sigma \cdot R[1,k], \text{when } k = J_{2k'+1} \end{cases} \tag{22}$$

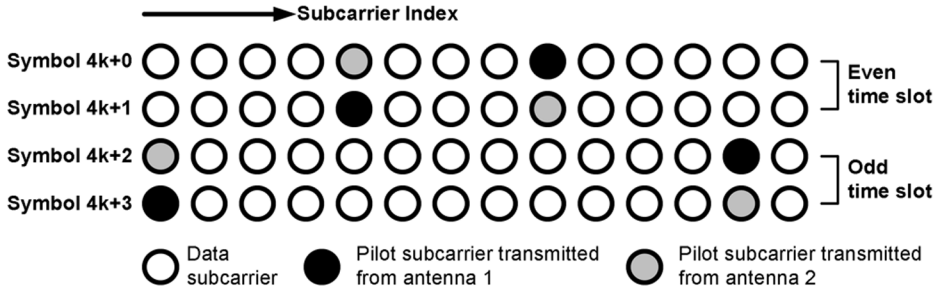

Fig. 13. Pilots transmitted in the cluster structures over different time slots.

$$\varepsilon_1^{(2)}[k] = \begin{cases} \pm\sigma \cdot R[1,k], \text{when } k = J_{2k'} \\ \pm\sigma \cdot R[2,k], \text{when } k = J_{2k'+1} \end{cases} \tag{23}$$

where the index k' is in the range $0 \le k' < N_{\mathrm{J}}/2$, and σ is a constant value to represent the absolute pilot value normalized by the pilot power. Therefore, the LS estimations at the first iteration can be implemented by a constant CSD multiplication.

After the LS estimation calculation, $\boldsymbol{\delta}_\nu[k]$ is acquired by subtracting the LS estimations from the latest estimated channel frequency responses.

E. Tracking Stage: Hessian Matrix Calculator and Path Decorrelator

We then pass $\boldsymbol{\Delta}_\nu^{(i)}$ through IFFT to obtain $\mathbf{q}_\nu^{(i)}$ in time domain. Only those gradient entries of $\mathbf{q}_\nu^{(i)}$ that have the same path delays as the significant paths identified in the initialization stage are considered; therefore, white noise will be filtered out, and the decision error propagation effect can also be alleviated. Since data and pilot subcarriers are not equally-spaced, the aliasing between the paths occurs. The path decorrelator works to decorrelate the inter-path interferences. Before the path decorrelation, the inverse of the Hessian matrix, $[\mathbf{E}^{(i)}]^{-1}$, should be calculated first. Although $[\mathbf{E}^{(i)}]^{-1}$ is only calculated once within a frame operation, the matrix inverse computation needs very high complexity of $\mathrm{O}(N_P^3)$ complex multiplications. Besides, each entry of $\mathbf{E}^{(i)}$ should take at least N_Θ cycles to calculate the cosine and sine summations by using a look-up table, where N_Θ is the dimension of $\boldsymbol{\Theta}$. If $[\mathbf{E}^{(i)}]^{-1}$ is implemented directly, it will require very large hardware module and memory. In order to reduce the requirement, the matrix inverse is avoided by considering the strongly diagonal property [10]. If $\kappa^{(i)}$ is the significant path number, $\mathbf{E}^{(i)}$ is decomposed to $N_\Theta(\mathbf{I}_{\kappa^{(i)}} + \mathbf{O}_{\text{off}})$, where $\mathbf{I}_{\kappa^{(i)}}$ is a $\kappa^{(i)} \times \kappa^{(i)}$ identity matrix, and $\mathbf{O}_{\text{off}}$ is a zero-diagonal matrix. If N_Θ is large enough, an approximate weighting matrix of $[\mathbf{E}^{(i)}]^{-1}$ takes the form as

$$\begin{aligned}[\mathbf{E}^{(i)}]^{-1} &= \frac{1}{N_\Theta}(\mathbf{I}_{\kappa^{(i)}} + \mathbf{O}_{\text{off}})^{-1} \\ &\approx \frac{1}{N_\Theta}(\mathbf{I}_{\kappa^{(i)}} - \mathbf{O}_{\text{off}}) = \frac{1}{N_\Theta^2}(2N_\Theta \mathbf{I}_{\kappa^{(i)}} - \mathbf{E}^{(i)}).\end{aligned} \tag{24}$$

Furthermore, the (l,l')-th entry value can be represented as

$$([\mathbf{E}^{(i)}]^{-1})_{l,l'} = \begin{cases} \frac{1}{N_\Theta}, \text{when } l = l' \\ \frac{1}{N_{\Theta^2}}[-(\mathbf{E}^{(i)})_{l,l'}], \text{when } l \ne l' \end{cases}. \tag{25}$$

If $\theta_{l,l'}^{(i)}$ is used to denote $(\tau_l^{(i)} - \tau_{l'}^{(i)})$, when $l \ne l'$, the value of $[-(\mathbf{E}^{(i)})_{l,l'}]/N_{\Theta^2}$, denoted as $E(\theta_{l,l'}^{(i)})$, can be further expressed to

$$E\left(\theta_{l,l'}^{(i)}\right) = -\frac{1}{N_{\Theta^2}} \cdot \left(\sum_{k \in \boldsymbol{\Theta}} \cos\left(\frac{2\pi k \theta_{l,l'}^{(i)}}{N}\right) + j \sum_{k \in \boldsymbol{\Theta}} \sin\left(\frac{2\pi k \theta_{l,l'}^{(i)}}{N}\right)\right). \tag{26}$$

Because the significant path delays are smaller than N_B, the value of $\theta_{l,l'}^{(i)}$ is in the range $1 < |\theta_{l,l'}^{(i)}| < N_B$. For low complexity implementation, all possible values of $E(\theta_{l,l'}^{(i)})$ can be calculated first. Since the inter-path interference degrades sharply when $|\theta_{l,l'}^{(i)}|$ gradually becomes large, many values of $E(\theta_{l,l'}^{(i)})$ after the numerical quantization are very small and near to zero. Hence, those nonzero quantized $E(\theta_{l,l'}^{(i)})$ values can be expressed in CSD codes and searched for their common subexpressions; then, the multiplications of $E(\theta_{l,l'}^{(i)})$ can be implemented by CSD multiplications. We merge the Hessian matrix calculator into the path decorrelator, and there are four components to compose this path decorrelator: Hessian precalculators, a Hessian control unit, selectors, and a final summation. Fig. 14 shows the block diagram of this path decorrelator implemented in parallel form. The Hessian precalculator employs CSD multiplications to multiply the un-decorrelated $\mathbf{q}_\nu^{(i)}$ with the possible values of $E(\theta_{l,l'}^{(i)})$, and it is only executed once during the operation of the path decorrelator. Then, based on the value of $\theta_{l,l'}^{(i)}$, the Hessian control unit generates the control signals for selecting the corresponding results generated by the Hessian precalculators. Finally, the corresponding results are selected by the selectors and summarized by the final summation to form one path decorrelation. The design just needs N_P cycles to complete all path decorrelations in parallel form; otherwise, it needs $N_P * N_P$ cycles in serial form.

As described above, the path decorrelator efficiently avoids computing the Hessian matrix and the matrix inverse. It uses only adders and multiplexers instead of many complex multipliers and a lot of memory; besides, the redundant computations are also removed by sharing the results of the Hessian precalculators. Therefore, the path decorrelator highly reduces the hardware complexity and leads to low-power application simultaneously.

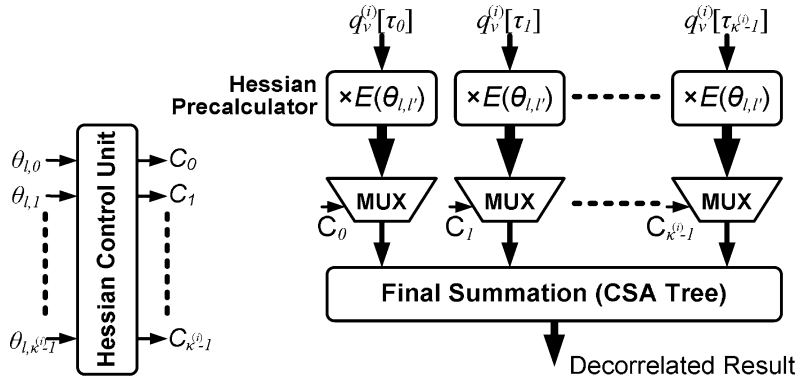

Fig. 14. Block diagram of the path decorrelator.

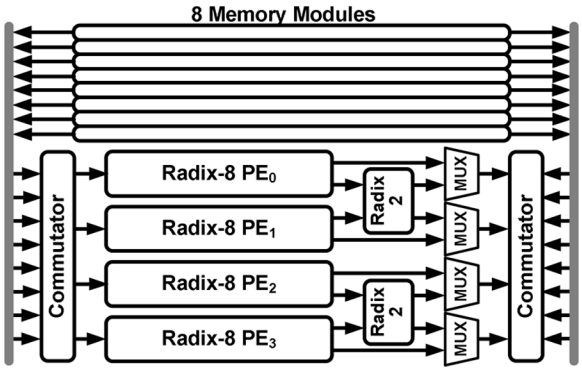

Fig. 15. Radix-8 1024-point parallel memory-based FFT architecture.

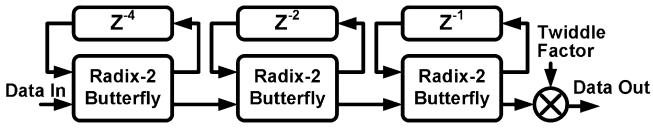

Fig. 16. Radix-8 processing element.

TABLE II
WORD LENGTHS OF SEVERAL KEY SIGNALS IN THE CHANNEL ESTIMATOR

Signal	Word Length
Received Data Input	13
Channel Response Output	16
Preamble Match Output	12
LS Estimator Output	14
STBC Decoder Output	9
IFFT/FFT Output	16
MPIC Decorrelator Output	14
Path Decorrelator Output	14

After the path decorrelator, the decorrelated gradients pass through FFT to acquire the gradients in frequency domain. Finally, the new estimated channel frequency responses are updated by subtracting these gradients from the latest estimated channel frequency responses.

F. FFT/IFFT Module

The FFT and IFFT are shared between the initialization stage and the tracking stage. Since the tracking stage tracks channel variations in an iterative way, the latency of FFT and IFFT is a main issue to achieve the design requirement. Therefore, a parallel memory-based FFT/IFFT architecture which provides multiple inputs and outputs in normal order is adopted to reduce the latency requirement (less than 1/4 of one OFDM symbol time) and to work in low clock rate. Fig. 15 illustrates the architecture of radix-8 1024-point parallel memory-based FFT with eight independent memory modules. This architecture consists of eight single port memory modules, four radix-8 processing elements (PE), two radix-2 butterfly elements, and commutators between memory modules and PEs. Two classes of PE architecture are popular in the literature: single-path delay feedback (SDF) and multi-path delay commutator (MDC) [19], [20]. Considering cost, complexity and throughput, the radix-8 PE employs an 8-point pipelined SDF FFT architecture, as shown in Fig. 16.

Only partial outputs of IFFT and partial inputs of FFT are used in the two-stage channel estimation method. Therefore, in the future, the FFT/IFFT module can be further studied with the partial FFT algorithm [21] to reduce the computational complexity and memory-access operations, and it can also be improved by the scaling algorithm [22] with shorter word length.

G. Word Length Optimization

The optimization of finite word length not only reduces hardware complexity but also guarantees acceptable system performance. The output SNR at the STBC decoder, as defined in (11)–(14), is used as a performance criterion to determine the appropriate word length of each building block. The word lengths of several key signals in the channel estimator are summarized in Table II.

V. SIMULATIONS AND DESIGN RESULTS

The performances of the proposed channel estimator are demonstrated through the simulation of an STBC-OFDM system with two transmit antennas and one receive antenna. The multipath channels adopt the International Telecommunication Union (ITU) Veh-A [23] channel model with relative path power profiles of 0, $-1, -9, -10, -15$, and -20 (dB), and the path excess delays are uniformly distributed from 0 to 50 sampling periods. Moreover, the Jakes model is used to generate a Rayleigh fading environment [24].

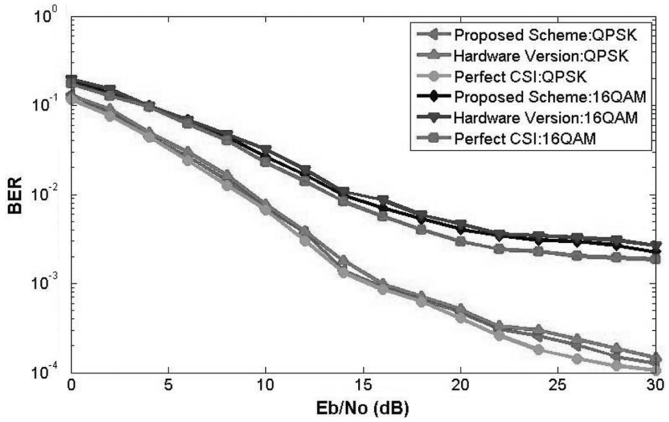

Fig. 17. BER performances at v_e of 120 km/hr.

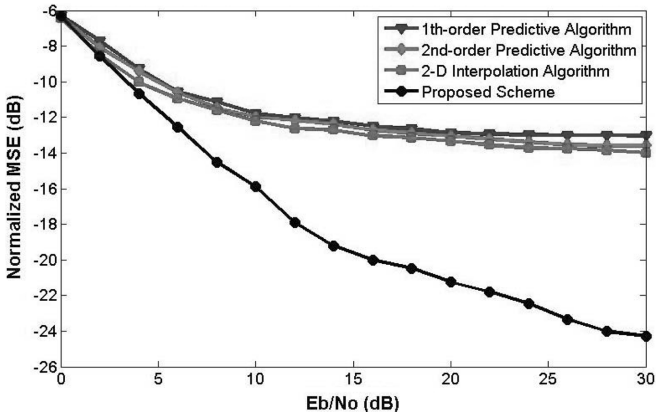

Fig. 18. Normalized MSE (relative to channel power gain) at v_e of 120 km/hr.

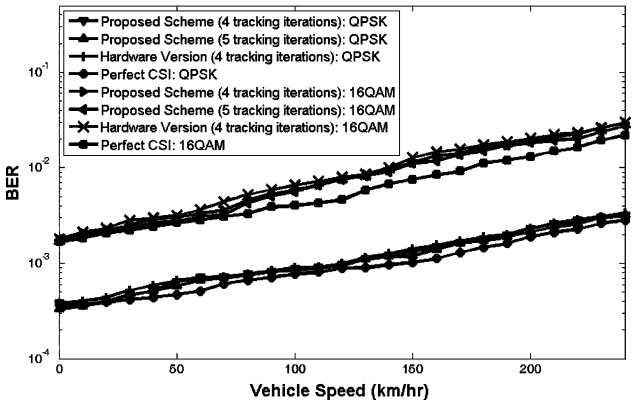

Fig. 19. BER performances versus the vehicle speed.

Fig. 17 shows the BER performances of the proposed scheme and the hardware version with four tracking iterations at the vehicle speed (v_e) of 120 km/hr which is equivalent to the Doppler frequency (f_D) of 277.8 Hz. The hardware version is simulated with fixed word length. The result of perfect channel estimation, denoted as perfect CSI, is included for benchmarking. The performance curves of the proposed scheme and the hardware version are very close to the perfect CSI curve. In QPSK modulation, the curve of the hardware version has about 0.2 dB gap in $\mathrm{E_b/N_0}$ as compared with the proposed scheme and about 0.8 dB gap in $\mathrm{E_b/N_0}$ as compared with the perfect CSI case at $\mathrm{BER} = 10^{-3}$. In 16QAM modulation, the curve of the hardware version has about 0.5 dB gap in $\mathrm{E_b/N_0}$ as compared with the proposed scheme and about 1.2 dB gap in $\mathrm{E_b/N_0}$ as compared with the perfect CSI case at $\mathrm{BER} = 10^{-2}$.

Three kinds of interpolation-based channel estimation methods, the 1st-order predictive algorithm, the 2nd-order predictive algorithm and the two dimensional (2-D) interpolation algorithm [11], [12], are simulated to make the performance comparison. Considering the IEEE 802.16e OFDMA downlink specification, these methods are executed based on the cluster structures (Fig. 13) where a cluster consists of 14 consecutive subcarriers with alternating structures in two successive time slots. These interpolation-based methods are applied as follows: 1) for each time slot, do LS channel estimations at pilot subcarriers as described in (22)–(23), where we assume that the channel within two consecutive OFDM symbols is quasi-static; 2) according to [11] and [12], among contiguous time slots, do the time-domain interpolation of the corresponding channel frequency response for each specific transceiver antenna pair; 3) perform linear frequency-domain interpolation by using pilot subcarriers and the interpolated subcarriers obtained from time-domain interpolation. Fig. 18 shows the normalized mean square errors (MSE) of channel estimation for QPSK modulation under different methods at v_e of 120 km/hr. As shown in the figure, the performance curves of the interpolation-based methods exhibit an error floor phenomenon. Generally, there are three factors contributing to the channel estimation error of the interpolation-based methods, which are AWGN noise and model errors from both time-domain and frequency-domain interpolations. At low $\mathrm{E_b/N_0}$ situation, the estimation error is mainly dominated by AWGN noise. However, the error floor phenomenon at high $\mathrm{E_b/N_0}$ is due to model errors. The longest interval between the pilot subcarriers transmitted from one antenna is 12 subcarrier spacing, and even that between the pilot and interpolated subcarriers is four subcarrier spacing. Because of both the frequency selective fading caused by larger multipath delay spreads and the time selective fading caused by higher Doppler effect, the interpolation-based methods under the situation of limited pilots in the cluster structures cannot recover the channel frequency response well. At $\mathrm{E_b/N_0} = 30$ dB, the normalized MSEs of the proposed scheme and the 2-D interpolation algorithm are about -24.3 dB and -13.9 dB. Although the interpolation-based methods have lower complexity for implementation, our proposed scheme has lower MSE of channel estimation and better performance especially in outdoor high-mobility environments.

Finally, Fig. 19 shows the BER performances under different v_e at $\mathrm{E_b/N_0} = 16$ dB. At v_e of 120 km/hr, the BER of the perfect CSI case, the proposed scheme and the hardware version with four tracking iterations for QPSK/16QAM can achieve about $8.9 \times 10^{-4}/4.6 \times 10^{-3}, 9.6 \times 10^{-4}/7.6 \times 10^{-3}$ and $9.9 \times 10^{-4}/8.0 \times 10^{-3}$, respectively, without using channel coding. In Fig. 19, we further provide the BER performance of the proposed scheme with five tracking iterations. The BER performance curves for four and five tracking iterations are very close. In other words, no further improvement in BER can be achieved after four tracking iterations with the vehicle speed

TABLE III
DESIGN RESULTS OF THE PROPOSED CHANNEL ESTIMATOR

Technology	CMOS 90nm
Sampling Frequency	11.9 MHz
Clock Frequency	83.3 MHz
Proposed Channel Estimator	
SRAM Size (bits)	196.2K
Total Area including SRAM (gates)	859,604 (3.43mm^2)
Power (mW)	43.71 @ 83.3 MHz
Proposed Channel Estimator excluding the FFT/IFFT module	
SRAM Size (bits)	7.8K
Total Area including SRAM (gates)	281,226 (1.12 mm^2)
Power (mW)	13.97 @ 83.3 MHz

up to 240 km/hr. Even at v_e of 240 km/hr which is equivalent to f_D of 555.6 Hz, the BER of the proposed scheme with four tracking iterations for QPSK/16QAM can achieve about $3.1 \times 10^{-3}/2.9 \times 10^{-2}$.

The proposed channel estimator is implemented in 90 nm CMOS technology. Several memory types are available. In our design, we relax the access time constrain and make only one read or write per memory module in the memory bank, so that we can use low cost single port register file. The area of single port register file which is 0.023 mm^2 is significant smaller than that of dual port SRAM which is 0.054 mm^2 for the size of 128 words $\times$ 38 bits.

The result of hardware implementation is listed in Table III. Since the process of four tracking iterations is enough to achieve an acceptable BER performance, an OFDM symbol time is dominated by the execution time in the initialization stage. Within a time slot (containing two OFDM symbol times), this design can support up to four tracking iterations in the tracking stage, and the iteration number can be adapted to the vehicle speed. The channel estimator outputs the decided data symbols of two OFDM symbols in each time slot. For this channel estimator, there are two clocks, 11.9 MHz and 83.3 MHz, to be used as the sampling frequency and the operation frequency, respectively. In 16QAM modulation, the uncoded throughput for this design is about 29.03 Mbps which is the number of bit transmission in a frame divided by the time duration of a frame. The area is 3.43 mm^2 and equivalent to 859 604 gates. Without the FFT/IFFT module, the area is only 1.12 mm^2 and equivalent to 281 226 gates. The power is evaluated to be 43.71 mW at the operating frequency of 83.3 MHz from a supply voltage 1 V. The power is 13.97 mW excluding the FFT/IFFT module.

Fig. 20 illustrates the hardware reduction of the proposed channel estimator. Under the same system timing requirement, the direct implementation of the two-stage channel estimation requires about 1891.2 K gates excluding the FFT/IFFT module. By using our proposed scheme and architecture mentioned in Section IV, the hardware is reduced to only 281.2 K gates, which is 14.8% of the original design. The percentage value in the bar denotes the step-by-step hardware reduction of each block as compared with the overall direct implementation architecture. In the initialization stage, the preamble match uses only adders and shifters instead of multipliers, and the SMPIC-based decorrelator efficiently reduces the execution cycles by 9.63 times as compared to the MPIC-based decorrelator. Moreover, in the tracking stage, the LS estimator only uses adders and multiplexers instead of complex multipliers and dividers. The implementation of matrix inverse is avoided in both the direct implementation and the proposed channel estimator. Since the Hessian matrix calculator is effectively merged into the path decorrelator, the path decorrelator further avoids a lot of execution cycles to compute each entry of the Hessian matrix and frees to use any multipliers and memory. Furthermore, the path decorrelator uses only adders and multiplexers instead of complex multipliers for matrix multiplication.

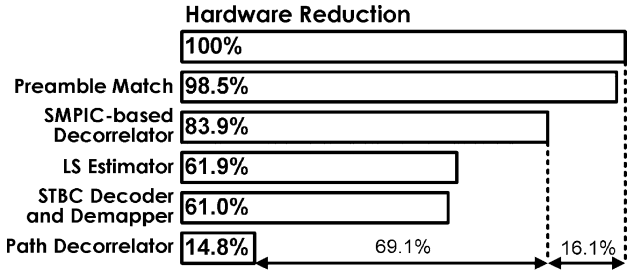

Fig. 20. Hardware reduction of the proposed channel estimator.

In summary, the interpolation-based channel estimation methods have the advantage of low implementation cost since they do not require FFT and IFFT to operate in transform domain. However, their disadvantage is difficult to estimate CSI accurately under the situation of limited pilot subcarriers over doubly selective channels. In contrast, the two-stage channel estimation method has significant performance improvement in outdoor high-mobility environments, but it requires high hardware cost. For realizing the successful high-mobility STBC-OFDM systems, the proposed channel estimator effectively improves the design complexity of the two-stage channel estimation with acceptable hardware cost while keeping the performance of the two-stage channel estimation.

VI. CONCLUSION

In this paper, a channel estimator for STBC-OFDM systems in high-mobility wireless channels is proposed. The design applied in IEEE 802.16e system adopts a high performance two-stage channel estimation method to provide precise CSI. In addition, the implementation complexity of the proposed design is reduced by 85.2% as compared with the direct implementation. When operating at v_e of 120 and 240 km/hr with $\mathrm{E_b/N_0}$ of 16 dB for QPSK modulation, the proposed design can achieve the BER of about 10^{-4} and 10^{-3} without using channel coding. As compared with interpolation-based channel estimation methods, our proposed scheme has significant performance improvements particularly in fast and selective fading channels. This channel estimator is implemented in 90 nm CMOS technology and operated at 83.3 MHz from 1 V supply voltage while drawing 43.71 mW. The design area costs 3.43 mm^2 (859 604 gates) and that excluding the FFT/IFFT module is only 1.12 mm^2 (281 226 gates), which is affordable in today's baseband system on chip. In the future, the proposed channel estimator can be implemented to be scalable for different FFT sizes as specified in IEEE 802.16e. With all these features, the proposed channel estimator can be applied to

high-mobility IEEE 802.16e systems over multipath fast fading channels.

REFERENCES

[1] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

[2] Y. H. Zeng, W. H. Lam, and T. S. Ng, "Semiblind channel estimation and equalization for MIMO space-time coded OFDM," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 2, pp. 463–474, Feb. 2006.

[3] Y. H. Chung and S. M. Phoong, "Unitary precoders for ST-OFDM systems using alamouti STBC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 9, pp. 2860–2869, Oct. 2008.

[4] *Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std 802.16-2004, Oct. 2004.

[5] *Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, IEEE Std 802.16e-2005, Feb. 2006.

[6] J. H. Park, M. K. Oh, and D. J. Park, "New channel estimation exploiting reliable decision-feedback symbols for OFDM systems," in *Proc. Int. Conf. Commun.*, Jun. 2006, pp. 3046–3051.

[7] L. Deneire, P. Vandenameele, L. Van Der Perre, B. Gyselinckx, and M. Engels, "A low-complexity ML channel estimator for OFDM," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 135–140, Feb. 2003.

[8] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3065–3073, Dec. 2001.

[9] M. L. Ku and C. C. Huang, "A derivation on the equivalence between Newton's method and DF DFT-based method for channel estimation in OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3982–3987, Oct. 2008.

[10] M. L. Ku and C. C. Huang, "A refined channel estimation method for STBC/OFDM systems in high-mobility wireless channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4312–4320, Nov. 2008.

[11] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for OFDM-based broadband transmission part II: A case study," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 571–578, Apr. 2001.

[12] T. A. Lin and C. Y. Lee, "Predictive equalizer design for DVB-T system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, vol. 2, pp. 940–943.

[13] K. Hwang, *Computer Arithmetic, Principles, Architecture, and Design*. New York: Wiley, 1979.

[14] S. Olariu, M. C. Pinotti, and S. Q. Zheng, "How to sort N items using a sorting network of fixed I/O size," *IEEE Trans. Parallel Distrib. Syst.*, vol. 10, no. 5, pp. 487–499, Mar. 1999.

[15] S. Olariu, M. C. Pinotti, and S. Q. Zheng, "An optimal hardware-algorithm for sorting using a fixed-size parallel sorting deveice," *IEEE Trans. Comput.*, vol. 49, no. 12, pp. 1310–1324, Dec. 2000.

[16] C. Y. Huang, G. J. Yu, and B. D. Liu, "A hardware design approach for merge-sorting network," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2001, vol. 4, pp. 534–537.

[17] K. E. Batcher, "On bitonic sorting networks," in *Proc. Int. Conf. Parallel Process.*, 1990, pp. 376–378.

[18] L. Horvath, I. B. Dhaou, H. Tenhunen, and J. Isoaho, "A novel, high-speed, reconfigurable demapper-symbol deinterleaver architecture for DVB-T," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 1999, vol. 4, pp. 382–385.

[19] H. Y. Lee and I. C. Park, "Balanced binary-tree decomposition for area-efficient pipelined FFT processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 889–900, Apr. 2007.

[20] Y. W. Lin and C. Y. Lee, "Design of an FFT/IFFT processor for MIMO OFDM systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 807–815, Apr. 2007.

[21] M. Li, D. Novo, B. Bougard, L. Van Der Perre, and F. Catthoor, "Generic multi-phase software-pipelined partial-FFT on instruction-level-parallel architectures and SDR baseband applications," in *Proc. Design Automation Test Eur.*, Mar. 2008, pp. 598–603.

[22] Y. Chen, Y.-C. Tsao, Y.-W. Lin, C.-H. Lin, and C.-Y. Lee, "An indexed-scaling pipelined FFT processor for OFDM-based WPAN applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 2, pp. 146–150, Feb. 2008.

[23] J. Laiho, A. Wacker, and T. Novosad, *Radio Network Planning and Optimisation for UMTS*. New York: Wiley, 2002.

[24] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.

Hsiao-Yun Chen was born in Tainan, Taiwan. She received the B.S. degree in electronics engineering from Feng Chia University, Taiwan, in 2002 and the M.S. degree in electrical engineering from National Central University, Chung-Li, Taiwan, in 2004. She is currently pursuing the Ph.D. degree in the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan.

Her research interests include baseband signal processing, integrated circuit and system designs for wireless and mobile communications.

Meng-Lin Ku was born in Taoyuan, Taiwan. He received the B.S., M.S., and Ph.D. degrees from the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 2002, 2003, and 2009, respectively.

His research interests are in the areas of wireless and mobile communications, optimization for communication engineering, and statistical signal processing.

Shyh-Jye Jou received his B.S. degree in electrical engineering from National Chen Kung University, Tainan, Taiwan, in 1982, and the M.S. and Ph.D. degrees in electronics from National Chiao Tung University, Hsinchu, Taiwan, in 1984 and 1988, respectively.

He joined Electrical Engineering Department, National Central University, Chung-Li, Taiwan, from 1990 to 2004 and became a Professor in 1997. Since 2004, he has been Professor of Electronics Engineering Department of National Chiao Tung University and became the Chairman from 2006. He was a visiting research Associate Professor in the Coordinated Science Laboratory at University of Illinois, Urbana-Champaign, during 1993–1994 academic years. In the summer of 2001, he was a visiting research consultant in the Communication Circuits and Systems Research Laboratory of Agere Systems. His research interests include design and analysis of high speed, low power mixed-signal integrated circuits, and communication integrated circuits and systems.

Dr. Jou served on the technical program committees in CICC, A-SSCC, ICCD, ISCAS, ASP-DAC, VLSI-DAT and other international conferences.

Chia-Chi Huang was born in Taiwan. He received the B.S. degree in electrical engineering from National Taiwan University in 1977 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1980 and 1984, respectively.

From 1984 to 1988, he was an RF and Communication System Engineer with the Corporate Research and Development Center, General Electric Co., Schenectady, NY, where he worked on mobile radio communication system design. From 1989 to 1992, he was with the IBM T.J. Watson Research Center, Yorktown Heights, NY, as a research staff member, working on indoor radio communication system design. Since 1992, he has been with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, currently as a Professor.