A Low-Complexity Hardware Implementation of Compressed Sensing-Based Channel Estimation for ISDB-T System

Rian Ferdian, *Student Member, IEEE*, Yafei Hou, *Senior Member, IEEE*, and Minoru Okada, *Member, IEEE*

***Abstract*—Compressed sensing (CS) is one of the hottest research topics in the sparse signal reconstruction problem. But CS implementation has a drawback of high computational complexity due to calculation between large size of matrices. In this paper, we will propose a low-complexity CS hardware realization for channel estimation in the integrated services digital broadcasting-terrestrial (ISDB-T) system using several optimization methods to reduce the implementation complexity of CS usage. Since the ISDB-T is based on orthogonal frequency division and multiplexing system, the measurement matrix of CS computation is a truncated discrete Fourier transform (DFT) matrix. We can exploit the symmetrical property of this DFT matrix to significantly reduce its multiplication complexity and random access memory usage. To achieve fast reconstruction period, this paper also provides a hardware architecture for the proposed method and its realization in field programmable gate array. The simulation results show that the proposed methods can achieve lower complexity CS-based channel estimation with almost the identical system performance with the conventional method. Moreover, the realized hardware can achieve the fastest execution time compare to that of other existing methods.**

***Index Terms*—Channel estimation, compressed sensing, ISDB-T, orthogonal matching pursuit, FPGA implementation.**

I. Introduction

MULTIPATH channel [1] will significantly reduce the transmission performance of the integrated services digital broadcasting-terrestrial (ISDB-T) system [2], [3]. The received signal will suffer from inter-symbol interference (ISI) due to multipath fading of wireless channel. To reduce its impact, the receiver can utilize an equalization process to remove the multipath effect. As the input for the equalizer, channel state information (CSI) from the channel estimation becomes one of the important aspects that determine the equalization quality. Usually, the channel estimation utilizes pilot interpolation method to estimate the CSI [4]–[6]. But, errors in interpolation method are inevitable and it requires a large number of pilot tones, so that it also will reduce the overall system bandwidth efficiency.

Recently, compressed sensing (CS) has became a new prominent technique in channel estimation [7]–[10]. Compressed sensing is a new data acquisition technique that enables the reconstruction of a sparse signal with sub-Nyquist samples [11], [12]. The implementation of CS based channel estimation in the next generation ISDB-T system [13] is offering better accuracy and less pilot utilization. The core of CS algorithms is to find the locations of the non-zero elements in sparse signal. There are two main groups of compressed sensing approach, basis pursuit and greedy pursuit [14]. Basis pursuit algorithms acquire L_p-norm to find all the locations of the non-zero element at once. While, greedy pursuit algorithms use the iterative method to get the locations. In most cases, the greedy pursuit will have some performance degradation compared to that of basis pursuit. However, it offers a lower cost of computation. The major greedy algorithms are matching pursuit (MP) and orthogonal matching pursuit (OMP). MP has the least complexity among all the CS algorithms, but it requires a high number of iterations in its process [15]. OMP reduces MPs high iterations number by acquiring least-squares in every step of its iteration, thus, it can avoid choosing the same location of non-zero element twice [16]. OMP is preferable in real-time application because it provides a short execution time. However, the implementation of OMP algorithm still requires a heavy cost of computation for its measurement matrix multiplication and least-squares computation.

There were only a few schemes for OMP hardware realization. Several proposals focused on reducing the least-squares complexity using modified Grahm-Schmidt [17], [18] and Cholesky decomposition [19], [20]. Since most of the realizations are based on real number, some methods such as Cholesky decomposition are not applicable for complex number problems. Moreover more than 80% of the OMP computation lies on the measurement matrix multiplication as mentioned in [17]. The implementation of OMP algorithm in a specific application can utilize several optimizations to reduce its complexity. Recently [21] has proposed an OMP hardware realization for radar application with an improvement in inner product computation by utilizing a zero padding FFT. However, for the application where the measurement matrix is truncated in both column and row directions, the inner product

Manuscript received July 1, 2016; revised August 25, 2016 and September 9, 2016; accepted September 12, 2016. Date of publication October 24, 2016; date of current version March 1, 2017.

The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma 630-0192, Japan (e-mail: rian.ferdian.qt7@is.naist.jp; yfhou@is.naist.jp; mokada@is.naist.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2016.2617286

using FFT method may not yield to the best reduction for its cost of computation.

In this paper, we show our results for OMP application as channel estimation in ISDB-T system. Since ISDB-T modulation is based on orthogonal frequency division and multiplexing (OFDM), the transceiver will acquire the discrete Fourier transform (DFT) and its inverse DFT (IDFT) for demodulation and modulation process. Channel estimation will fetch the pilot tones in the frequency domain after the DFT process. As the OMP computation will reconstruct the sparse CSI in the time domain, the relation between the measured pilot with the time domain CSI can be represented by a truncated DFT in the pilot position as the measurement matrix [7].

A. Contribution of This Paper

In this paper, we propose both algorithm improvements and hardware realization for the CS based channel estimation in the ISDB-T system. The contributions of this paper is briefly described as follows

- We exploit the periodicity and symmetry property which the measurement matrix inherits from DFT matrix to achieve lower complexity in the measurement matrix inner product computation.
- We optimize the random access memory (RAM) usage in the CS computation. Our proposal can avoid to save the whole measurement matrix into memory which reduces the requirement of huge memory.
- We propose a hardware architecture and its realization in field programmable gate array (FPGA). Moreover we compare the proposed hardware realization to the existing methods to show the efficiency of our proposal.

This paper is organized as follows. Section II explains the conventional method for channel estimation in the ISDB-T system and the basic of the CS and OMP algorithms. Section III describes our proposed method for the complexity reduction. Sections IV and V show the simulation results for the proposed method and give the detail of the proposed hardware architecture and its implementation results. Section VI gives the conclusions of this paper.

B. Notation

In this paper, we use bold letter for vectors and matrices. Bold lower case is for signals or matrices in the time domain and bold capital case for signals or matrices in the frequency domain. Inner product multiplication is denoted by $\langle , \rangle$. We also use $(.)^H$ to denote Hermitian transform of a matrix, notation $(.)^*$ is the conjugate of a parameter and notation *diag*(.) defines a diagonal matrix. We use notation $|.|$ to denote the absolute value of a scalar and notation $\lfloor . \rfloor$ for the floor operation. The set of complex number is denoted by notation $\mathbb{C}$.

II. Background

A. Compressed Sensing

The main purpose of the Compressed Sensing (CS) algorithm is to allow a signal reconstruction with the sub-Nyquist samples. Hence, its sampling process can be described as an under-determined linear system as

$$\mathbf{b} = \mathbf{\Psi a} + \mathbf{w}, \tag{1}$$

where $\mathbf{\Psi} \in \mathbb{C}^{N \times M}$ is the measurement matrix, $\mathbf{a} \in \mathbb{C}^{M \times 1}$ is the reconstructed signal, $\mathbf{b} \in \mathbb{C}^{N \times 1}$ is the sampled signal, N is the number of samples and $N < M$. However, if signal $\mathbf{a}$ is sparse, we can neglect the columns in matrix $\mathbf{\Psi}$ that correspond to the zero elements. Consequently, the overall system in Eq. (1) can be a determined linear system.

As proposed by Donoho [11], the CS algorithm can utilize the inner product of a matrix that has a restricted isometric property (RIP) to locate the non-zero element location α in the sparse signal $\mathbf{a}$ as

$$\alpha = \underset{i=1...M}{\arg\max} |\langle \mathbf{\Psi}_i^H, \mathbf{b} \rangle|, \tag{2}$$

where $\mathbf{\Psi}_i^H$ is the i-th row in the matrix $\mathbf{\Psi}^H$. The RIP matrix multiplication holds a special property where the location of its maximum result α will also be the location of one of the non-zero elements in signal $\mathbf{a}$.

In this paper we apply CS algorithm for wireless channel estimator. Since wireless communication system is a real-time application that requires strict execution time for its processes, greedy pursuit based CS algorithm is a good practical approach to met fast reconstruction time with acceptable hardware complexity. In this paper we use orthogonal matching pursuit (OMP) which is one of the prominent greedy pursuit based CS algorithms that offer small iteration number and good reconstruction performance [16].

Algorithm 1 OMP Algorithm

1: *init*:
2: $\mathbf{r} \leftarrow \mathbf{b}$; $\mathbf{\Phi}_0 \leftarrow [\quad]$
3: *loop*:
4: $\quad \mathbf{g} \leftarrow \langle \mathbf{\Psi}^H, \mathbf{r} \rangle$
5: $\quad \boldsymbol{\alpha} \leftarrow \underset{i=1...M}{\arg\max} |\mathbf{g_i}|$
6: $\quad$ *augment*:
7: $\quad \mathbf{\Phi}_t \leftarrow \mathbf{\Phi}_{t-1} | \mathbf{\Psi}_{\boldsymbol{\alpha}}^H$
8: $\quad$ *least square*:
9: $\quad \mathbf{a} \leftarrow arg\ min_a ||\mathbf{r} - \mathbf{\Phi}_t \mathbf{a}||^2$
10: $\quad$ *update*:
11: $\quad \mathbf{r} \leftarrow \mathbf{b} - \mathbf{\Phi}_t \mathbf{a}$
12: **if** the stopping criterion of $\mathbf{r}$ is not met **goto** *loop*.

Algorithm 1 shows the detail of the OMP computation. After the initialization step, every iteration will require inner product computation. The measurement matrix inner product as in Eq. (2) is the requirement for every CS algorithm to get the locations of the non-zero elements. The $\mathbf{\Phi}$ matrix will hold the columns from $\mathbf{\Psi}$ that correspond to the sparse locations. Next, OMP uses least-squares to get the approximate solution for the current set of non-zero element locations. To evaluate this current solution, a residue is computed as in the step 11 of Algorithm 1. The residue is the subtraction of the sampled signal $\mathbf{b}$ with the current solution contribution ($\mathbf{\Phi}_t \mathbf{a}$). If the residue value satisfies the stopping criterion, the current approximate solution can become the final output. In contrast,

Algorithm 2 Givens Rotations Algorithm

1: *input*: $\mathbf{A} \in \mathbb{C}^{d \times d}$
2: The givens algorithm create $\mathbf{Q}^H\mathbf{A} = \mathbf{R}$
3: *loop*:
4: *for* i=1; i≤d; i++;
5: *for* j=d; j≥i; j- -;
6: $Ve^{j\theta_a} = \mathbf{A}(j-1, i)$
7: $Ze^{j\theta_b} = \mathbf{A}(j, i)$
8: $\theta_1 = tan^{-1}(V/Z)$
9: $\mathbf{A}_{tmp} = \mathbf{A}(j-1 : j, i : d)$
10: $\mathbf{A}_{tmp} = \begin{bmatrix} cos\theta_1 e^{-j\theta_a} & sin\theta_1^{-j\theta_b} \\ -sin\theta_1^{-j\theta_a} & cos\theta_1^{-j\theta_b} \end{bmatrix} \mathbf{A}_{tmp}$
11: $\mathbf{A}(j-1 : j, i : d) = \mathbf{A}_{tmp}$
12: *end for*
13: *end for*

If the stopping criterion cannot be met, OMP will require an additional iteration. This residue will be the input for the next inner product multiplication. Since the current solution contribution has been removed from the residue, the maximum inner product result will always be calculated for the new location.

However, compared to the interpolation method, the implementation of OMP will still raise the hardware's resources. There are two main operations in OMP that require a high computational complexity, the inner product multiplication in step 4 and the least-squares in step 9 of Algorithm 1. Since the $\mathbf{\Psi}^H$ matrix has $M \times N$ dimension, its multiplication will also require at least $M \times N$ number of complex multipliers. In the ISDB-T system case, this multipliers requirement will become very high that make it hard to be realized in a mobile system. The practical approach is to compute the matrix multiplication iteratively row by row as proposed in [17] and [20]. However, the iterative method has a drawback of long hardware cycle period.

The most common way to solve the least-squares problem is using the matrix inversion. Furthermore, the combination of the QR decomposition and the backward substitution can provide a less complex solution for the matrix inversion. In this paper we use givens rotations for the QR decomposition method which is shown in Algorithm 2. This algorithm decomposes $\mathbf{A}$ matrix into upper triangular matrix $\mathbf{R}$ using a series of rotation matrix multiplications. Every iteration of this algorithm will change the element $\mathbf{A}(j, i)$ to be zero. The step 5 to step 8 are to define the phase difference between the element $\mathbf{A}(j, i)$ and its lower row element $\mathbf{A}(j-1, i)$. An rotation matrix which this phase difference is multiplied to the whole elements in row j and $(j-1)$. In this algoritm we use a temporary matrix $\mathbf{A}_{tmp}$ to address all the non zero elements in these j-th and $(j-1)$-th rows. The whole processes are repeated for every pair of elements in lower diagonal parts until matrix $\mathbf{A}$ becomes triangular. Then, the inversion of this triangular matrix can utilizes backward substitution easily. This algorithm has an advantage in hardware implementation because the series of rotation matrix multiplications can be realized with a systolic array of coordinate rotation digital computer (CORDIC) [22].

B. Integrated Services Digital Broadcasting-Terrestrial System

Integrated services digital broadcasting-terrestrial (ISDB-T) system is a Japan's standard for digital broadcasting that provides many services includes high definition and mobile television, digital radio and notification for disasters. ISDB-T system uses OFDM for its core modulation technique. OFDM has two main features, high spectrum efficiency and robustness to inter-symbol interference (ISI). The high spectrum efficiency can be gained because each subcarrier is orthogonal to each others, so that the adjacent subcarriers can have a smaller spacing without interferences. While the robustness to ISI can be obtained with the utilization of a cyclic prefix (CP).

Fig. 1 shows the structure of ISDB-T block diagram used in this paper. To create the transmission signal, first the mapper maps the bit stream into multilevel quadrature amplitude modulation (QAM) symbols. The pilots are then inserted in the data sequence in every 12 subcarriers. Next, the IDFT processor will create an OFDM symbol by transforming the data sequence into the time domain. To prevent ISI, the CP is inserted in the OFDM symbol by copying certain part of the symbols rear to its front. Then, the transmit signal will be sent over a multipath channel. The time domain structure of the multipath channel can be modeled as a Toeplitz matrix $\mathbf{h} \in \mathbb{C}^{(S+L-1)\times S}$ as

$$\mathbf{h} = \begin{bmatrix} h_0 & & & 0 \\ \vdots & h_0 & & \\ h_{L-1} & \vdots & \ddots & \\ & h_{L-1} & \vdots & h_0 \\ & & \ddots & \vdots \\ 0 & & & h_{L-1} \end{bmatrix}, \tag{3}$$

where L is the number of CP and S is the number of OFDM sub-carriers. Here, we assume channel has L paths which includes many zero elements for easy analysis.

As the receiver removes the CP from the received signal, the equivalent channel $\mathbf{h}_{eq} \in \mathbb{C}^{S\times S}$ can be treated as a circulant Toeplitz matrix. The matrix $\mathbf{h}_{eq}$ has an advantage that its frequency domain form $\mathbf{H}$ is a diagonal matrix. Let $\mathbf{H}$ is defined as

$$\mathbf{H} = \mathbf{F}\mathbf{h}_{eq}\mathbf{F}^H, \tag{4}$$

where $\mathbf{F} \in \mathbb{C}^{S\times S}$ is the DFT matrix, and its hermitian $\mathbf{F}^H \in \mathbb{C}^{S\times S}$ is the IDFT matrix. The structure of matrix $\mathbf{H} \in \mathbb{C}^{S\times S}$ is represented as

$$\mathbf{H} = \begin{bmatrix} H_0 & & & 0 \\ & H_1 & & \\ & & \ddots & \\ 0 & & & H_{S-1} \end{bmatrix}, \tag{5}$$

where H_i is the frequency domain CSI for i-th subcarrier. Accordingly the received signal can be described as

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{W}, \tag{6}$$

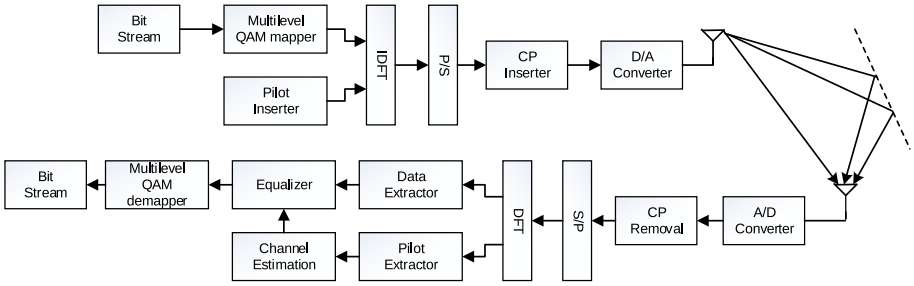


Fig. 1. Block diagram of minimum ISDB-T system.

where $\mathbf{W}$ is the receiver's noise $\mathbf{w}$ in the frequency domain. Thus, to equalize $\mathbf{X}$ from $\mathbf{Y}$, it only requires a simple diagonal $\mathbf{H}$ matrix inversion.

To get the frequency domain CSI, the receiver needs to separate the pilot tones from the demodulated signal. Then, by using the known transmitted pilot value, an observation vector $\mathbf{H}_{ob}$ can be calculated as

$$\mathbf{H}_{ob} = \frac{\mathbf{Y}_{((i-1)(S/T))}}{\mathbf{X}_{((i-1)(S/T))}}, \qquad i = 1, 2, 3 \ldots, T \tag{7}$$

where T is number of the pilot tones. The observation points are spreading evenly in the frequency domain CSI. Because of that, to get the unknown CSI values, receiver can utilize an interpolation method between two observation points [5]. After the equalization, then the receiver can utilize multilevel QAM de-mapper to convert the signal into its original bit stream form.

C. CS Based Channel Estimation in ISDB-T System

The implementation of CS algorithm in ISDB-T channel estimation can lead to better channel reconstruction quality and less pilot utilization. To apply the CS algorithm in ISDB-T channel estimation, we need to redefine the observation vector in Eq. (7) as an multiplication of a RIP matrix with a sparse signal as in Eq. (1). Because CSI is sparse in the time domain [23], the sparse requirement can be met by changing the frequency domain CSI into time domain one as

$$\mathbf{H} = diag(\mathbf{F}\mathbf{h}_{[0]}), \tag{8}$$

where $\mathbf{h}_{[0]}$ is the time domain CSI impulse response. The Eq. (7) then can be redefined as

$$\mathbf{H}_{ob} = \mathbf{F}_P\mathbf{h}_{[0]} + \mathbf{W}. \tag{9}$$

where $\mathbf{F}_P \in \mathbb{C}^{N\times M}$ is a truncated DFT matrix. The row elements of $\mathbf{F}_P$ are from the DFT matrix rows that correspond to the pilot positions, and N equals to T as number of pilots. Because the receiver is only able to equalize a signal with maximum delay spread less than CP, the column elements of $\mathbf{F}_P$ can be chosen to be the first M columns of DFT matrix that correspond to the CP length. Now, Eq. (9) has met all the CS requirements, as $\mathbf{H}_{ob}$ is the sampled vector, $\mathbf{h}_{[0]}$ is the sparse reconstructed signal and $\mathbf{F}_P$ is the measurement matrix [24]. The CS representation of this OFDM channel estimation is shown in Fig. 2.

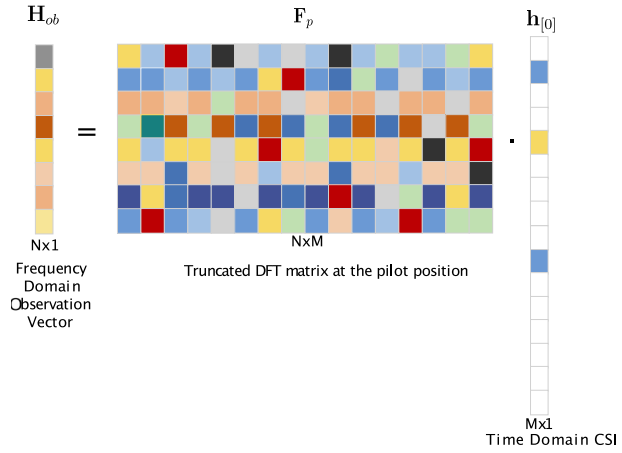


Fig. 2. Compressed sensing representation for OFDM channel estimation problem.

The CP and pilot configuration will influence the structure of the measurement matrix $\mathbf{F}_P$. CP length will define the number of column in the $\mathbf{F}_P$ matrix and the number of pilot will determine the row length in the $\mathbf{F}_P$ matrix. The size of $\mathbf{F}_P$ matrix is closely related to the complexity in both inner product and least square operation in the OMP algorithm. Moreover, in term of reconstruction performance, the structure of the measurement matrix will define the number of path that the system can reconstruct [25]. For OMP system, the maximum sparsity number (κ_{max}) is defined as

$$\kappa_{max} = \left\lfloor \frac{1}{2}\left(\frac{1}{\mu(\mathbf{F}_P)} + 1\right)\right\rfloor, \tag{10}$$

where $\mu(\mathbf{A})$ is the mutual coherence of $\mathbf{A} \in \mathbb{C}^{N\times M}$ matrix which defined as

$$\mu(\mathbf{A}) = \max_{1\leqslant i,j\leqslant M, i\neq j} \frac{\left|\mathbf{a}_i^H\mathbf{a}_j\right|}{\|\mathbf{a}_i\|_2.\|\mathbf{a}_j\|_2}. \tag{11}$$

where $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$ matrix.

The performance of the OMP reconstruction for the channel estimation is guaranteed when the number of multiple taps in the channel (κ) is less than or equal to κ_{max} [14].

III. Proposed Methods

A. Multiplier Reduction

The measurement matrix has some unique properties which inherited from DFT matrix. Let us describe the measurement

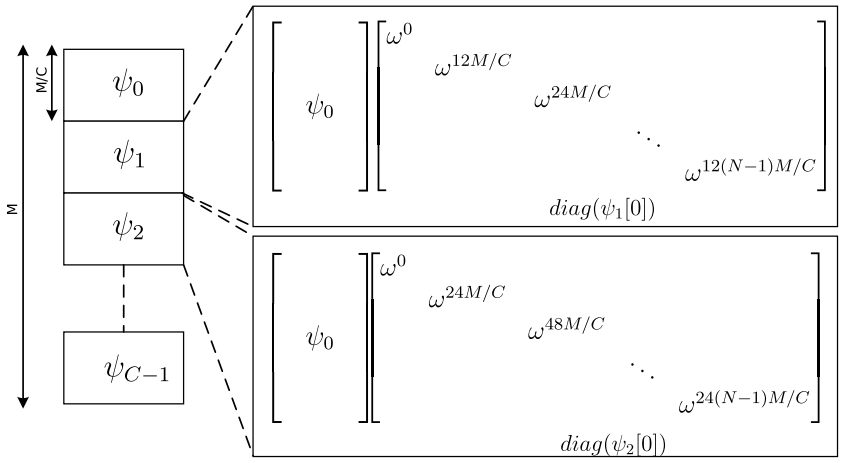


Fig. 3. Measurement matrix decomposition ($p = 12$).

matrix $\mathbf{\Omega}$ as the Hermitian of $\mathbf{\Psi}$ matrix represented as

$$\mathbf{\Omega} = \mathbf{\Psi}^H = \mathbf{F}_P^H = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^p & \omega^{2p} & \cdots & \omega^{(N-1)p} \\ \omega^0 & \omega^{2p} & \omega^{4p} & \cdots & \omega^{2(N-1)p} \\ \omega^0 & \omega^{3p} & \omega^{6p} & \cdots & \omega^{3(N-1)p} \\ \omega^0 & \omega^{4p} & \omega^{8p} & \cdots & \omega^{4(N-1)p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega^0 & \omega^{Mp} & \omega^{2Mp} & \cdots & \omega^{(M-1)(N-1)p} \end{bmatrix}, \quad (12)$$

where ω is $e^{-2\pi j/S}$ from S size DFT and p is the pilot period which is set as 12 for ISDB-T system. The $\mathbf{\Omega}$ matrix elements are well structured that for the element at a certain k-th row and l-th column can be obtained as

$$\Omega(k, l) = \omega^{(pkl)}. \quad (13)$$

Since the elements in $\mathbf{\Omega}$ are exponent function, any matrix element can be obtained with multiplication of two others elements as

$$\Omega(k_1 + k_2, l) = \Omega(k_1, l) \times \Omega(k_2, l) = \omega^{(k_1 pl)} \omega^{(k_2 pl)}. \quad (14)$$

We can divide the $M \times N$ size $\mathbf{\Omega}$ matrix into C sub-matrices ψ_i with size of $M/C \times N$ ($i = 0, \ldots, C-1$) in the row direction as

$$\mathbf{\Omega} = \left[\boldsymbol{\psi_0}\, \boldsymbol{\psi_1}\, \boldsymbol{\psi_2}\, \ldots\, \boldsymbol{\psi_{C-1}}\right]^T. \quad (15)$$

Let us define the first row of i-th sub-matrix as $\psi_i[0]$ which can be represented as

$$\begin{aligned} \psi_i[0] &= [\psi_i[0](0), \ldots, \psi_i[0](N-1)] \\ &= \left[\omega^{p(\frac{iM}{C})\times 0}, \ldots, \omega^{p(\frac{iM}{C})(N-1)}\right]. \end{aligned} \quad (16)$$

and its element $\psi_i[0](j)$ is $\omega^{p(\frac{iM}{C})\times j}$ with $j \in [0, \ldots, N-1]$.

From the Eq. (14), we can get the following relationship between the i-th sub-matrice ($i \in [1, \ldots, C-1]$) and first sub-matrix ψ_0 as

$$\psi_i = \psi_0 diag\{\psi_i[0]\}. \quad (17)$$

Such property can be shown as in Fig. 3.

In addition, the element of k-th row of i-th sub-matrix $\psi_i[k](j)$ can be calculated using the element of its first row $\psi_i[0](j)$ and that of the k-th row of first sub-matrix $\psi_0[k](j)$ as

$$\psi_i[k](j) = \psi_i[0](j) \times \psi_0[k](j) \quad j \in [0, \ldots, N-1]. \quad (18)$$

Therefore, to compute the inner product computation $\mathbf{g} = \langle \mathbf{\Psi}^H, \mathbf{r} \rangle$ as in the step 4 of Algorithm 1, we first define a matrix $\mathbf{\Gamma}$ which is formed by the first row vectors of all sub-matrices as

$$\mathbf{\Gamma} = \left[\psi_0[0]\, \psi_1[0]\, \psi_2[0]\, \ldots\, \psi_{C-1}[0]\right]^T. \quad (19)$$

Then, we can calculate C rows in the step 4 of Algorithm 1 as

$$\begin{bmatrix} g_{[k]} \\ g_{[k+M/C]} \\ g_{[k+2M/C]} \\ \vdots \\ g_{[k+M(C-1))/C]} \end{bmatrix} = \mathbf{\Gamma} diag(\psi_0[k])\mathbf{r}, \quad (20)$$

and the whole inner product computation $\mathbf{g}$ can be executed in M/C cycles.

Next, we exploited the periodicity and symmetry property of the measurement matrix to decompose the $\mathbf{\Gamma}$ matrix into some matrices with lower complexity form. The periodicity property in this matrix can be explained as that some ω elements will be identical value after the modulo operation with 2π. We can express the elements in $\mathbf{\Gamma}$ matrix as

$$\mathbf{\Gamma}(k, l) = e^{-j\left[mod\left(\frac{(2\pi Mpkl)}{SC}, 2\pi\right)\right]}, \quad (21)$$

where $mod(a, b)$ defines the modulo operation of dividend a and divisor b. Here, the elements phases are in the range of 0 to 2π, because of that, Eq. (21) also can be redefined as

$$\mathbf{\Gamma}(k, l) = e^{-j2\pi\left(\frac{[mod(Mpkl, SC)]}{SC}\right)}, \quad (22)$$

which also corresponds to the same element phases in range 0 to 2π. Now, to find the number of different elements at each k-th row, we can reduce the constant fraction $(Mpk)/SC$ to its

lowest term using greatest common divisor (GCD). Let d_k is the GCD between (Mpk) and SC as

$$d_k = GCD(Mpk, SC). \tag{23}$$

Then Eq. (22) can be redefined as

$$\mathbf{\Gamma}(k,l) = e^{-j2\pi\left(\frac{[mod(Mpkl/d_k, SC/d_k)]}{SC/d_k}\right)}, \tag{24}$$

Now, it is easy to see that at k-th row of $\mathbf{\Gamma}$ matrix, the number of different elements is SC/d_k. The total different elements of $\mathbf{\Gamma}$ matrix can be expressed as

$$U_T = \sum_{k=1}^{C-1} \frac{SC}{d_k}. \tag{25}$$

In Eq. (23), variable M is the guard interval length in range such as $\frac{1}{4}S$, $\frac{1}{8}S$, $\frac{1}{16}S$ and $\frac{1}{32}S$. While, the pilot spacing p in an ISDB-T system is set as 12. Because of that, the GCD result between Mpk and SC will at least have a value of $4M$. In result, the number of different elements in $\mathbf{\Gamma}$ matrix will be smaller than NC.

Since the elements of $\mathbf{\Gamma}$ matrix can be represented using complex unit circle and these elements will divide this complex unit circle evenly. Thus, by using only the elements in the first quadrant of complex unit circle (phase less than $\pi/2$), we can obtain all the different elements located in other quadrants. Let us define the phase of each element as

$$\theta_{k,l} = 2\pi \frac{[mod(Mpkl/d_k, SC/d_k)]}{SC/d_k}, \tag{26}$$

in term of quadrants, the elements of $\mathbf{\Gamma}$ matrix can be defined as

$$\mathbf{\Gamma}_{(k,l)} = \begin{cases} \mathbf{\Gamma}_{q1(\theta_{k,l})} = e^{-j\theta_{k,l}}, & 0 < \theta_{k,l} \le \frac{\pi}{2} \\ \mathbf{\Gamma}_{q2(\theta_{k,l})} = -j\mathbf{\Gamma}_{q1(mod(\theta_{k,l}, \frac{\pi}{2}))}, & \frac{\pi}{2} < \theta_{k,l} \le \pi \\ \mathbf{\Gamma}_{q3(\theta_{k,l})} = -\mathbf{\Gamma}_{q1(mod(\theta_{k,l}, \frac{\pi}{2}))}, & \pi < \theta_{k,l} \le \frac{3\pi}{2} \\ \mathbf{\Gamma}_{q4(\theta_{k,l})} = j\mathbf{\Gamma}_{q1(mod(\theta_{k,l}, \frac{\pi}{2}))}. & \frac{3\pi}{2} < \theta_{k,l} \le 2\pi \end{cases} \tag{27}$$

The multiplications with -1, j and $-j$ are needed to recover all the elements in other quadrants. Therefore, we only require the U elements in first quadrant as

$$U = U_T/4. \tag{28}$$

Then we can utilize the algorithm strength reduction technique to reduce the complexity in the $\mathbf{\Gamma}$ matrix inner product. The aim of the strength reduction technique is to calculate the $\mathbf{\Gamma}$ matrix with only the U kind of multipliers to reduce its complexity. The matrix $\mathbf{\Gamma}$ can be represented using three parts as

$$\mathbf{\Gamma} = \mathbf{\Gamma}_3\mathbf{\Gamma}_2\mathbf{\Gamma}_1. \tag{29}$$

First, using the symmetry property of this measurement matrix, we multiply the input vector elements that correspond to the coefficients in the quadrant 2, 3 and 4 of the complex plane with $-j$,-1 and j respectively. Then, we sum all the elements that correspond to the same coefficient. This operation is described using $\mathbf{\Gamma}_1 \in \mathbb{C}^{U\times N}$ matrix which defined as

$$\mathbf{\Gamma}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & -j & 0 & -1 & \cdots & 0 \\ 0 & 1 & 0 & -j & 0 & \cdots & j \\ & & & \vdots & & & \\ 0 & j & 0 & -1 & 0 & \cdots & 1 \end{bmatrix}. \tag{30}$$

Then we can multiply the output from $\mathbf{\Gamma}_1$ with the U unique coefficients, which represented by $\mathbf{\Gamma}_2 \in \mathbb{C}^{U\times U}$ matrix as

$$\mathbf{\Gamma}_2 = \begin{bmatrix} e^{\frac{-j2\pi d_1}{SC}} & 0 & 0 & 0 & 0 \\ 0 & e^{\frac{-j4\pi d_1}{SC}} & 0 & 0 & 0 \\ 0 & 0 & e^{\frac{-j6\pi d_1}{SC}} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & e^{\frac{-j\pi}{2}} \end{bmatrix}. \tag{31}$$

The diagonal of $\mathbf{\Gamma_2}$ matrix is filled with U different elements in $\mathbf{\Gamma}$ matrix, which values in range of $e^{\frac{-j2\pi d_i}{SC}}$ to $e^{\frac{-j\pi}{2}}$. The last step is to sum the output from $\mathbf{\Gamma}_2$ that corresponds to the same row at the original $\mathbf{\Gamma}$ matrix. This operation can be described by the $\mathbf{\Gamma}_3 \in \mathbb{C}^{C\times U}$ matrix as

$$\mathbf{\Gamma}_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{32}$$

The matrix $\mathbf{\Gamma}_1$ only consists of 1, -1, j and $-j$ elements, because of that, the multiplication complexity with this matrix can be neglected and the $\mathbf{\Gamma}_1$ matrix complexity lies on the summation process only. The additional multipliers only exist in the diagonal matrix $\mathbf{\Gamma}_2$ which require U complex multipliers. The $\mathbf{\Gamma}_3$ is a summation matrix which its elements are just 1 and 0. This matrix can be easily implemented using an adder tree. Using the proposed method, we only need an additional U multipliers to compute $C-1$ rows at once. Thus, the average multipliers requirement for one row computation in inner product operation can be reduced to $(U+N)/C$, while the conventional implementation requires N multipliers.

B. Memory Reduction

The memory requirement for OMP implementation is very huge because we need to save the whole measurement matrix with the size of $N \times M$. The implementation in [18] and [20] stated that it already used 62% and 69% of the FPGA Virtex 6 RAM resource respectively for real number implementation. Thus it can be estimated that for the complex number implementation, it will require at least two times memory resource. However, in the implementation for the ISDB-T system, we can also exploit the DFT matrix properties to reduce the RAM requirement. There are three parts in the OMP computations from Algorithm 1 that require $\mathbf{\Gamma}$ data, the inner product in the

step 4, the least squares in the step 9 and the residue computation in the step 11. Instead of saving the $N \times M$ complex number, here we can reduce it to $N + M$ size.

Since the hardware will compute the inner product computation as Eq. (20) iteratively, we can use the measurement matrix property in Eq. (14) to compute the $diag(\psi_0[k])\mathbf{r}$ as

$$diag(\psi_0[k])\mathbf{r} = diag(\psi_0[1]) \times diag(\psi_0[k-1])\mathbf{r}. \quad k > 0 \tag{33}$$

Here, using the previous iteration result, for each new iteration, we only need to multiply the same $diag(\psi_0[1])$ which is the second row vector from the ψ_0 sub-matrix. Using this method, we only need to store N complex coefficients from the second row of the ψ_0 sub-matrix for the inner product computation.

The least squares and residue operations require the $\mathbf{\Phi}_t$ matrix, which formed by the columns of the matrix $\mathbf{\Psi}$ that correspond to the sparse locations. Since the number of pilots are always larger than the channel sparsity, the least squares here can be categorized as overdetermined linear system problem. The Moore-Penrose method is the common approach to solve this overdetermined linear system problem as

$$\mathbf{a} = (\mathbf{\Phi}_t^H \mathbf{\Phi}_t)^{-1} \mathbf{\Phi}_t^H \mathbf{r}. \tag{34}$$

The matrix $(\mathbf{\Phi}_t^H \mathbf{\Phi}_t)$ can be obtained by taking the row and column of $(\mathbf{\Psi}^H \mathbf{\Psi})$ matrix that correspond to the sparse element locations. The $(\mathbf{\Psi}^H \mathbf{\Psi}) \in \mathbb{C}^{M \times M}$ matrix has a Hermitian Toeplitz structure as

$$(\mathbf{\Psi}^H \mathbf{\Psi}) = \begin{bmatrix} \rho_0 & \rho_1 & \cdots & \rho_{M-2} & \rho_{M-1} \\ \rho_1^* & \rho_0 & \rho_1 & & \rho_{M-2} \\ \vdots & \rho_1^* & \ddots & \ddots & \vdots \\ \rho_{M-2}^* & & \ddots & \rho_0 & \rho_1 \end{bmatrix}. \tag{35}$$

Therefore, by only storing the first row M coefficients, we can obtain all the matrix elements.

Furthermore, the same $(\mathbf{\Psi}^H \mathbf{\Psi})$ matrix can also be used to compute the residue in each iteration. As proposed in [17] after the first iteration, the computation of inner product in the line 4 of algorithm 1 can be defined as

$$\mathbf{g} = \mathbf{\Psi}^H \mathbf{b} - \mathbf{\Psi}^H \mathbf{\Psi} \mathbf{a}. \tag{36}$$

Since $(\mathbf{\Psi}^H \mathbf{b})$ is the inner product from first iteration, we only need to compute $(\mathbf{\Psi}^H \mathbf{\Psi} \mathbf{a})$. Because of the DFT property, the $(\mathbf{\Psi}^H \mathbf{\Psi})$ matrix will be a diagonal centric matrix where the value in its diagonal significantly larger than off-diagonal ones. By exploiting this property, the computation of $\mathbf{\Psi}^H \mathbf{\Psi} \mathbf{a}$ in Eq. (36) can be approximated by removing the current set of maximum value from the vector $\mathbf{g}$. Therefore, to compute both least squares and residue, we only need an additional M size memory to save the first row of the $(\mathbf{\Psi}^H \mathbf{\Psi})$ matrix. The total memory usage in our proposed method will be $(N + M)$ size, while the conventional way requires $N \times M$ memory.

The computation steps for the proposed OMP operation can be described by the flowchart in Fig. 4. The first computation is the inner product which is realized using the proposed algorithm strength reduction method. However different with the conventional OMP, the number of iteration for inner product

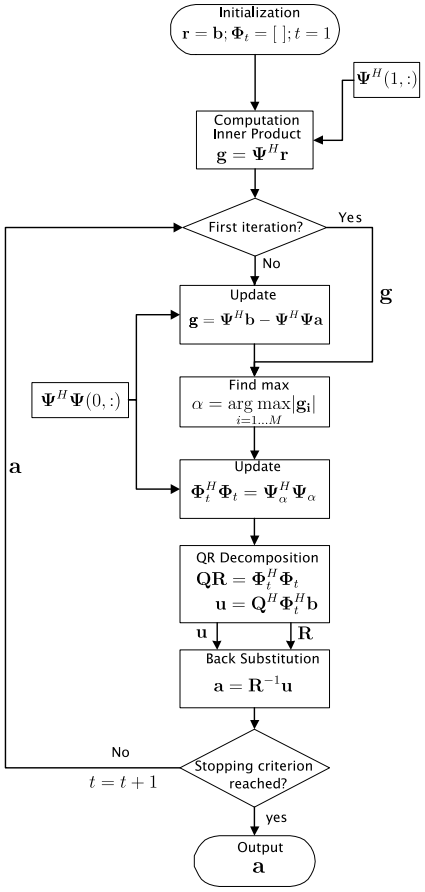


Fig. 4. Flowchart of the proposed OMP.

computation is M/C and we only calculate this inner product once. In the following iterations, the hardware only updates this inner product result $\mathbf{g}$ vector. This computation is realized in the residue unit by subtracting the vector $\mathbf{g}$ from $\mathbf{\Psi}^H \mathbf{\Psi} \mathbf{a}$. The maximum finder unit will find the location of the largest value in the vector $\mathbf{g}$. The maximum value will be used to update the $\mathbf{\Phi}^H \mathbf{\Phi}$ matrix. Since the hardware only save the first column of the $\mathbf{\Psi}^H \mathbf{\Psi}$, the $\mathbf{\Phi}^H \mathbf{\Phi}$ matrix is formed by only utilizing shifter and conjugate operation.

IV. Hardware Architecture

In this section, we provide a VLSI architecture for the proposed CS based ISDB-T channel estimation. There are two considerations in this proposed implementation, a fast execution time to meet the real-time strict deadline and a low resource hardware to support the mobility in ISDB-T one-seg service. Fig. 5 shows the block diagram for the proposed VLSI architecture. The hardware mainly consists of five processing elements: inner product, maximum finder, $\mathbf{\Phi}^H \mathbf{\Phi}$ matrix updater, least squares and residue unit.

Table I shows the theoretical memory usage in this hardware implementation. In this table, κ defines the sparsity in the channel impulse response, where $\kappa << N < M$. We save the static variables in the dedicated block RAM (BRAM). While the dynamic variables are saved in the registers, so it can be updated in parallel. By utilizing the proposed

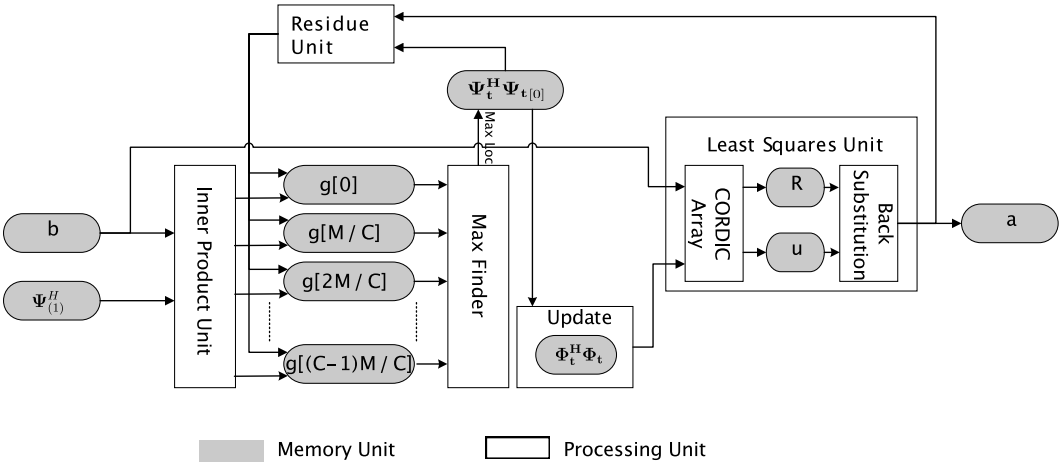


Fig. 5. Proposed hardware's block diagram.

TABLE I
THEORETICAL MEMORY REQUIREMENT

Memory Element	**b**	**a**	**g**	**u**	**R**	$\Psi_{(1)}$	$\Psi^H \Psi_{(0)}$	$\Phi^H \Phi$
Type	BRAM	BRAM	Reg	Reg	Reg	BRAM	BRAM	Reg
Size	N	κ	M	κ	$\kappa\kappa$	N	N	$\kappa\kappa$

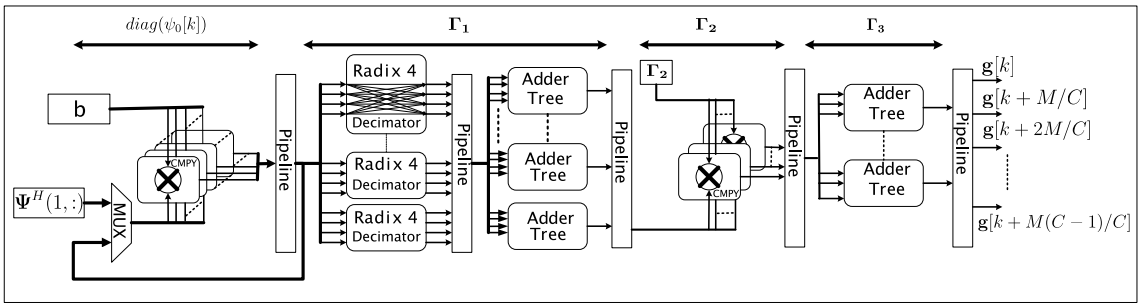


Fig. 6. Hardware architecture for inner product unit.

memory reduction technique, we can avoid to save the whole complex $\mathbf{\Psi}$ matrix which can significantly reduces the RAM usage.

The least squares operation is realized in two steps. The first step is the QR decomposition using the givens rotations technique as described in algorithm 2. This operation is done by the systolic array of CORDIC as proposed in [26]. Since $\mathbf{\Psi}^H\mathbf{\Psi}$ is a complex matrix, each processing unit in this systolic array is based on three angles complex rotation (TACR) CORDIC method as proposed in [22]. The outputs from this systolic array are vector $\mathbf{u}$ which is equal to $\mathbf{Q}^H\mathbf{\Phi}^H\mathbf{b}$ and the triangular $\mathbf{R}$ matrix. As the second step, we utilize a back substitution method to get the approximate solution for the channel impulse response $\mathbf{a}$ in each iteration.

The details of hardware architecture for the proposed inner product unit is shown in Fig. 6. This inner product unit is an iterative realization of the proposed strength reduction method. To compute the inner product $\mathbf{g} = \langle \mathbf{\Psi}^H, \mathbf{r} \rangle$, this module will compute C number of $\mathbf{\Psi}^H$ rows at once in each iteration, thus the whole inner product computation only require M/C clock cycles. This module can be divided into 5 sub-modules. In this implementation, each part is separated by a pipelined register to obtain faster clock speed. In this unit, only the first and the fourth sub-modules that require the complex multiplication. The first sub-module is the realization of $diag(\psi_0[k_1])$ multiplication from Eq. (20) which requires N multipliers. While the fourth sub-module uses U number of multipliers as the realization of $\mathbf{\Gamma}_2$ matrix multiplication from Eq. (31). The second sub-module is implemented with parallel of radix 4 decimators to realize the multiplication with $\mathbf{\Gamma}_1$ matrix from Eq. (30) which only contains 1, j, -1 and $-j$ elements. The third and the fifth sub-modules are implemented with adder threes to perform the summation computation.

TABLE II
SIMULATION PARAMETERS

Modulation	OFDM
Channel model	ITU 6 Typical Urban
FFT size	4096
Guard interval	1/4 of OFDM symbol
Equalization	Zero Forcing
Pilot	256 Subcarriers
Data	3849 Subcarriers
Noise	AWGN
Bandwidth	5.575 MHz

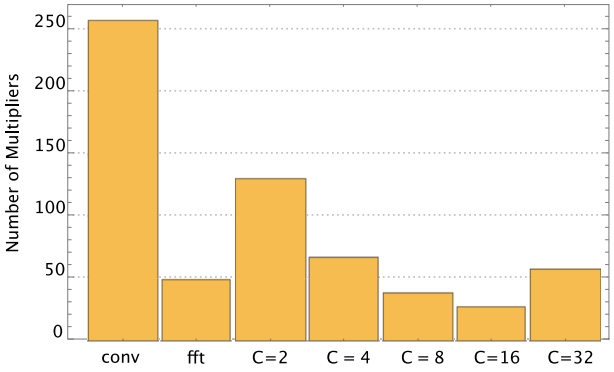


Fig. 7. Average multiplier requirement for different C.

V. EXPERIMENTAL RESULTS

A. Simulation Results

In this paper, we choose the same OMP parameters with the implementation in [18] and [20] for comparison. The parameters are $N = 256$, $M = 1024$ and sparse elements κ=12. This configuration equal to the ISDB-T system with FFT size = 4096, CP = 1/4 symbol length. Before implementing the hardware, we use MATLAB to simulate the proposed method performance. The simulation parameters are shown in the Table II.

First, we simulate the proposed inner product method with different C from Eq. (25) to get the optimum complexity reduction for the OMP configurations. Fig. 7 shows the average multipliers requirement for each row in the inner product computation. Here the configuration with $C = 16$ can achieve the optimum average 26 multipliers. While, the conventional method and the FFT method require average 256 and 48 multipliers respectively. Thus, the proposed method can save 89.8% multipliers requirement compared to that of the conventional method and 45.9% reduction compared to that of the FFT method.

The bit error rate (BER) performance of the proposed method in different modulation scheme is shown in Fig. 8. We compare our proposed method with that of conventional system where no any complexity reduction has been designed. It can be seen that, the proposed method can achieve almost identical BER performance to that of conventional system but our proposed method has low complexity. We also use the MATLAB simulation to find the minimum fixed point data representation which will not affect our proposed system

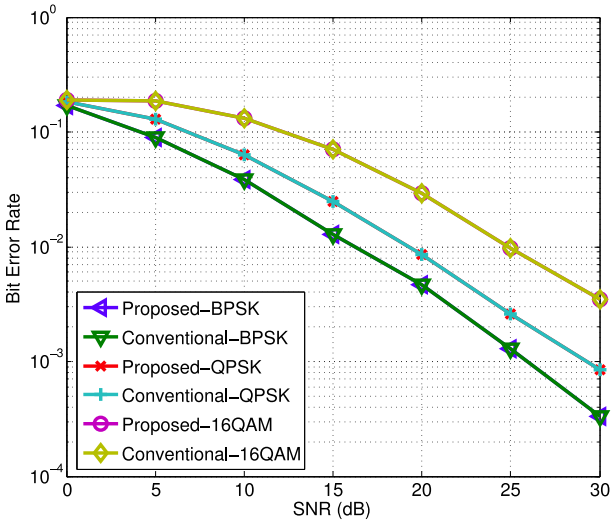


Fig. 8. BER performance of the proposed and conventional method in different modulation schemes.

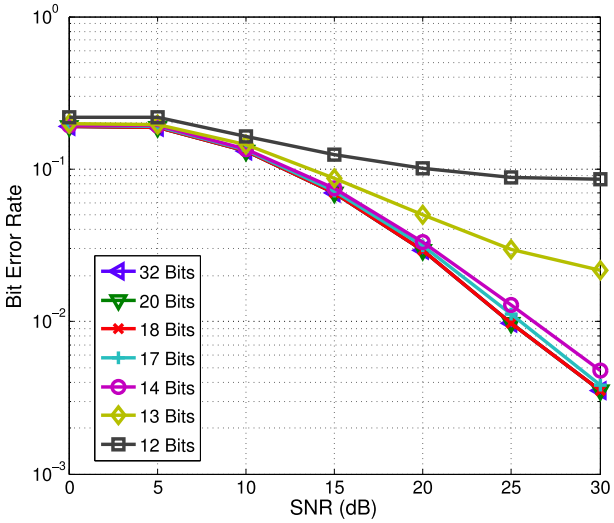


Fig. 9. BER performance with different fixed point data representations (16QAM).

performance to benefit the hardware implementation. Fig. 9 shows the bit error rate performance for this simulation. It can be seen that by reducing the data representation to using 18 bits will not largely affect the BER performance. However, the BER will start to increase if using 17 bits as fixed point for data representation, and the whole system will fail to operate if using 12 bits for data representation.

B. Implementation Results

The proposed hardware is realized in Verilog HDL and synthesized into Xilinx Virtex 6 FPGA. Based on the MATLAB simulation results, the hardware is realized with specification $C = 16$ and data representation with 18 bits as fixed points. The measurement matrix Ψ has a dimension of $N = 256$ and $M = 1024$. The sparsity degree of the channel impulse response is $\kappa = 12$.

The synthesis results for the FPGA resources utilization can be seen in table IV. Using the proposed method, the hardware

TABLE III
COMPARISON TO EXISTING DESIGN

Hardware	Measurement Matrix Size	Sparsity	Clock Frequency	Reconstruction Time	Data Format
Intel Core i7 [27]	64 x 512	12	3 GHz	25ms	single precision real data
NVIDIA GTX480 [28]	512x8192	64	N/A	15ms	N/A
FPGA Virtex 6 [18]	256 x 1024	12	100MHz	158.7 μs	18 bit fixed point real data
FPGA Virtex 6 [20]	256 x 1024	36	120MHz	350 μs	18 bit fixed point real data
FPGA Virtex 7 [21]	512 x 2048	12	165MHz	391 μs	hybrid single precision and fixed point complex data
FPGA Virtex 6 (proposed)	256 x 1024	12	100MHz	11.97 μs	18 bit fixed point complex data

TABLE IV
FPGA XILINX VIRTEX 6 RESOURCE UTILIZATION

Max Frequency	100 MHz
Occupied Slices	104,160 (35%)
DSP48 Cores	1,808 (89%)
BRAM	61.4kb (1.6%)
Register	184 kb (31 %)

can be fit into Xilinx Virtex 6 XC6VSX475T. The proposed architecture only require 1.6% BRAM and 31% register utilization. Moreover, this design be operated with 100 MHz clock speed.

The inner product multiplication execution time requirement can be calculated as

$$T_{ip} = \left(\frac{M}{C} + 5\right) \times T_{clock} \tag{37}$$

Since $C = 16$ and $M = 1024$, the M/C will be equal to 64, thus, it will require 69 clock cycles which is equal to 690 ns. While the least squares execution time can be calculated as

$$T_{ls} = 5\kappa \times T_{clock}. \tag{38}$$

The least squares will require 60 clock cycle or 600 ns. The rest of the hardware modules execution time are as follows. The maximum finder requires 10 clock cycle for $M = 1024$, the $\mathbf{\Phi}^H\mathbf{\Phi}$ needs κ clock cycle and the residue unit require 12 clock cycle. Therefore, the total reconstruction time can be calculated as

$$T_{tot} = T_{ip} + \kappa(T_{ls} + 22 + \kappa), \tag{39}$$

for the sparsity degree as 12, the reconstruction time is about 11.97 μs.

The comparison between the proposed method with the existing OMP hardware design can be seen in Table III. The implementation in [27] and [28] show the conventional OMP realization by software. It can be seen that, the reconstruction time are larger than 10 ms. These realizations show that it is very important for the hardware implementation to achieve faster reconstruction time.

The hardware dedicated OMP in [18] and [20] worked on real data reconstruction. Both the implementations provided a new approach for the least squares operation but still using a conventional way in the inner product calculation. While the implementation in [21] shows a inner product computation using FFT. From this table, it can be seen that our proposed hardware design has the fastest reconstruction time compared to that of the existing hardware dedicated OMP design.

VI. CONCLUSION

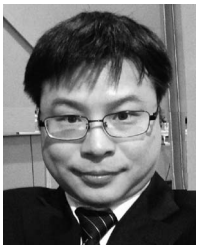
In this paper, we focused on solving the bottleneck in the OMP hardware implementation for CS based channel estimation. The main computational complexity in this implementation is the measurement matrix inner product multiplication. However, for the OFDM channel estimation, we can utilize the unique structure of measurement matrix, which is from a truncated DFT matrix, to reduce the inner product complexity. This paper proposed the strength algorithm reduction technique to achieve smaller multipliers usage in parallel inner product computation. Furthermore, we exploited the Hermitian and Toeplitz structure of the measurement matrix to avoid the requirement of whole matrix for memory. We have simulated the proposed method performance for ISDB-T OFDM system. From the simulation result, the proposed method can achieve the same bit error rate (BER) performance compare to that of the conventional method. Moreover, we also provided a VLSI architecture realization of the proposed method. The hardware is implemented in Xilinx Virtex 6 FPGA. The synthesis results show the proposed method can be fit into Xilinx Virtex 6 FPGA and achieves the fastest reconstruction time compared to that of the existing methods.

REFERENCES

[1] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. Commun. Syst.*, vol. 11, no. 4, pp. 360–393, Dec. 1963.
[2] "Transmission system for digital terrestrial television broadcasting," ARIB Standard STD-B31, Rev. 1.9, 2010.
[3] M. Takada and M. Saito, "Transmission system for ISDB-T," *Proc. IEEE*, vol. 94, no. 1, pp. 251–256, Jan. 2006.
[4] J.-S. Baek and J.-S. Seo, "Efficient pilot patterns and channel estimations for MIMO-OFDM systems," *IEEE Trans. Broadcast.*, vol. 58, no. 4, pp. 648–653, Dec. 2012.
[5] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. Broadcast.*, vol. 48, no. 3, pp. 223–229, Sep. 2002.
[6] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349–1387, Sep. 1985.
[7] G. Taubock, F. Hlawatsch, D. Eiwen, and H. Rauhut, "Compressive estimation of doubly selective channels in multicarrier systems: Leakage effects and sparsity-enhancing processing," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 255–271, Apr. 2010.
[8] X. Ma, F. Yang, W. Ding, and J. Song, "Novel approach to design time-domain training sequence for accurate sparse channel estimation," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 512–520, Sep. 2016.

[9] W. Ding, F. Yang, C. Pan, L. Dai, and J. Song, "Compressive sensing based channel estimation for OFDM systems under long delay channels," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 313–321, Jun. 2014.
[10] Z. Ma, S. Guo, H. Liu, and M. Okada, "Sliding-MOMP based channel estimation scheme for ISDB-T systems," *Math. Problems Eng.*, vol. 2016, Jun. 2016, Art. no. 4571359.
[11] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[12] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
[13] S. Saito *et al.*, "8K terrestrial transmission field tests using dual-polarized MIMO and higher-order modulation OFDM," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 306–315, Mar. 2016.
[14] K. Hayashi, M. Nagahara, and T. Tanaka, "A user's guide to compressed sensing for communications systems," *IEICE Trans. Commun.*, vol. E96-B, no. 3, pp. 685–712, Mar. 2013.
[15] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
[16] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
[17] A. Septimus and R. Steinberg, "Compressive sampling hardware reconstruction," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Paris, France, 2010, pp. 3316–3319.
[18] L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, "High-speed compressed sensing reconstruction on FPGA using OMP and AMP," in *Proc. 19th IEEE Int. Conf. Electron. Circuits Syst. (ICECS)*, Seville, Spain, Dec. 2012, pp. 53–56.
[19] J. L. V. M. Stanislaus and T. Mohsenin, "High performance compressive sensing reconstruction hardware with QRD process," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seoul, South Korea, May 2012, pp. 29–32.
[20] H. Rabah, A. Amira, B. K. Mohanty, S. Almaadeed, and P. K. Meher, "FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 10, pp. 2209–2220, Oct. 2015.
[21] Y. Quan, Y. Li, X. Gao, and M. Xing, "FPGA implementation of real-time compressive sensing with partial Fourier dictionary," *Int. J. Antennas Propag.*, vol. 2016, Jan. 2016, Art. no. 1671687.
[22] A. Maltsev, V. Pestretsov, R. Maslennikov, and A. Khoryaev, "Triangular systolic array with reduced latency for QR-decomposition of complex matrices," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kos, Greece, May 2006, pp. 385–388.
[23] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.
[24] M. Cheraghchi, V. Guruswami, and A. Velingker, "Restricted isometry of Fourier matrices and list decodability of random linear codes," in *Proc. 24th Annu. ACM-SIAM Symp. Discrete Algorithm.*, New Orleans, LA, USA, 2013, pp. 432–442.
[25] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
[26] D. Boppana, K. Dhanoa, and J. Kempa, "FPGA based embedded processing architecture for the QRD-RLS algorithm," in *Proc. Annu. IEEE Symp. Field Program. Custom Comput. Mach. (FCCM)*, Napa, CA, USA, Apr. 2004, pp. 330–331.
[27] A. Borghi, J. Darbon, S. Peyronnet, T. F. Chan, and S. Osher, "A simple compressive sensing algorithm for parallel many-core architectures," *J. Signal Process. Syst.*, vol. 71, no. 1, pp. 1–20, 2013.
[28] Y. Fang, L. Chen, J. Wu, and B. Huang, "GPU implementation of orthogonal matching pursuit for compressive sensing," in *Proc. 17th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Tainan, Taiwan, Dec. 2011, pp. 1044–1047.

Rian Ferdian received the B.S. and M.S. degrees from the School of Electrical Engineering and Informatics, Institut Tekonologi Bandung, Bandung, Indonesia, in 2008 and 2012, respectively, all in electrical engineering. He is currently pursuing the Ph.D. degree with the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. Since 2014, he has been a Lecturer with the Faculty of Information Technology, Andalas University, Indonesia. His research includes circuit for telecommunication system, digital broadcasting technologies, and digital signal processing.



Yafei Hou received the B.S. degree in electronic engineering from the Anhui University of Technology and Science, China, in 1999, the M.S. degree in computer science from Wuhan University, China, in 2002, and the Ph.D. degrees from Fudan University, China, and the Kochi University of Technology, Japan, in 2007. He was a Post-Doctoral Research Fellow with Ryukoku University, Japan, from 2007 to 2010. He was a Research Scientist with Wave Engineering Laboratories, ATR Institute International, Japan, from 2010 to 2014. Since 2014, he has been an Assistant Professor with the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. His research interests are communication systems, wireless networks, and signal processing. He is a member of IEICE.

Minoru Okada received the B.E. degree from the University of Electro-Communications, Tokyo, Japan, in 1990, the M.E. and Ph.D. degrees from Osaka University, Osaka, Japan, in 1992 and 1998, respectively, all in communications engineering. In 2000, he joined the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan, as an Associate Professor and became a Professor. From 1993 to 2000, he was a Research Associate with Osaka University. From 1999 to 2000, he was a Visiting Research Fellow with the University of Southampton, Southampton, U.K. His research interests are wireless communications, including WLAN, WiMAX, CDMA, OFDM, and satellite communications. He was a recipient of the Young Engineer Award from the Institute of Electrical, Information, and Communication Engineers (IEICE) in 1999. He is a member of the Institute of Image, Information, and Television Engineers of Japan, IEICE of Japan, and the Information Processing Society of Japan.