Folding Memory Shared Processor Array (FMSPA) Architecture for Channel Estimation of Downlink OFDMA IEEE 802.16e System

Savitri Galih\textsuperscript{1}, Trio Adiono\textsuperscript{2}, Adit Kurniawan\textsuperscript{3}, Iskandar\textsuperscript{4}

\textsuperscript{1}Department of Informatics Widyatama University
JI Cikutra 204 Bandung, Indonesia
\textsuperscript{1,2,3,4}School of Electronics and Informatics, Bandung Institute of Technology
JI Ganesha 10 Bandung, Indonesia

Abstract—The implementation of complex signal processing algorithms are required to achieve robust transmission, whereas mobile wireless application require low power dissipation. This paper describes an algorithm and a corresponding hardware architecture for the implementation of OFDMA 802.16e channel estimation. The advantage of the proposed architecture are low power and efficient resource utilization since we use iterative memory shared architecture that exploits reutilization of the processor elements and memory units. The higher data access scheme is utilized by scheduled memory sharing with common bus. Furthermore, we increase parallel efficiency by folding the architecture to reduce the number of processor elements.

Keywords—channel estimation; OFDMA; mobile WiMAX

I. INTRODUCTION

Recent wireless communication system are characterized by high bit rate data transmitted over severely time and frequency varying channels. The implementation of complex signal processing algorithms are required to achieve robust transmission, whereas mobile wireless application require low power dissipation. In recent years numerous research contributions have appeared on the topic of channel transfer function estimation. MMSE channel estimation has been known as a superior performance channel estimation especially in low SNR. However, this algorithm has high computational complexity which prohibits its direct real-time implementation low complexity partial-sampled MMSE\textsuperscript{[1,3,7,8]} (Minimum Mean Square Error) channel estimation for IEEE 802.16e (mobile WiMAX) Orthogonal Frequency Division Multiplexing Access (OFDMA) downlink system. Galih et.al \textsuperscript{[15]} propose partial sampling MMSE which reduce original MMSE channel estimation complexity by reducing the size of MMSE weight matrix. The hardware implementation of the MMSE OFDMA channel estimation basically employs large matrix vector multiplication corresponds to the number of OFDMA subcarriers. The most area efficient architecture for matrix vector multiplication is sequential architecture which consist of only one processor element. However this minimum-area solution can not meet the low latency requirement of mobile WiMAX OFDMA system. Fast but large matrix vector multiplication architecture, i.e Systolic array architecture \textsuperscript{[10, 11, 12, 13]} has advantages in regularity and high throughput but still suffers the overlapping data storages, idle processing and high area requirement. MSPA (Memory Sharing Processor Array) \textsuperscript{[9]} overcomes these problems by minimize the data storage by sharing memory units to several processor array and achieve better efficiency than conventional systolic architecture, but the data throughput is not suitable and far beyond the clock allocation compute based on standard parameter i.e sample period is much larger than the computational delay of MSPA hardware unit. In this paper we will present methods that jointly optimize algorithm and hardware performance and improve the parallel efficiency of MSPA architecture for DMMSE channel estimation architecture by fold the processor array and reducing the number of processing element through time multiplexing, the proposed architecture is called Folding MSPA.

II. SYSTEM MODEL

The OFDM/OFDMA system with pilot based channel estimation is given in figure 1.

![Fig. 1 Block Diagram of The Pilot Based OFDM System](image)

The data bits provided from the source are converted from serial to parallel to form parallel data of some subchannels\textsuperscript{[5]}. Each parallel subchannel modulated to complex QAM symbols of $N_q$ active subcarriers. The modulated data with other null carrier as guardband and DC form $N$ subcarriers. This data sequence of length $N \{X_k\}$ are then fed into IDFT
block symbol by symbol to transform them into time domain
and generate an OFDM signal \{x_n\} with the following
equation:
\[
x_n = \text{IDFT} \left\{ X_k \right\} = \sum_{k=0}^{N-1} X_k e^{j2\pi nk/N},
\]
(1)
Where N is the DFT length or 
\( n = 0,1,\ldots, N-1 \) the number of
subcarriers. To prevent inter-symbol interference (ISI), a cyclic prefix of \( N_p \)
samples is inserted at the beginning of every symbol. After
D/A conversion, the signal is transmitted through the
frequency selective time varying fading channel with additive
noise.

Assumed that the impulse response of the multipath fading channel is given by [18]:
\[
h(t, \tau) = \sum h_r(t) \delta(\tau - \tau_r).
\]
(2)

Where \( h_r(t) \) and \( \tau_r \) are the gain and delay of the \( r \)-th path,
respectively. The received signal, which has been corrupted by the multipath fading channel and contaminated by the
additive white Gaussian noise can be formulated as
\[
y(t) = \sum r h_r(t) x(t - \tau_r) + w(t).
\]
(3)

Where \( x(t) \) is the continuous-time representation of the transmitted discrete-time signal, \( x_n \). The received continuous-
time signal then convert back to a discrete –time signal \( y_n \), the receiver do synchronization, downsampling, and removes the
cyclic prefix. The simplified baseband model of the received samples takes the form of :
\[
y_n = \sum_{l=0}^{L-1} h(n-l) x(n-l) + w(n)
\]
(4)

Where \( L \) is the number of sample-spaced channel taps, \( w(n) \) is additive white Gaussian noise (AWGN) sample with zero
mean and variance of \( \sigma^2_w \) and \( h(l) \) is the time domain channel
impulse response (CIR) for the current OFDM symbol. It is
assumed that time and frequency synchronization is perfect.

FFT transforms \( y_n \) to the frequency domain received base
band data :
\[
Y_k = \text{FFT}(y_n) = X_k H_k + W_k
\]
(5)

Where \( H \) and \( W \) are FFT of \( h \) and \( w \) repectively.
Following FFT block, the pilot signals are extracted and the
Channel Estimation is carried out to obtain estimated channel
response \( \hat{H}_k \) for the data sub-channels. Then the transmitted
data is estimated by equalization process :
\[
\hat{X}_k = \frac{Y_k}{\hat{H}_k}
\]
(6)

After signal demapping, the source binary information
data are re-constructed at the receiver output.

OFDMA is based on OFDM modulation. Based on the
OFDM principle, the pilot both in time domain and in
frequency domain is assigned for channel estimation calculation process . The OFDMA downlink IEEE 802.16e
symbol structure is using pilots, data, and zero subcarriers.
The symbol is first divided into basic clusters and zero carriers
are allocated. Pilots and data carriers are allocated within each
cluster. Figure 2 below depicts the cluster structure

![Downlink OFDMA 802.16e cluster/tile structure](image)

**III. CHANNEL ESTIMATION**

In this section, the different types of channel estimators
classified in this paper are explained. After channel
estimation process at pilot subcarrier position, the channel
responses at the rest of data subcarrier are estimated by
interpolation. First is interpolation at time domain which has
2 symbols time spacing. In this paper we use linear
interpolation for time domain interpolation because it is
sufficient for small time spacing. \( H \) is estimated by vertically
1D linear interpolation, after vertical time interpolation, tile
structure is described at figure 3.a.

**A. MMSE Channel Estimation**

The MMSE channel estimator employs the second order
statistics of the channel condition to minimize the mean-
square error. The major disadvantage of the MMSE estimator
is its high complexity, which grow exponentially with the
observation sample. The frequency domain MMSE estimate
of channel response is given by[4]:
\[
\hat{H}_{P,\text{MMSE}} = R_{H,H}^{-1} + \sigma^2_n \left( X_P X_P^H \right)^{-1} \hat{H}_{P,\text{LS}}
\]
(7)

Where \( \hat{H}_{P,\text{LS}} \) is the LS estimate of channel condition at pilot
position, \( \sigma_n^2 \) is the variance of noise , \( X_P \) is a matrix
containing the transmitted pilot on its diagonal, \( R_{H,H} \) is the
channel autocorrelation matrix defined by
\[
R_{H,H} = \text{E}[H_h H_h^H]
\]
(8)

For this case, the correlation function between the channel
frequency response value is given by[5] :
\[
E[H_m H_n^H] = \begin{cases} 1, & m = n \\
1 - e^{-j2\pi (m-n)/N} & m \neq n
\end{cases}
\]
(9)

From equation (9 ) we can get \( R_{H,H} \) .
MMSE interpolation for all subcarrier can be perform by modifying the MMSE estimator at equation (7) to obtain all data subcarrier's channel responses, with this equation[1]:

$$\hat{H}_{\text{MMSE}} = R_{HH} \left( R_{HH} + \sigma_n^2 \left( X_p X_p^H \right) \right)^{-1} \hat{H}_{LS} \quad (10)$$

= \hat{Q} \hat{H}_{PLS}

The MMSE estimator (7 and 10) uses a priori knowledge of \( \sigma_n^2 \) (or SNR) and \( R_{HH} \), and is optimal when these statistics of the channel are known. As will become clear from the further discussion, SNR value can be predefined: higher target SNRs are preferable to obtain more accurate estimates. Also the robust estimator design necessitates account for the worst correlation of the multipath channel, namely when the channel power-delay profile (PDP) is uniform [14].

B. Down-sampled MMSE (DMMSE) Channel Estimation

To reduce the complexity of the MMSE estimator in equation (10), we must reduce the size of correlation matrix and thus also the size of the weight matrix \( Q \). This can be achieved by sampling the subcarrier positions in one symbol and thus also the size of the weight matrix \( Q \). This can be done by reducing the dimension of the cross correlation matrix, namely when the channel power-delay profile (PDP) is uniform [14].

For the original MMSE channel estimator, one element of the cross correlation matrix is given by equation (9)

$$r_{H,H'} = E \left[ H_n H_n^* \right] = \begin{cases} 1, & d = n \\ 1 - \frac{j2\pi}{N_p} (d-n)/N_p, \quad d \neq n \end{cases} \quad (11)$$

Where \( d = 1 \) to \( D \), \( n = 0 \) to \( N_p \), \( N_p = 360 \) is the number of down-sampled subcarriers from \( N = 840 \) original subcarriers in one OFDMA symbol.

We used subcarrier number 1, 3, 5. The picture below shows which subcarrier is sampled to form down-sampled MMSE weight matrix. This picture display \( m \), \( n \) and \( d \) indexed subcarriers from two subsequent tile of one OFDMA frame.

Fig. 3. (a) 1 indexed pilot subcarriers after time interpolation (number of time interpolated pilot = \( L = 240 \)), (b) \( n \) indexed interpolated subcarrier or all subcarriers in one symbol (number of all subcarriers = \( N = 840 \)), (c) \( d \) indexed subcarrier or down sampled subcarrier in one symbol (number of down-sampled subcarriers = \( D = 360 \))

IV. HARDWARE IMPLEMENTATION

The selection of an appropriate hardware architecture for the implementation of DMMSE Algorithm depends on the system specification, and the constraints on power, area and speed. These considerations is used to obtain the optimal architecture by balancing the algorithm throughput requirement with the capacity of hardware's building blocks. The optimization of hardware to increase the hardware efficiency is performed in two steps. First, the clock allocation of the channel estimation hardware is calculated to the requirements of OFDMA mobile WiMAX system. Second, increasing hardware efficiency by exploiting effective data storage schemes and implements architectural transformation [parhi] such as retiming, pipelining, parallel processing, and folding which support more dense operations with concurrency or time serial execution to achieve low power and high throughput communication system. Hardware efficiency . By utilizing basic concept of parallelism and time multiplexing an algorithm can be mapped into a specific architecture with extensively varying throughput and latency. (iscas08b IV).

A. MSPA Architecture

The architecture of the Memory Sharing Processor Array (MSPA) consists of a processor array with direct links and buses connected with memory units and their address generation unit (AGU). The memory units can supply the multiple data stream into the processor arrays. The memory units store and load input data to processor array according to the control signal generated by the AGU.[9] Memory Sharing Processor Array (MSPA) is different from systolic array, the architecture is designed under the specific number of Processing Elements (PE). The number of PE depend on the algorithm size (i.e 6x6 matrix vector multiplication has 6 algorithm size). Any number of PEs can be selected for an algorithm size. Those processor elements work synchronously at discrete time steps. MSPA For the practical design, the number of processing elements is fixed to ~half of the algorithm size. For MSPA architecture design for DMMSE channel estimation the number of PE is 180 for 360 x 240 matrix vector multiplication [15]. Picture 4 below shows MSPA Architecture for DMMSE Channel Estimation.

![MSPA Architecture](image)

Fig. 4. MSPA Architecture for Partial Sampled MMSE Channel Estimation

B. Folding Transformation

Folding is serially ordered execution of some algorithmic [16] as shown in fig 5 below
Input to PE is supplied by a multiplexer, which select first data \( b_1 \) or intermediate result \( b_2, b_3 \). The output of PE is then folded back to its input to calculate \( b_2, b_3 \).

V. FOLDING MSPA ARCHITECTURE

The objective of this paper is to choose the best (high parallel efficiency, low power) architecture for the Downlink OFDMA IEEE 802.16e Channel Estimation algorithm, therefore we will emphasize this paper in how we choose the best architecture and show the architecture implementation results.

The proposed architecture, Folding MSPA, implement time multiplexing to the MSPA architecture for DMMSE channel estimation to match the clock allocation suitable to the standard parameter (table 1).

In this paper we extend the parallel efficiency of MSPA architecture for DMMSE channel estimation architecture by fold the processor array and reducing the number of processing element(PE) or executing unit (EXU) through time multiplexing, the proposed architecture is called Folding MSPA. The selection of PE’s number is fitted with the clock allocation for channel estimation module in system hardware. This clock allocation is calculated under assumptions of low power specification and IEEE 802.16e primitive and derived parameters. The clock allocation is defined by frequency clock times useful symbol time in clock cycle. In this research we use 56 MHz frequency clock for low power consideration.

The following alinea, we explain methodology for the choice of an appropriate processor array size.

Based on [9] Processor Elements numbers of array processor determine by theorem:

\[
T = \left\lfloor \frac{n}{N_{PE}} \right\rfloor \tag{12}
\]

where \( T \) is the optimal mapping vector on processing elements which achieves the minimum total execution time for the uniform recurrence algorithm with the two \( n \) iteration loops of \( N_{PE} < n \), where \( n = \) algorithm size (i.e, \( n = 6 \) for 6x6 Matrix Vector Multiplication). For practical concern, we fixed \( T \) value to \( T = \{1,2\} \) therefore \( N_{PE} = n/2 \).

Total operation time of Channel Estimation Block with MSPA architecture:

\[
t_{MSPA} = \frac{C}{N_{PE}} + N_{PE} \tag{12}
\]

Where : \( C = \) Matrix Vector Multiplication Complexity = Matrix Column Number \( \times \) Matrix rownumber; \( N_{PE} = \) Processor Element Numbers.

Total operation time of Channel Estimation Block with the folding MSPA architecture is

\[
t_{FMSPA} = F \left( \frac{C}{N_{PE} \cdot F} + N_{PE} \right) + 2(N_{PE} + F) \tag{13}
\]

Where : \( F = \) Folding/Iteration number

Processor number chosen for Channel Estimation block must satisfy operation time equal with the clock allocation.

\[
CA = F \left( \frac{C}{N_{PE} \cdot F} + N_{PE} \right) + 2(N_{PE} + F) \tag{14}
\]

Where \( CA = \) clock allocation.

We used following primitive parameters and derived parameters from IEEE 802.16e 8.4.2.3 and 8.4.2.4 to determine clock allocation

<table>
<thead>
<tr>
<th>Primitive Parameters</th>
<th>Remark</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW(MHz)</td>
<td>Nominal Channel Bandwidth (MHz)</td>
<td>10</td>
</tr>
<tr>
<td>( n_{used} )</td>
<td>number of used subcarriers</td>
<td>840</td>
</tr>
<tr>
<td>( n )</td>
<td>sampling factor</td>
<td>1.142857143</td>
</tr>
<tr>
<td>( G )</td>
<td>Cyclic Prefix</td>
<td>0.03125</td>
</tr>
<tr>
<td>( N_{FFT} )</td>
<td>1024</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Derived Parameters</th>
<th>Remark</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_{s} ) = floor(( n \cdot BW/8000 ))x8000</td>
<td>sampling frequency (Hz)</td>
<td>11142000</td>
</tr>
<tr>
<td>( Df = F_{s}/N_{FFT} )</td>
<td>subcarrier spacing (Hz)</td>
<td>11156.25</td>
</tr>
<tr>
<td>( T_{s} = 1/Df )</td>
<td>Useful symbol time (s)</td>
<td>8.96359E-05</td>
</tr>
<tr>
<td>( T_{g} = G \cdot T_{s} )</td>
<td>CP Time (s)</td>
<td>2.80112E-06</td>
</tr>
<tr>
<td>( T_{S} = T_{g} + T_{s} )</td>
<td>OFDMA Symbol time (s)</td>
<td>9.2437E-05</td>
</tr>
</tbody>
</table>

Freq clock (MHz) 56

Clock Allocation = freq.cock x useful symbol time (clockcycle) 5020
Based on the calculated clock allocation, to complete the DMMSE channel estimation i.e 360 x 240 Matrix Vector Multiplication, the number of PE is fixed to 18 with 10 iteration input to folding MSPA Architecture. To perform the iteration, multiplexers are added to the input of each PE in folding MSPA architecture as shown at picture below.

**Fig. 6 Folding MSPA Architecture**

**VI. IMPLEMENTATION RESULT**

In this section, we compare the folding MSPA architecture with other architectures from the reference. The table below, shows the hardware performance comparison. The sequential architecture has higher computation time than the computed clock allocation for the OFDMA mobile WiMAX channel estimation calculated as shown from table 2. The other two architectures (systolic array and MSPA) have much lower computation time than the clock allocation, as a result, the architectures have more iddle time during computation process of channel estimation system. Folding MSPA architecture is more suitable with the considered system, since the architecture has equivalent computation time with the stated allocation clock.

Dynamic Power Consumption is defined by [19]

$$P_{\text{dynamic}} = \alpha C V_{\text{DD}}^2 f$$  \hspace{1cm} (13)

Where : \(\alpha = \) activity factor = 0.5 for dynamic gates and 0.1 for static gates, \(C = \) gate capacitance, \(V_{\text{DD}} = \) Voltage Drain Drain of transistor gate.

\(\alpha, C, V_{\text{DD}}\) are constant for all architecture, since they use same kind of transistor gate. Assume that \(B = \alpha C V_{\text{DD}}^2 = 2 \text{ mW/MHz} \). We can compare the power consumption of the architectures at the table II.

In addition, the proposed architecture has higher processor utility than the three other architectures, showed by its highest parallel efficiency. The parallel efficiency is defined as sequential computation time divide by processor number times the parallel computation time.

**VII. CONCLUSION**

In this paper we introduce Folding MSPA Architecture for downlink OFDMA IEEE 802.16e (mobile WiMAX) system. This architecture is suitable for MMSE channel estimation which require large matrix vector multiplication, since its computation time is equal with clock allocation computed from IEEE 802.16e standard parameter. Moreover, this architecture has higher parallel efficiency than systolic array and original MSPA architecture.

**REFERENCES**


**TABLE II**

<table>
<thead>
<tr>
<th>Architecture Type</th>
<th>Number of Processor Elements</th>
<th>Total Operati on Time</th>
<th>Power Consumption (mW)</th>
<th>Architectu re Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>1</td>
<td>86.400</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Systolic Array</td>
<td>360</td>
<td>722</td>
<td>341.6</td>
<td>0.332</td>
</tr>
<tr>
<td>(Milovanovic et al)</td>
<td>180</td>
<td>660</td>
<td>373.3</td>
<td>0.727</td>
</tr>
<tr>
<td>MSPA (Kunieda)</td>
<td>45</td>
<td>2202</td>
<td>112</td>
<td>0.872</td>
</tr>
<tr>
<td>Folding MSPA (Proposed)</td>
<td>45</td>
<td>2202</td>
<td>112</td>
<td>0.872</td>
</tr>
</tbody>
</table>


