SUMMARY A time-domain equalization (TEQ) algorithm is presented to shorten the effective channel impulse response to increase the transmission efficiency of the 54 Mbps IEEE 802.11a orthogonal frequency division multiplexing (OFDM) system. In solving the linear equation $Aw = B$ for the optimum TEQ coefficients, $A$ is shown to be Hermitian and positive definite. The $LDL^T$ and $LU$ decompositions are used to factorize $A$ to reduce the computational complexity. Simulation results show high performance gains at a data rate of 54 Mbps with moderate orders of TEQ finite impulse response (FIR) filter. The design and implementation of the algorithm in field programmable gate array (FPGA) are also presented. The regularities among the elements of $A$ are exploited to reduce hardware complexity. The $LDL^T$ and $LU$ decompositions are combined in hardware design to find the TEQ coefficients in less than 4 $\mu$s. To compensate the effective channel impulse response, a radix-4 pipeline fast Fourier transform (FFT) is implemented in performing zero forcing equalization. The hardware implementation information is provided and simulation results are compared to mathematical values to verify the functionalities of the chips running at 54 Mbps.

**key words:** time domain equalization, OFDM, $LDL^T$ and $LU$ decompositions, IEEE 802.11a, FPGA, pipeline FFT

1. Introduction

Orthogonal frequency division multiplexing (OFDM) has been adopted in many applications such as digital audio broadcasting (DAB) and asymmetrical digital subscriber line (ADSL). In the last few years, OFDM broadband wireless communication systems have received a considerable attention. IEEE 802.11a [1], which is also based on OFDM technique, is accepted as one of the world’s standards for wireless LAN (WLAN) in the 5 GHz band. In an OFDM communication system, the broad band is partitioned into many orthogonal sub-channels in which data is transmitted in a parallel fashion. Typically, a guard interval using a cyclic prefix (CP) is inserted to avoid inter-symbol interference and inter-channel interference. This guard interval is required to be at least equal to, or longer than, the maximum channel delay spread. This reduces the transmission efficiency by a factor of $\frac{N_g}{N - N_g}$, where $N_g$ is the length of the guard interval and $N$ is the number of sub-channels. When the maximum channel delay spread is long, the efficiency is significantly reduced for OFDM systems with 64 sub-channels as in IEEE 802.11a system. To avoid this problem, a time domain pre-fast Fourier transform (FFT) equalizer can be introduced to shorten the effective channel impulse response. Most TEQ algorithms are developed for low data rate systems and have high computational complexity [3]–[9]. Millisecond-solutions for the TEQ coefficients are acceptable when the system’s transmission rate is low, such as in ADSL. For 54 Mbps IEEE 802.11a system, the OFDM symbol interval is 4.0 $\mu$s, so the millisecond-solutions and the conventional TEQ algorithms are not suitable.

In this paper, an equalization technique based on the minimum mean square error (MMSE) criterion is adopted for high rate IEEE 802.11a OFDM system. The emphasis is on practical implementation of the algorithm in hardware. In solving the linear equation $Aw = B$ to obtain the optimum TEQ coefficients that minimize the error signal, the regularities among the elements of $A$ are exploited to design the matrix multiplication module. The $LDL^T$ and $LU$ decompositions are also used instead of finding the matrix inversion directly to reduce the computational complexity. After the optimum TEQ coefficients are found, the zero forcing equalization is employed in order to compensate the effective channel impulse response in the frequency domain. Thus a high performance FFT module needs to be designed. To speed up the calculation, the parallelism of radix-4 pipeline architecture is adopted to increase the computational efficiency in the hardware implementation of 64-point FFT. The technique is verified using simulation and implemented in FPGA to operate at 54 Mbps. The time taken to find the TEQ coefficients in hardware can be realized in only a few microseconds ($\mu$s).

2. Equalization Technique in OFDM Systems

In WLAN IEEE 802.11a, OFDM technique is adopted as the modulation and demodulation method in the physical layer. Punctured convolution encoder, Viterbi decoder, interleaving, and de-interleaving are included to improve the system performance. For higher data transmission rates of 48 Mbps and 54 Mbps, the system employs 64-QAM. The key components of the system are the inverse fast Fourier transform (IFFT) in the transmitter and the FFT in the receiver. Assume the length of the original channel is longer than the length of cyclic prefix (i.e., channel equalization is required), a TEQ is introduced in cascade with the physical channel before the FFT operation.

The $N$-point IFFT generates a complex base-band sig-
nal \( x(n) \):

\[
x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \frac{2 \pi kn}{N}}, \quad n = 0, 1, 2, \ldots N - 1
\]

where \( X(k) \) is the modulated data of the \( N \) sub-carriers. After adding the cyclic prefix, the modulated signal passes through a multi-path fading channel. Using the tap-delay-line model, a channel vector can be described as:

\[
h = [h_0, h_1, \ldots, h_m]^*\]

(2)

In (2), * means conjugate transpose and \( h_i \) is the attenuation of the \( i \)th path. The parameter \( (m+1) \) is the number of paths taken. At the receiver, the received signal \( y(n) \) can be written as:

\[
y(n) = x(n) \otimes h(n) + z(n)
\]

(3)

where \( \otimes \) denotes convolution operation and \( z(n) \) is additive white Gaussian noise (AWGN).

Assume that the TEQ is described by a vector:

\[
\vec{w} = [w_0, w_1, \ldots, w_p]^*\]

(4)

where \( p \) is the order of TEQ. After the insertion of TEQ, the received signal becomes:

\[
s(n) = x(n) \otimes h_{\text{eff}}(n) + z_w(n)
\]

(5)

where the effective channel impulse response is defined as:

\[
h_{\text{eff}}(n) = h(n) \otimes w(n)
\]

(6)

and \( z_w(n) = z(n) \otimes w(n) \). This received signal can be written in the frequency domain as:

\[
S(f) = X(f)H_{\text{eff}}(f) + Z_w(f)
\]

(7)

where \( S(f), X(f), H_{\text{eff}}(f) \) and \( Z_w(f) \) are the FFT of \( s(n), x(n), h_{\text{eff}}(n) \) and \( z_w(n) \), respectively. The effective channel can be modeled by \( \vec{h}_{\text{eff}} \) and the filtered noise \( z_w \). If the effective channel impulse response is limited to be in the range of the guard interval, the received signal will be free from ISI and ICI. Observed from Eq. (7) that a simple zero forcing equalization (ZFE) can be implemented to compensate the effective channel. It is done by multiplying the received signal with \( 1/H_{\text{eff}}(f) \). After FFT, QAM demodulation, de-interleaving, Viterbi decoding, the approximated signal \( \hat{x}(n) \) is obtained.

Equalization technique for OFDM system was originally discussed by Hirosaki [2]. Since then, a number of equalization algorithms, including TEQ algorithms, have been developed [3]–[9]. The main approaches of TEQ design can be formulated as follows. Given a physical multi-path fading channel, described by a FIR filter with coefficient vector \( \vec{h} \), another FIR filter with filter response \( \vec{w} \) is used to cascade with the original channel. The effective channel can now be modeled as an FIR filter with impulse response \( \vec{h}_{\text{eff}} \), known as the target impulse response (TIR) with some delays.

The equalization design begins with TIR and tries to find the optimum coefficients \( \vec{w}_{\text{opt}} \) of the TEQ to shorten the effective channel impulse response to be in the range of the guard interval. Criteria used for finding the coefficients of TEQ and TIR may vary, which lead to different algorithms. Considering the high data rate and no initialization process, the channel equalization for 54 Mbps IEEE 802.11a is one of the most computational intensive tasks. For practical use in high rate OFDM system, it is necessary for the equalization algorithm to have reasonable computational complexity and take a short time to fulfill. The algorithms in [4]–[9] are therefore impractical to be implemented in a high rate WLAN.

3. A Reduced Complexity TEQ Algorithm

In [10], a minimum mean square error (MMSE) method is discussed. Based on this MMSE criterion, an equalization algorithm for a high rate OFDM system with a relatively low complexity is proposed. In what follows, this algorithm is described. More importantly, necessary modifications to the algorithm are also introduced to make the algorithm suitable for practical implementation in a high rate OFDM system. The structure of the algorithm is shown in Fig. 1.

Given the length of TIR and the original channel impulse response \( \vec{h} \), one wants to find the optimum TEQ coefficients to minimize the error signal \( e(n) \). To avoid trivial solution, the MMSE method places a unit-tap constraint on the TIR coefficients \( \vec{h}_{\text{eff}} \).

As mentioned before, the effective channel impulse response \( \vec{h}_{\text{eff}} \) is the convolution of the two FIR filters, namely \( \vec{h}_{\text{eff}} = \vec{h} \otimes \vec{w} \). This filter has the length of \( (m+p+1) \) with the cyclic prefix of \( N_p \) samples long. It is desired that the major components of \( \vec{h}_{\text{eff}} \) are in the range of \( N_p \), while components outside the range of cyclic prefix are very small (i.e., these components should produce little ISI and ICI). Figure 2 illustrates the effective channel impulse response resulted from the convolution between TEQ and the channel impulse response.

Hence, channel state information at the receiver is necessary and many channel estimation algorithms have been developed. Since this paper focuses on the equalization algorithm, an ideal knowledge of the original channel impulse response is assumed. First, a convolution matrix \( H \) of size \( (m+p+1) \times (p+1) \) is generated as follows:

\[
\begin{bmatrix}
\end{bmatrix}

Fig. 1 Structure of the MMSE equalizer.
The optimum solution that minimizes the mean square error by discarding the lines between 2 and \(A\) in hardware. However, this involves matrix inversion, which is difficult. In practice, the energy of samples outside the effective channel impulse response means that the actual values of those components are not important. In practice, the energy of samples outside the effective channel impulse response is set to 1 to avoid trivial solutions. The reduced channel impulse response \(h_{\text{re}}\) refers to the \(k\)th element of the vector \(\vec{h}_{\text{eff}}\) and \(X\) means that the actual values of those components are not important. In practice, the energy of samples outside the range of cyclic prefix should be minimized and the first element of \(\vec{h}_{\text{eff}}\) is set to 1 to avoid trivial solutions. The reduced effective channel impulse response \(\vec{h}_{\text{re-eff}}\) can be written in a simpler form:

\[
\vec{h}_{\text{re-eff}}(k) = \begin{cases} 1 & k = 0 \\ X & 0 < k \leq N_g \\ 0 & N_g < k \leq m + p \end{cases}
\] (9)

Then forms: \(\vec{h}_{\text{reff}} = H \cdot \vec{w}\).

\[
\vec{h}_{\text{reff}}(k) = \begin{cases} 1 & k = 0 \\ X & 0 < k \leq N_g \\ 0 & N_g < k \leq m + p \end{cases}
\] (10)

It then follows that a reduced convolution matrix \(H_{\text{re}}\) with size \((m + p + 1 - N_g) \times (p + 1)\) can be generated accordingly by discarding the lines between 2 and \(N_g + 1\) in \(H\). Thus

\[
\vec{h}_{\text{re-eff}} = H_{\text{re}} \cdot \vec{w}
\] (11)

The optimum solution that minimizes the mean square error is given by [10]:

\[
\vec{w}_{\text{opt}} = \left( H_{\text{re}}^* \cdot H_{\text{re}} + \frac{1}{SNR} I \right)^{-1} \cdot H_{\text{re}}^* \cdot \vec{d}
\] (12)

where \(I\) is a \((p+1) \times (p+1)\) identity matrix. Observe that (12) involves matrix inversion, which is difficult to implement in hardware. However, this difficulty can be overcome as described below. Rewrite (12) as follows:

\[
A \cdot \vec{w} = B
\] (13)

where \(A = H_{\text{re}}^* \cdot H_{\text{re}} + \frac{1}{SNR} I\) and \(B = H_{\text{re}}^* \cdot \vec{d}\). It is easy to see that \(A^* = A\), i.e., \(A\) is symmetrical. This means that it is not necessary to calculate all the elements in \(A\), and there are regularities among the elements of \(A\). Here \(A\) can be shown to be positive definite. Assume that \(\{\vec{x}_k\}\) is the input stream, \(\{\vec{y}_k\}\) is the response of matrix \(H_{\text{re}}\), and the noise \(\{\vec{z}_k\}\).

It can be shown that

\[
E[\vec{y}_k^* \vec{y}_k] = x_k^* \cdot A \cdot x_k + |\vec{z}_k|^2 + x_k^* H_{\text{re}} \vec{z}_k + \vec{z}_k^* H_{\text{re}} \vec{x}_k
\] (14)

For all nonzero \(x_k\), \(E[\vec{y}_k^* \vec{y}_k] > 0\) as noise \(\vec{z}_k\) is far less than the desired signal, so \(x_k^* A \cdot x_k > 0\) and \(A\) is positive definite and Hermitian. Furthermore, a special case of \(LU\) decomposition, the \(LDL^T\) decomposition can be used to factorize \(A\).

It is known that when a matrix is ill conditioned, sometimes the \(LU\) decomposition does not exist or is numerical inaccurate even if it is possible to have the \(L\) and \(U\) matrices. Define the following quantity [11]:

\[
\Omega = \frac{\|S \cdot T^{-1} \cdot S\|_2}{\|A\|_2}
\] (17)

with \(T = (A + A^T)/2, S = (A - A^T)/2\) and \(\|A\|_2\) is the norm-2 of the matrix. It is desired to have the value of \(\Omega\) not very large [11]. It can be observed that this condition is always satisfied whenever \(A\) is symmetric and positive definite. Therefore, the decomposition is always numerical stable and it is safe not to pivot in this case.

\(LDL^T\) decomposition requires only half the computation of \(LU\) decomposition, but \(LDL^T\) decomposition needs one more substitution step to calculate the last result. Therefore, although \(LDL^T\) algorithm is used to factorize \(A\), the \(LU\) decomposition’s forward and backward substitutions are used to calculate the optimum TEQ coefficients. Based on the properties of matrices, there exist a lower triangular matrix \(L\), a diagonal matrix \(D\) and an upper triangular matrix \(U\) that satisfy the following equation:

\[
LDL^T = L(DL^T) = LU = A
\] (18)

where

\[
L = \begin{bmatrix}
I_{11} & 0 & \cdots & 0 \\
I_{21} & I_{22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
I_{N1} & I_{N2} & \cdots & I_{NN}
\end{bmatrix}
\] (19)

\[
U = \begin{bmatrix}
u_{11} & u_{12} & \cdots & u_{1N} \\
0 & u_{22} & \cdots & u_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & u_{NN}
\end{bmatrix}
\] (20)

Fig. 2 A TEQ shortens the impulse response to cyclic prefix.

\[
H = \begin{bmatrix}
h_0 & h_1 & \cdots & h_m & 0 & \cdots & 0 \\
0 & h_0 & h_1 & \cdots & h_m & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & h_0 & h_1 & \cdots & h_m
\end{bmatrix}
\] (8)
and
\[ D = \text{diag}(d_1, \ldots, d_N) \] (21)

From the above expressions, the equations to calculate the elements of \( L, D \) and \( U \) matrices can be written as [12]:
\[
\begin{align*}
    u_{ij} &= a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}; \quad j \geq i \\
    d_i &= u_{ii}; \quad i = 1, \ldots, p + 1 \\
    l_{ij} &= \frac{1}{u_{ij}} \left( a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right); \quad j \leq i - 1
\end{align*}
\] (22)

where \( l_{ii} = 1 \) \((i = 1, \ldots, p + 1)\). The relationship between \( l_{ji} \) and \( u_{ij} \) is:
\[ l_{ji} = \frac{u_{ij}}{u_{ii}}, \quad 1 \leq j < i \leq p + 1 \] (23)

Thus begin from the first row of \( U \), assign \( u_{1j} = a_{1j} \). The first column of \( L \) is found as: \( l_{1j} = u_{1j}/u_{11}, 2 \leq j \leq p + 1 \). All the elements of \( L \) and \( U \) matrices can be found accordingly. After finding the \( L \) and \( U \) matrices, it is ready to solve for the optimum TEQ coefficients through substitutions. For forward substitution, it can be seen that \( B = H_{re}^H \cdot d = [b_1, \ldots, b_{p+1}] \). The equations to solve for \( Y = [y_1, \ldots, y_{p+1}] \) can be written as:
\[
y_1 = \frac{1}{u_{11}} b_1 \]
\[
y_i = \frac{1}{u_{ii}} \left( b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right) = - \sum_{j=1}^{i-1} l_{ij} y_j; \quad i = 2, \ldots, p + 1 \] (24)

In backward substitution, the matrices \( D \) and \( L^T \) are combined to save one step in solving the values of \( \tilde{w}_{opt} \). Similar to the forward substitution, the equation to find the coefficients of \( \tilde{w}_{opt} \) can be written as:
\[
w_{p+1} = \frac{1}{u_{p+1}} y_{p+1} \] (25)
\[
w_i = \frac{1}{u_{ii}} \left( y_i - \sum_{j=i+1}^{p+1} u_{ij} w_j \right); \quad i = p, \ldots, 1 \] (26)

To our best knowledge, only the maximum shortening SNR (MSSNR) algorithm [7] and min-ISI algorithm [19] are implemented in hardware. Since MSSNR algorithm depends only on the channel impulse response and the parameter of TEQ, Appendix compares the complexity of this algorithm and that of the algorithm presented in this paper. As an example, assume that the lengths of the channel impulse response, the TEQ FIR filter and the guard interval are 15, 8 and 8 respectively. Based on the analysis in Appendix, the MSSNR algorithm requires 7620 computations. On the other hand, the proposed algorithm requires only 296 computations. The computation complexity is reduced by 25 times.

4. Simulation Results and Discussions

To show the validity of the proposed algorithm, an OFDM system was set up according to the IEEE 802.11a standard [1]. The data transmission rate is 54 Mbps, including a rate-3/4 punctured convolution code. The inner and outer interleaving patterns are formulated according to the standard. The bit stream is mapped to 64-QAM complex signals. The 64-point IFFT and FFT are used. AWGN is also included in the simulation.

Figure 3 shows the effect of the algorithm with the channel model \( h_1 \) (shown in part-(a)). It is assumed that the channel impulse response lasts 15 samples and the cyclic prefix is 8 samples long. This implies that if TEQ is not included, the ISI/ICI will greatly degrade the system performance. After the pre-FFT TEQ is inserted, the effective channel impulse response is shortened to 9 samples long as shown in Fig. 3(c). All samples beyond cyclic prefix are very small (i.e., producing insignificant ISI/ICI).

Figure 4 illustrates the effect of TEQ on the system performance. The top line shows the bit error rate (BER) of the system without using TEQ. It can be seen that if the maximum channel delay spread exceeds the guard interval...
and if TEQ is not performed and without extension of the guard interval, the system performance cannot be improved by simply increasing SNR. This indicates that the existence of ISI/ICI causes an error floor. Increasing SNR from 12 dB to 22 dB, the bit error rate is only decreased to an order of 10^{-2}. It can also be seen that if a pre-FFT TEQ ($p = 31$) is inserted to shorten the effective channel impulse response to 9 samples and the ZFE is used to compensate the channel effect, the system performance improves significantly. This improvement makes the performance approach the system performance with cyclic prefix equal to 16 samples long and compensated with ZFE method.

The performance using TEQ ($p = 31$) with two other channel modes $\tilde{h}_2$ and $\tilde{h}_3$ are also shown in Fig. 4. The impulse responses of these two channels are as follows. $\tilde{h}_2 = 0.85\delta(k) + 0.402\delta(k - 6) + 0.19\delta(k - 14) + 0.09\delta(k - 20) + 0.042\delta(k - 25)$ and $\tilde{h}_3 = 0.881\delta(k) + 0.416\delta(k - 6) + 0.197\delta(k - 11) + 0.093\delta(k - 15) + 0.044\delta(k - 18) + 0.021\delta(k - 22) + 0.01\delta(k - 30) + 0.005\delta(k - 37)$. The maximum channel delays are 25 and 37 respectively and the path gains follow a decaying exponential distribution. As can be expected, the system performance degrades when the maximum delay spread increases. However the performance with TEQ is still acceptable at high SNR values.

Figure 5 shows the simulation results with different orders of FIR filter to realize the TEQ with $\tilde{h}_1$. It can be observed that even when the TEQ order is 3, the amount of ISI and ICI can be significantly reduced. The gain is very high compared to the case when the channel impulse response is longer than the guard interval and no TEQ is applied. As the order increases to 7, TEQ already achieves the most of its benefits. At the order of 17 or 31, the OFDM system is already at its highest gain. As shown in Fig. 5, there is no significant additional gain when the order exceeds 31.

The optimum TEQ FIR order depends on the channel impulse response and the required BER level. Based on the results tested with all three channel models, for hardware implementation a reasonable order of the FIR filter can be selected to be higher than 7 and less than 17 to achieve the most performance improvement.

All the previous simulation results are obtained assuming perfect channel state information at the receiver. Using the channel estimation method in [10], the influence of the channel estimation error on the system performance is also investigated. This is demonstrated in Fig. 6 for the channel model $\tilde{h}_1$ and a 15-order TEQ. The system performance with an ideal knowledge of the channel impulse response is also provided for comparison. It can be seen that there is a performance loss due to imperfect channel state information, but the loss is minimal and in general it is less than 1 dB compared to the case of having ideal channel state information.

5. Hardware Design Model and Discrete-Time Implementation of the Algorithm

From the discussions in Sect. 3, a block diagram of the hardware design is presented in Fig. 7. First, the fix point matrix multiplication needs to be designed. Recall that $A$ is Hermitian and positive definite. Moreover there are certain regularities existing among the elements of $A$. To calculate the elements of $A$, it is efficient to exploit the properties to further reduce the computational complexity. In this hardware implementation it is also assumed that the physical channel is given by $\tilde{h}_1$ and the order of TEQ is $p = 7$. The matrix $H_{re}$ and $H_{ce}$ can be generated accordingly and the $A$ is a $(p + 1) \times (p + 1)$ matrix. As $A$ is symmetrical, one has

$$a_{ij} = a_{ji} \quad \text{for } i \neq j \quad (27)$$

Thus only the diagonal and the upper (or lower) elements of $A$ need to be calculated. The equation to calculate the diagonal elements can be expressed as follows:

$$\begin{align*}
a_{11} &= h_0^2 + h_3^2 + h_6^2 + h_{10}^2 + h_{13}^2 + h_{16}^2 + h_{19}^2 + h_{22}^2 + h_{25}^2 + h_{28}^2 + h_{31}^2 + h_{34}^2 \\
a_{22} &= h_0^2 + h_3^2 + h_6^2 + h_{10}^2 + h_{13}^2 + h_{16}^2 + h_{19}^2 + h_{22}^2 + h_{25}^2 + h_{28}^2 + h_{31}^2 + h_{34}^2 \\
& \vdots \\
a_{88} &= h_0^2 + h_3^2 + \cdots + h_{28}^2
\end{align*} \quad (28)$$

Fig. 5 Simulation results with different orders of TEQ.

Fig. 6 Bit error rate with estimated and ideal channel.
From the above equation, it can be seen that the result of the last calculation can be exploited to calculate the next element. For $a_{11}$, it can be separated into two parts: $h_0^2 + h_2^2$ and $h_3^2 + h_4^2$. The second part can be used to calculate the value of $a_{12}$ as follows:

$$a_{12} = h_3^2 + h_4^2.$$  

Then $a_{22}$ can be used to calculate $a_{23} = h_5^2 + a_{22}$, and so on. It is observed that this property exists among other elements of matrix $A$. In general this rule can be written as:

$$a_{i,j} = \Delta + a_{i,j}, \quad 1 < i < j \leq 7 \quad i \neq j$$  

$$a_{i,i} = \Delta + a_{i,i}, \quad 1 < i \leq 7$$  

(29)

One needs only one more multiplication to calculate the value of $\Delta$ and one more addition to get the value of the corresponding element in the second row and so on. The values of the elements in other rows can be calculated easily. This property greatly reduces the computational complexity and therefore decreases the requirements of hardware implementation. The registers required to store the intermediate results are also reduced accordingly.

After implementation of the matrix multiplication, LDL$T$ and LU decomposition module is implemented to factorize $A$, and forward and backward substitutions are used to find the optimum coefficients $\tilde{w}_{opt}$. These circuits can be designed easily according to Eqs. (22), (24)–(26). Finally the coefficients of $\tilde{w}_{opt}$ are used to implement the FIR filter (TEQ) accordingly by multipliers and shift registers.

After the pre-FFT TEQ is included in the OFDM receiver, zero forcing equalization is applied to compensate for the effective channel impulse response. This can be done as follows: first FFT transform the vectors of $\tilde{h}$ and $\tilde{w}_{opt}$. Next calculate the inversion of $FFT(\tilde{h}_{opt}) = FFT(\tilde{h}) \times FFT(\tilde{w}_{opt})$. Then in the frequency domain multiply each sub-channel’s output with the individual element of the inversion of $FFT(\tilde{h}_{opt})$. In IEEE 802.11a, the interval of the OFDM symbol is 4.0 $\mu$s. Because of its high data rate and no initialization process, a high performance FFT for equalizer is desired so that the transfer operation can be done in the order of microseconds. A large number of FFT algorithms [13]–[17] have been developed for efficient computation of the discrete Fourier transform (DFT). However not all of these algorithms are suitable for hardware implementation because of their high computational complexity. For FFT, radix-2, radix-4, radix-8 or split radix can be used. In the case of 64-point FFT, the number of complex multiplication is calculated in [18], to be 98, 76 and 80 for radix-2, radix-4 and radix-8 respectively. The radix-4 FFT is also faster than the split radix FFT. In this hardware implementation, the more computationally efficient radix-4 method is preferred.

In hardware implementation of FFT algorithm, pipeline is employed to speed up the calculation. The radix-4 multi-path delay commutator (R4MDC) pipeline FFT is adopted as the fundamental architecture. A block diagram of a 64-point R4MDC pipeline FFT is shown in Fig. 8. This FFT structure consists of $\log_4^N = \log_4^{24} = 3$ stages. Each stage includes:

1. A delay commutator, to rearrange the input data sequences as required by the algorithm.
2. An arithmetic element (AE) which performs the arithmetic functions and butterfly.
The parallel parameters, $\tilde{w}_{opt}$ and $\tilde{h}$ are converted to a serial input. In the first stage, the delay commutator distributes the one path input to four path sub-sequences. The first path does not have a delay, the delays for the second, third and fourth path are $4^{n-i-1}$, $2 \times 4^{n-i-1}$ and $3 \times 4^{n-i-1}$ respectively, with $i$ is the stage number and $n = \log_4 N$. After the delay commutator, the sub-sequences are passed to the first stage arithmetic element. At the same time, the twiddle factors are feed into the arithmetic element in correct order. The other stages are almost the same as the first stage. The FFT result is available at the end of stage 3, but it is in digital inverse order, a shuffler is needed to reorder the output.

6. Information on Hardware Implementation, Simulation Results and Discussions

Detailed information on hardware implementation is listed in the Tables 1, 2 and 3. In Table 1, it can be seen that it needs about 3.9 $\mu$s to calculate the $\tilde{w}_{opt}$; this time is less than one OFDM symbol in IEEE 802.11a. The TEQ FIR filter and pipeline FFT can run at 20 MHz clock and satisfies the system requirement to transmit data up to 54 Mbps in IEEE 802.11a system.

In hardware implementation, since the number is presented by finite binary bits, some error will be introduced through the truncation or rounding. Here it assumes that the data width is 16-bit width, 3-bit integer, 13-bit fraction. Therefore the least significant bit (LSB) of the calculation is $2^{-13}$. If the actual value is not a multiplication of LSB, there are some errors in the fix point presentation. It is well known that if 16 bits $\times$ 16 bits, the result is 32 bits, and truncation is required to shorten the result. This also introduces some errors. The expectation and variance of error generated by binary truncation can be calculated by the following equations:

$$\mu = \frac{2^N - 1}{2^{N+1}} \text{LSB} = \frac{2^{16} - 1}{2^{17}} 2^{-13} \approx 2^{-14} \ \ (30)$$

and

$$\sigma^2 = \frac{(2^N - 1)(2^{N+1} - 1)}{6 \cdot 2^{2N}} \left( \frac{(2^N - 1)^2}{4 \cdot 2^{2N}} \right) \text{LSB}^2 \ \ \ (31)$$

The simulation results with $\tilde{h}_1$ in the corresponding decimal values are listed in Table 3. The mathematic calculation is also shown for comparison. Observe that there are differences between the TEQ filter coefficients $\tilde{w}_{opt}$ obtained from hardware implementation and mathematical calculation. But they are insignificant as shown in Table 3. The inversion of $FFT(\tilde{h}_{eff})$ is calculated in hardware. The difference between the hardware implementation and the mathematical calculation is also plotted in Fig. 9. It can be seen that the difference is not always stable, some sub-channels have larger offset, some have smaller. However all are tolerable.

System performance was also tested. The vectors $\tilde{w}_{opt}$, $\tilde{h}_1$ and the inversion of $FFT(\tilde{h}_{eff})$ calculated in hardware were used as parameters to test the system performance.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Hardware simulation and mathematical calculation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEQ</td>
<td>Decimal Values</td>
</tr>
<tr>
<td>$w_0$</td>
<td>1.89099121103750</td>
</tr>
<tr>
<td>$w_1$</td>
<td>-0.53015136718500</td>
</tr>
<tr>
<td>$w_2$</td>
<td>-0.80749517178500</td>
</tr>
<tr>
<td>$w_3$</td>
<td>-0.25217849768500</td>
</tr>
<tr>
<td>$w_4$</td>
<td>0.74645996093750</td>
</tr>
<tr>
<td>$w_5$</td>
<td>0.18432617187500</td>
</tr>
<tr>
<td>$w_6$</td>
<td>-0.12048339843750</td>
</tr>
<tr>
<td>$w_7$</td>
<td>-0.38659667968500</td>
</tr>
</tbody>
</table>

![Fig 9](image-url)
The result is shown in Fig. 10. The bit error rate obtained with mathematical calculation for the algorithm’s parameters is also shown for comparison. It can be seen that due to the finite length of binary representation; the error introduced somewhat degrades the system performance, but the degradation is very minimal. This hardware implementation result verifies the proposed equalization technique.

7. Conclusions

In this paper a computational efficient design of a TEQ for high rate IEEE 802.11a application to reduce the ISI and ICI has been proposed and implemented in FPGA. Simulation results are provided to demonstrate the validity of the technique. Performance analysis shows that the high dB gain can be obtained by using only a moderate order of FIR filter in the equalizer. Characteristics of the matrices are exploited to reduce computational intensity and hardware complexity. The optimum TEQ coefficients can be found in less than 4 $\mu$s. A high performance R4MDC pipeline FFT is implemented to perform zero forcing equalization in frequency domain. The equalizer implemented in FPGA satisfies the OFDM system requirements to operate at a data rate of 54-Mbps. The simulation results and performance analysis verify the functionalities of the chips and the performance loss is minimal compared with the mathematical calculation results. The proposed technique is attractive when it is necessary to shorten the long delay communication channels to increase the transmission efficiency. The simplicity of the technique makes its implementation possible in FPGA for high rate IEEE 802.11a systems.

Acknowledgements

The authors would like to thank the reviewers for their helpful comments, which have improved the presentation of this paper.

References


Appendix: Complexity Analysis

This Appendix gives the complexity analysis for the equalization algorithms employed in this paper and the one presented in [7]. In [7], after the insertion of the TEQ, the effective channel impulse response $\hat{h}_{\text{eff}}$ is represented by the equivalent impulse response composed of $\hat{h}_{\text{win}}$ and $\hat{h}_{\text{aft}}$. The part $\hat{h}_{\text{win}}$ represents the desired signal, a window of $(N_g + 1)$ consecutive samples from $\hat{h}_{\text{eff}}$. The part $\hat{h}_{\text{aft}}$ represents the
signal falling out of the desired window, which contributes to the ISI. Assume that the guard interval is $N_g$, the lengths of channel impulse response and TEQ FIR filter are $m$ and $p$, respectively. $H_{win}$ and $H_{wall}$ are the convolution matrices generated accordingly with dimensions of $(N_g + 1) \times p$ and $(m + p - N_g - 2) \times p$. With setting the delay to a specific value, the optimum TEQ coefficient vector $\bar{a}$ can be calculated as follows:

1. Calculate the matrices $A = H_{wall}^T H_{wall}$ and $B = H_{win}^T H_{win}$ first with a constraint of $\bar{w}^T B \bar{w} = 1$.
2. Calculate the square root matrix of $B$ and matrix $C$, as shown below: $B = \sqrt{B} \sqrt{B}^T$, $C = (\sqrt{B})^{-1} A (\sqrt{B})^{-1}$.
3. Solve for the optimum coefficients $\bar{w}_{opt}^T = (\sqrt{B})^{-1} \bar{l}_{min}$, where $\bar{l}_{min}$ is the unit-length eigenvector corresponding to the minimum eigenvalue $\lambda_{min}$ of matrix $C$.

In order to find the optimum value, the above steps are iterated setting different values of the delay. Since the algorithm in the presented paper does not involve iterative steps, for a fair comparison, the complexity of the MSSNR algorithm is determined without any iteration. This is calculated as follows (The calculation does not include additions and subtractions):

1. The calculation of $A$ and $B$ requires $p^2 (N_g + 1)$ and multiplications $p^2 (m + p - N_g - 2)$.
2. The square root matrix of $B$ can be calculated by Cholesky decomposition. The computation complexity of the decomposition is about $\frac{p^2 + 1 + 2p + 1}{12} p^3$. Substitutions require $p (p + 1)$ computations. Without considering the pivoting, it requires $\frac{p^2 + 1 + 2p + 1}{12} - \frac{p (p + 1)}{4} + 1$ operations. The calculation of $C$ requires the inversion of $\sqrt{B}$ which also can be calculated by Cholesky decomposition. Then it requires $p^4$ more computations. The overall computation amounts to $p^4 + \frac{p^2 (p + 1)^2 (2p + 1)}{6} - \frac{3p (p + 1)}{4}$ for the second step.
3. As it is required to calculate the minimum eigenvalue and the corresponding minimum eigenvector. From the popular algorithms, assume the inverse power method is chosen. The computation complexity of the inverse power method is determined as follows. The first step requires $\frac{p (p + 1)(2p + 1)}{6}$ operations. The later step requires $p^2$ operations. If it is assumed that there are 24 iterations, then combining with the calculation of $\bar{w}_{opt}$, the complexity would be $\frac{p (p + 1)(2p + 1)}{6} + 25p^2$ computations.

The computation complexity of MSSNR algorithm can be approximated by the following equation:

$$O(\text{MSSNR algorithm}) = p^2 (m + p - 1) + \frac{3p (p + 1)}{2} + \frac{p (p + 1)(2p + 1)}{3} + 25p^2 + p^4.$$  \hspace{1cm} (A-1)

The computational complexity of the proposed algorithm is calculated as follows. The dimension of the convolution matrix is reduced by $N_g$, the number of multiplication is $p (m + p - N_g - 2) + \frac{p (p + 1)}{2}$ (exploit the regularity of the elements in the matrix). The $LDLT$ decomposition and substitutions require about $\frac{p (p + 1)(2p + 1)}{12} + \frac{3p (p + 1)}{4}$. The total computation amounts to:

$$O(\text{proposed algorithm}) = p (m + p - N_g - 2) + \frac{p (p + 1)(2p + 1)}{12} + \frac{5p (p + 1)}{4}.$$  \hspace{1cm} (A-2)

Based on the above analysis, an example to illustrate the computation advantage of the algorithm used in the paper over the MSSNR algorithm was provided at the end of Sect. 3.