FPGA Implementation of Blind Adaptive Decision Feedback Equalizer

Doaa Ashmawy(1) Esam Abdel-Raheem(2) Hala Mansour(1) Mohamed Youssif(3) Mahmoud Mohanna(4)
(1) ECE Dept., Benha University, Cairo, Egypt
(2) ECE Dept., University of Windsor, Ontario, Canada
(3) ECE Dept., Al-Azhar University, Cairo, Egypt
(4) Seismology Dept., NRIAG, Cairo, Egypt

Abstract—This paper considers field programmable gate array (FPGA) implementations for blind adaptive decision feedback equalizer (DFE) based on the IP core reported in [1]. The design can achieve channel equalization for 16-QAM and 64-QAM. Constant modulus algorithm (CMA) and multi-modulus algorithm (MMA) are considered for update the coefficients in the blind mode of operation which are followed by decision-directed (DD) mode. The system can work at a maximum clock frequency of 22 MHz. The design steps first consider fixed-point simulations using MATLAB fixed-point toolbox follows by FPGA implementations. The implementations are divided into complex weight update module, output computation module, error adjustment module, and decision device module. Finally, the DFE is implemented using Xilinx Virtex-II XC2VP100 FPGA.

I. INTRODUCTION

DFEs are commonly used in digital communication systems to reduce the effects of intersymbol interference (ISI) introduced by time-dispersive channels. ISI distortion arises when the instantaneous value of the received signal depends on several transmitted symbols. DFE uses a weighted sum of previous symbol decisions to form an estimate of the ISI due to previously transmitted symbols. This estimate is then subtracted from the signal at the input to the decision device to reduce the effect of ISI. However, decision errors result in residual ISI which may increase the number of incorrect decisions about subsequent symbols. This may lead to error propagation in DFE [2]-[5]. The DFE error propagation is not catastrophic, on typical channels errors it occurs in short bursts that degrade performance slightly. Unlike the linear equalizers that attempt to invert the distortive channel, the DFE uses the past decisions to cancel the ISI without noise enhancement.

FPGA platform maintains reconfigurability and flexibility which are key requirements for communication [6]. This paper presents FPGA implementation of a 36-tap adaptive DFE for QAM demodulators. The design can achieve channel equalization for 16-QAM and 64-QAM signals. The implementation is targeting Xilinx Virtex-II XC2VP100 FPGA. The remainder of this paper is organized as follows: section II describes the system model, while section III introduces blind equalization algorithms. The design of a 36-tap DFE for QAM signals is presented in section IV, while FPGA simulation and implementation results are given in section V. Some concluding remarks are given in section VI.

II. CHANNEL AND EQUALIZER MODEL

A multirate model [7] of the channel and equalizer is illustrated in Fig. 1, where the index ‘n’ denotes T-spaced quantitie, while ‘k’ denotes T/2-spaced quantities with T denotes the symbol period. A T-spaced source symbol s(n) is transmitted through a pulse-shaping filter and modulated onto a T/2-spaced propagation channel, whose impulse response is given by h = [h_0, h_1, ..., h_{Nc-1}]^T, where (^T) is the transpose operator. The channel impulse response represents the combined effect of the transmit filter, channel impulse response, and the anti-alias filter at the receiver front end. The data sequence \{s_m\} consists of complex, zero mean random variable that is independent and identically distributed (i.i.d.) and is drawn from the finite set \{s_{m,R} + j s_{m,I}\}_{m=1}^{M} for an M-QAM constellation, where the subscripts R and I denote the magnitude of the real and imaginary quantities, respectively. The received T/2-spaced input signal x(t) is corrupted by ISI and the additive white Gaussian noise (AWGN) signal w(t) and is defined as

\[ x(t) = \sum_{m} s_m (t - mT) + w(t) \]  \hspace{1cm} (1)

The distortion introduced by the channel is removed by the DFE. As shown in Fig. 1, the DFE consists of two filters, a fractionally-spaced (FS) feedforward equalizer (FFE) that has L_{FFE} taps and a symbol-spaced feedback equalizer (FBE) that has L_{FBE} taps. The FFE and FBE tap-coefficients are represented by weights \( C = [c_0, c_1, ..., c_{L_{FFE}-1}]^T \) and \( B = [b_0, b_1, ..., b_{L_{FBE}-1}]^T \), respectively. The symbol estimate, \( \hat{z}(n) = \hat{z}_R(n) + j \hat{z}_I(n) \), is generated from z(n) using a decision-device and is decoded into the received binary sequence. An adaptive algorithm is applied to iteratively adjust the FFE and FBE tap coefficients. The channel, FFE, and FBE are modeled as complex finite-impulse response (FIR) filters.

![Figure 1: Baseband model of DFE system.](image-url)
III.  BLIND DECISION FEEDBACK EQUALIZERS
ALGORITHMS

The stochastic gradient algorithms adjust the equalizer tap weights in the direction of the negative gradient. The FFE and FBE coefficients are adjusted according to the following equations [8]

\[
C(n+1) = C(n) + \mu (V_{I,FFE}) ,
\]

\[
B(n+1) = B(n) + \mu (V_{I,FBE})
\]

where \(V_I\) denotes stochastic gradient of the FFE cost function \(J_{FFE}\) with respect to the tap weights vector \(c(k)\) and \(V_b\) denotes stochastic gradient of the FBE cost function \(J_{FBE}\) with respect to the tap weights vector \(b(n)\) while \(\mu\) is the step-size parameter. The output of the decision device is a sequence of estimates. These estimates are fed back to the FBE. The DFE output \(z(n)\) is the sum of the FFE output, \(y(n)\) and the negative of the output of the FBE, \(p(n)\) and is given by

\[
z(n) = y(n) - p(n)
\]

where

\[
y(n) = \sum_{i=0}^{L_{FFE}-1} c_i(n)x(n-i) = X^T(n)C(n),
\]

\[
p(n) = \sum_{i=0}^{L_{FBE}-1} b_i(n)z(n-i) = \hat{Z}^T(n)B(n),
\]

where \(X(n) = [x(n), x(n-1), ..., x(n-L_{FFE} + 1)]^T\) is the regressor vector (i.e., values of the delay line) for FFE and \(\hat{Z}(n) = [\hat{z}(n), \hat{z}(n-1), ..., \hat{z}(n-L_{FBE} + 1)]^T\) is regressor vector for FBE.

A. Constant Modulus Algorithm (CMA)

The CMA [9, 10] is a popular scheme for blind equalization of QAM systems. The CMA cost function has the form

\[
J_{CMA}(n) = \frac{1}{2p} E[(|z(n)|^p - \gamma_{CMA,p})^2],
\]

For the special case of \(p = 2\), the CMA cost function has the form

\[
J_{CMA}(n) = \frac{1}{4} E[(|z(n)|^2 - \gamma_{CMA})^2],
\]

where \(E[\cdot]\) denotes statistical expectation and \(\gamma_{CMA}\) is a positive real constant defined by

\[
\gamma_{CMA} = \frac{E[|s(n)|^4]}{E[|s(n)|^2]^2}.
\]

The FFE coefficients are updated according to the relation

\[
C(n+1) = C(n) - \mu e_{CMA}(n)X^*(n),
\]

\[
e_{CMA}(n) = z(n)(|z(n)|^2 - \gamma_{CMA})
\]

\[
= z_R(n)(z_R^2 + z_I^2 - \gamma_{CMA})
\]

\[
+ j z_I(n)(z_R^2 + z_I^2 - \gamma_{CMA})
\]

where \(e_{CMA}\) is the CMA error signal and the asterisk (*) denotes complex conjugation, \(z(n) = z_R(n) + j z_I(n)\).

The CMA error signal defined by equation (10) is realized in Fig. 2(a).

B. Multimodulus Algorithm

The multimodulus algorithm (MMA) introduced by Yang et al. [11], has low steady-state mean-squared error (MSE) and eliminates the need for phase recovery. By modifying the CMA cost function in the form of cost functions for real and imaginary parts, the MMA cost function can be written as

\[
J_{MMA}(n) = J_{MMA,R}(n) + J_{MMA,I}(n),
\]

where

\[
J_{MMA,R}(n) = \frac{1}{4} E[|z_R(n)|^2 - \gamma_{MMA,R}],
\]

\[
J_{MMA,I}(n) = \frac{1}{4} E[|z_I(n)|^2 - \gamma_{MMA,I}],
\]

Assuming that the input data sequence \(s(n) = s_R(n) + j s_I(n)\) is i.i.d., the constants \(\gamma_{MMA,R}\) and \(\gamma_{MMA,I}\) can be defined as

\[
\gamma_{MMA,R} = \frac{E[|s_R(n)|^4]}{E[|s_R(n)|^2]^2}, \quad \gamma_{MMA,I} = \frac{E[|s_I(n)|^4]}{E[|s_I(n)|^2]^2}.
\]

The FFE coefficients are updated according to the relation

\[
C(n+1) = C(n) - \mu e_{MMA}(n)X^*(n),
\]

where the MMA error signal \(e_{MMA}(n) = e_{MMA,R} + j e_{MMA,I}\) is given by

\[
e_{MMA,R}(n) = z_R(n)(|z_R(n)|^2 - \gamma_{MMA,R}),
\]

\[
e_{MMA,I}(n) = z_I(n)(|z_I(n)|^2 - \gamma_{MMA,I}),
\]

and is realized as in Fig. 2(b).

C. Decision directed (DD) algorithm

The cost function of the DD algorithm [12] is described by

\[
J_{DD}(n) = \frac{1}{2} E[(z(n) - \hat{z}(n))^2],
\]

The FFE and FBE coefficients are updated according to the relations

\[
C(n+1) = C(n) - \mu e_{DD}(n)X^*(n),
\]

\[
B(n+1) = B(n) + \mu e_{DD}Z^*(n),
\]

where \(e_{DD} = e_{DD,R} + j e_{DD,I}\) is the DD error and is given by

\[
e_{DD,R}(n) = z_R(n) - \hat{z}_R(n),
\]

\[
e_{DD,I}(n) = z_I(n) - \hat{z}_I(n),
\]

The DD error signal defined in (19) is realized directly as shown in Fig. 2(c).

IV. DESIGN OF DFE FOR QAM SIGNALS

This section discusses the design of a 36-tap DFE for QAM signals. The DFE is composed of 16-tap FFE and 16-tap FBE. In order to avoid error propagation problem, the FBE will be initialized by a unitary double center spike and adapted blindly using CMA or MMA while the FBE will be fixed at zero. \(e_{DD}\) represents the minimum MSE required to transfer to DD.
CMA error signal.

MMA error signal.

DD error signal.

Figure 2: Error adjustment module output.

\( \varepsilon_{DD} \) is approximately 11.19 dB for 16-QAM and 17.40 dB for 64-QAM [13]. Once the MSE is below \( \varepsilon_{DD} \), the adaptation of FBE using DD algorithm starts and the DFE is switched to the DD mode.

A. Fixed-Point Analysis

The word-length (WL) and fractional word-length (FWL) for the target algorithms are determined by simulations using MATLAB fixed-point toolbox. The data, processed by FPGA, such as the input signals, the coefficients of filter, or the filter output may be positive or negative. Accordingly, two’s complement arithmetic has been adopted. Saturation is used to handle overflow conditions, while truncation is applied instead of rounding [1].

The wordlengths used are shown in table I. The algorithms are simulated for microwave channels from SPIB [14] and converted to fixed-point using the MATLAB fixed-point toolbox. Fixed-point simulations results for SPIB cable channel #4 are illustrated in Fig. 3 and Fig. 4 for 16-QAM and 64-QAM, respectively. For 16-QAM, a step size \( \mu = 2^{-9} \) was used for both the FFE and FBE. The same step size was used for 64-QAM. The MSE curves were obtained by averaging the instantaneous squared-error across the decision device over 200 realizations. For 16-QAM, the fixed-point and floating-point curves are nearly identical for CMA and MMA. For 64-QAM, the convergence time, which is defined as the number of samples necessary to reach 90% of the steady-state MSE, remains the same for fixed-point CMA, while the convergence time increases for fixed-point MMA. For the steady-state MSE, the fixed-point CMA and MMA steady-state MSE are higher than their floating-point counterparts.

Figure 3: Fixed-point simulation results for 16-QAM.
Figure 4: Fixed-point simulation results for 64-QAM.

<table>
<thead>
<tr>
<th>Signal(s)</th>
<th>Wordlength (Total, Fractional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x(n), e(n), y(n)</td>
<td>(16,14)</td>
</tr>
<tr>
<td>p(n), z(n)</td>
<td>(16,14)</td>
</tr>
<tr>
<td>w(n)</td>
<td>(20,18)</td>
</tr>
</tbody>
</table>

### TABLE I. SIGNAL WORDLENGTHS.

#### B. FPGA Implementation of DFE

The DFE implemented is a complex 36-tap DFE. The equalizer output data and tap coefficients are updated once per symbol period. A block diagram of the DFE is illustrated in Fig. 5, where the input signal is $x(k)$, while the output signal is $z(n)$. The whole system is divided into four main modules: complex weight updates module, output computation module, error adjustment module, and decision-device module.

1) **Complex weight updates module**

The complex weight updates module stores the current value of the equalizer coefficient and calculates the next coefficient value using the output error from the error adjustment module, the step-size, and the input data vector. This is realized as shown in Fig. 6(a) for the FFE, according to the following relation [1]

$$
C(n + 1) = C(n) - \mu e(n)X^*(n) \\
= C(n) - \mu e_R(n)X_R(n) - \mu e_I(n)X_I(n) \\
- j\left(\mu e_I(n)X_R(n) - \mu e_R(n)X_I(n)\right),
$$

(20)

where $e(n) = e_{\text{CMA}}(n)$, $e_{\text{MMA}}(n)$, or $e_{\text{DD}}(n)$.

For the FBE, the coefficients are updated as shown in Fig. 6(b) according to the relation

$$
B(n + 1) = B(n) + \mu e_{\text{DD}}(n)\hat{Z}^*(n) \\
= B(n) + \mu e_R(n)\hat{Z}_R(n) + \mu e_I(n)\hat{Z}_I(n) \\
+ j\left(\mu e_I(n)\hat{Z}_R(n) - \mu e_R(n)\hat{Z}_I(n)\right)
$$

(21)

where $e(n) = e_{\text{DD}}(n)$.

![Figure 5: Direct form DFE.](image)

---

978-1-4244-5950-6/09/$26.00 ©2009 IEEE 498
the QAM constellation into four quadrants based on the sign bit of $z_R$ and $z_I$ signals and then biasing the center of the selected quadrant to the center of the signal constellation. This is repeated until the constellation is 4-QAM. By determining the final quadrant, the decision device output is estimated.

V. FPGA SIMULATION AND IMPLEMENTATION RESULTS

The XC2VP100 FPGA chip of Virtex-II family from Xilinx corporation is selected, and the VHDL code is synthesized and simulated on Xilinx Integrated Software Environment (ISE).

A. RTL simulation

A module was tested as in [1] by creating a set of fixed-point inputs using the fixed-point toolbox in Matlab. The data is then saved to a stimulus file and applied by Xilinx ISE testbench. The simulation output was written to a file in Matlab format to plot a graphical output. MSE curves for the CMA-DD and MMA_DD algorithms are illustrated in Fig. 8(a) and (b) for 16-QAM and 64-QAM, respectively. The MSE curves shown in Fig. 9(a) and (b) are for the signed error CMA-DD and the signed error MMA-DD. These curves were obtained by averaging the MSE across the decision device for a single realization.
The design is divided into four modules: the complex weight update module, the output computation module, the error adjustment module, and the decision device module. The circuit is synthesized and simulated on Xilinx ISE software platform for Xilinx Virtex-II XC2VP100 FPGA. This circuit can be used in the design of QAM demodulators for channels with severe ISI and has achieved a maximum clock frequency of 22MHz.

VI. CONCLUSION

A 36-tap DFE complex equalizer has been implemented for QAM signals. It can implement CMA, MMA, or DD error signals, while it can achieve channel equalization for 16-QAM and 64-QAM signals. A signed version of the circuit was also implemented to reduce the computation overhead. The design is

ACKNOWLEDGMENT

The second author acknowledges the support provided by Canadian Microelectronics Corporation (CMC). The valuable assistance offered by Kevin Banovic is also greatly acknowledged.

REFERENCES