

# A Parallel Low-Complexity Coefficient Computation Processor for the MMSE-DFE

Naofal Al-Dhahir Communications Program GE Corporate R & D Center Niskayuna, NY 12309 Email : aldhahir@birch.crd.ge.com	Ali H. Sayed * Electrical Engineering Dept. University of California Los Angeles, CA 90095 Email : sayed@ee.ucla.edu
---	--

## Abstract

A modular parallel architecture for a MMSE-DFE coefficient computation processor is presented. The architecture is based on the **QR** factorization of a channel-and-noise-dependent data matrix and is implemented using CORDIC processors within a systolic array architecture. Implementation issues including the number of CORDIC stages and the bit precision required in a fixed-point implementation are investigated through computer simulations.

## 1 Introduction

On severe-ISI channels, such as those encountered in GSM-based digital cellular systems and the emerging broadband wireless services, the mean-square-error decision feedback equalizer (MMSE-DFE) has been demonstrated to be a high-performance receiver structure [1, 2]. Furthermore, it has a much lower complexity than MLSE receivers (whose complexity grows exponentially with the channel delay spread). Also, a channel-estimate-based MMSE-DFE has been shown to offer undeniable performance advantages over an adaptive LMS-based MMSE-DFE that suffers from slow convergence on severe-ISI channels (due to the large eigenvalue spread problem) [3].

The main factor limiting the wide-spread use of channel-estimate-based MMSE-DFE's in practical systems has been the high-complexity in computing the optimal MMSE-DFE coefficients from the CIR coefficients in real-time which results in unacceptable power consumption, delay, and storage requirements for small-size, light-weight, and cost-effective wireless receivers.

In this paper, we present a low-complexity, numerically-stable, and parallelizable algorithm for computing the optimum coefficients of the MMSE-DFE from the CIR coefficients and the noise auto-correlation sequence.

\*The work of A. H. Sayed was supported in part by NSF award MIP-9796147.

## 2 QR-Based MMSE-DFE Coefficient Computation

We adopt the standard discrete-time representation of a linear additive-noise intersymbol interference (ISI) channel given by

$$\mathbf{y}_k = \sum_{m=0}^{\nu} \mathbf{h}_m x_{k-m} + \mathbf{n}_k, \quad (1)$$

where  $\mathbf{h}_m \stackrel{def}{=} [h_{l-1,m} \cdots h_{0,m}]^t$ , the notation  $(\cdot)^t$  denotes the transpose, is the  $m^{th}$  (vector) channel coefficient, assuming an oversampling factor of  $l$ , and  $\nu$  is called the channel memory. Both the input sequence,  $\{x_k\}$ , and the noise sequence,  $\{\mathbf{n}_k\}$ , are assumed to be complex, zero-mean, and have (non-singular) auto-correlation matrices denoted by  $\mathbf{R}_{xx}$  and  $\mathbf{R}_{nn}$ , respectively.

### 2.1 The FIR MMSE-DFE

The MMSE-DFE consists of a fractionally spaced feedforward filter that has  $(lN)$  taps and is denoted by  $\mathbf{w}^*$ , and a symbol-spaced feedback filter that consists of  $N_b (\leq \nu)$  taps denoted by  $\{-b_1, -b_2, \dots, -b_{N_b}\}$ . For analytical convenience, we define the augmented vector  $\tilde{\mathbf{b}}^* \stackrel{def}{=} [0_{1 \times \Delta} \ 1 \ b_1^* \ \cdots \ b_{N_b}^*]$ , where  $\Delta (0 \leq \Delta \leq N + \nu - 1)$  is the delay of the feedforward filter, and  $(\cdot)^*$  denotes the conjugate transpose. Consider the Cholesky factorization [4]

$$\mathbf{G} \stackrel{def}{=} \mathbf{R}_{xx}^{-1} + \mathbf{H}^* \mathbf{R}_{nn}^{-1} \mathbf{H} \stackrel{def}{=} \mathbf{R}^* \mathbf{R}, \quad (2)$$

where  $\mathbf{H}$  is the channel matrix and  $\mathbf{R}$  is an upper-triangular matrix. It was shown in [5] that the optimum FIR MMSE-DFE settings are given by

$$\tilde{\mathbf{b}}_{opt} = \frac{\mathbf{R}^* \mathbf{e}_{(\Delta_{opt} + 1)}}{\mathbf{R}(\Delta_{opt} + 1, \Delta_{opt} + 1)} \quad (3)$$

$$\mathbf{w}_{opt}^* = \frac{\mathbf{e}_{(\Delta_{opt} + 1)}^* \mathbf{R}^{-*} \mathbf{H}^* \mathbf{R}_{nn}^{-1}}{\mathbf{R}(\Delta_{opt} + 1, \Delta_{opt} + 1)}, \quad (4)$$

where  $\mathbf{e}_i$  denotes the  $i^{\text{th}}$  unit column vector and  $\mathbf{R}(i, j)$  denotes the  $(i, j)$  element of  $\mathbf{R}$ . The unbiased decision-point SNR of the MMSE-DFE is given by

$$SNR_{MMSE-DFE,U} = |\mathbf{R}(\Delta_{opt.}+1, \Delta_{opt.}+1)|^2 - 1. \quad (5)$$

The optimum delay,  $\Delta_{opt.}$ , that maximizes  $SNR_{MMSE-DFE,U}$  is given by

$$\Delta_{opt.} = (\arg\max_{1 \leq i \leq N+\nu} \{\mathbf{R}(i, i)\}) - 1. \quad (6)$$

## 2.2 QR Factorization

To make the Cholesky factorization parallelizable, we perform a **QR** factorization [4] of the following augmented channel matrix

$$\tilde{\mathbf{H}} \stackrel{def}{=} \begin{bmatrix} \mathbf{R}_{xx}^{\frac{\nu}{2}} \\ \mathbf{R}_{n\tilde{n}}^{\frac{\nu}{2}} \mathbf{H} \end{bmatrix} = \mathbf{Q}\mathbf{R}, \quad (7)$$

which implies that  $\mathbf{G} = \tilde{\mathbf{H}}^* \tilde{\mathbf{H}} = \mathbf{R}^* \mathbf{R}$ . The notation  $\mathbf{R}_{xx}^{\frac{\nu}{2}}$  denotes the upper-triangular square root of the positive definite matrix  $\mathbf{R}_{xx}^{-1}$ . The following remarks are in order

- For the case of white input and noise, the matrix  $\tilde{\mathbf{H}} = \begin{bmatrix} \frac{1}{\sqrt{S_x}} \mathbf{I}_{N+\nu} \\ \frac{1}{\sqrt{N_0}} \mathbf{H} \end{bmatrix}$ , where  $S_x$  and  $N_0$  are the input energy and noise power spectral density per complex dimension, respectively. We focus on this case for the remainder of this paper.
- **QR** factorization is numerically stable and requires less numerical precision than direct matrix inversion of an augmented auto-correlation matrix which is commonly used to compute DFE coefficients [6]. In addition, it is parallelizable which makes it attractive for VLSI implementation.
- For a time-varying channel, the CIR is continuously estimated, using a training sequence embedded in each block, the matrix  $\tilde{\mathbf{H}}$  is updated, and the **QR** factorization is performed to update the MMSE-DFE filter settings.

The orthogonal transformation  $\mathbf{Q}$  that triangularizes  $\tilde{\mathbf{H}}$  can be implemented in a variety of ways, the most popular of which is Givens rotations [4] due to its suitability for parallel implementation.

## 2.3 Effects of Fixed-Point Implementation

The two main reasons for implementing the algorithm on fixed-point rather than floating-point

programmable DSPs are the former's lower power consumption and smaller size both of which are attractive features for wireless applications.

The channel impulse responses assumed in the simulation are snapshots generated according to Jake's famed model for Rayleigh fading dispersive channels [7]. The baseline case is depicted in Figure 1 and is a 2-tap channel with an input SNR of 10 dB and 8 feedforward taps in the DFE. Under these conditions, a 10-bit representation results in less than 0.1 dB performance loss. To maintain this same performance level on a 4-tap channel (corresponds to a delay spread of  $\approx 15\mu\text{sec}$  at GSM bit rates) and an input SNR = 30 dB, 16 bits of precision are required as shown in Figure 2.

Next, we examine the delay and computational complexity requirements of the algorithm.

## 2.4 MIPS Estimate

Figure 3 shows a MIPS estimate of the MMSE-DFE coefficient computation algorithm assuming that DFE coefficients are updated every 5 msec. This update interval was chosen to be approximately equal to the GSM frame length of 4.6 msec so that the CIR coefficients are held constant for the duration of the frame. It can be seen from the Figure that the MIPS estimate becomes quite substantial on severe-ISI channels and long DFE filters. As an example, for  $\nu = 4$  (which corresponds to a delay spread of  $\approx 15$  microsec at the GSM bit rate) and  $N = 30$ , the complexity is around 60 MIPS. This estimate is for a *symbol-spaced* FFF (MIPS estimate increases approximately linearly with oversampling factor) and does not include the computation involved in computing the CIR and input SNR and any I/O overhead. More importantly, it does not include the complexity involved in computing the square roots (involved in the Sine and Cosine parameter calculation) which could be significant even if implemented using table look-ups. In spite of that, Figure 4 clearly shows that the time required to compute the DFE coefficients could still be much longer than the coefficient update interval which would cause delays in applying the DFE weights and increases storage requirements. For the GSM system, each 4.6 msec frame is shared by 8 users. Therefore, the processing time at the mobile should be upper-bound by the frame duration to allow for real-time processing; this restricts the number of DFE coefficients that can be used. Based on Figures 3 and 4 it appears that complexity and delay considerations favor an ASIC over a fixed-point DSP implementation for  $\nu \geq 6$  and  $N \geq 15$ . These restrictions become much more stringent at the base station, where the processing time is restricted to be less

than the *slot* time duration (about 0.57 msec) at the base station *if the same processor* is used to equalize data from all 8 users. If this requirement can not be met, then the number of processors needed is determined by the ratio of the frame duration to the processing time of each user. The presence of co-channel interference, which is a performance-limiting impairment in TDMA digital cellular systems, further increases the computational complexity and hence the required time to compute the DFE coefficients. Finally, for many of the emerging broadband wireless services, the channel delay spread could last up to 100 symbol periods [1] resulting in prohibitively long processing delays in computing the DFE coefficients and hence degrading performance appreciably. This effect becomes even more pronounced on rapidly time-varying channels (e.g. high Doppler rates on trains), where the coefficient update interval is reduced to keep up with the channel variations resulting in a higher DFE computational complexity. The aforementioned considerations have motivated us to investigate a parallel implementation to speed up the DFE coefficient computation.

### 3 Parallel Implementation

In this section, we present a *systolic* implementation of the MMSE-DFE coefficient computation processor where the rows of  $\mathbf{R}$  (that determine the feedback filter) are computed in parallel. We implement the orthogonal transformation  $\mathbf{Q}$  as a  $2 \times 2$  Givens rotation matrix whose entries are completely determined by the channel coefficients  $\{\mathbf{h}_i\}_{i=0}^{\nu}$  and the input SNR. For each column in  $\mathbf{H}$ , we *pivot* on the element equal to  $\frac{1}{\sqrt{SNR}}$  and annihilate the elements corresponding to the channel coefficients.

#### 3.1 Systolic Implementation

The systolic implementation is shown in Figure 5 for the case of  $N + \nu = 5$ . There are two types of cells : boundary cells (denoted by a circle) and internal cells (denoted by a rectangle). The number  $r$  represents the value stored in the cell. The boundary cells *compute* the rotation parameters needed to annihilate the lower-triangular portion of the incoming data while the internal cells *apply* these rotation parameters to the cell contents to compute the updated upper-triangular portion of the array.

**Algorithm** (White Input and Noise Case)

**Input :**

CIR coefficients, Input SNR, number of feedforward filter taps.

**Recursions :**

1. Initialize the upper-triangular systolic array to  $\frac{1}{\sqrt{SNR}} \mathbf{I}_{N+\nu}$ .
2. Start feeding the array with the CIR coefficients in a time-skewed manner.
3. After all  $N$  rows of  $\mathbf{H}$  have passed through the systolic array, the array cell contents will be the desired upper-triangular Cholesky factor  $\mathbf{R}$ .
4. Optimum feedback filter coefficients are given by the row of the systolic array that has the *largest* leading element.
5. Calculating the optimum feedforward filter involves solving a triangular system of equations by back substitution. This computation can be performed either on a DSP or by augmenting the triangular systolic array with a linear section as described in [8].
6. Freeze and apply DFE coefficients to buffered received data to equalize it.
7. Recompute CIR and DFE coefficients according to the desired update rate.

Note that the boundary cells require computation of divisions and square roots. While these operations can be performed using table lookups or polynomial approximations, they require a fairly large number of instruction cycles; higher throughput is achieved using a COordinate Rotation DIgital Computer (CORDIC) implementation [9].

#### 3.2 CORDIC Implementation

In any CORDIC implementation, we need to determine the number of CORDIC stages and the bit precision required to achieve acceptable performance for the application of interest. In our simulations, we used  $SNR_{MMSE-DFE,U}$  as a performance measure and we considered a “low dynamic range” scenario with  $\nu = 1$  and input SNR = 10 dB and a “high dynamic range” scenario with  $\nu = 3$  and input SNR = 30 dB. Figures 6 and 7 show that using 9 CORDIC stages results in the same performance as the Givens-rotations-based algorithm of Section 2.3. Assuming 9 stages, it can be seen from Figures 8 and 9 that 16 bits of precision result in less than 0.2 dB performance loss from the floating-point representation for GSM scenarios.

### References

- [1] S. Ariyavisitakul and L.J. Greenstein. ”Reduced-Complexity Equalization Techniques for Broadband Wireless

Channels". *IEEE Journal on Selected Areas in Communications*, pages 5–15, January 1997.

- [2] N. Lo, D.D.Falconer, and A. Sheikh. "Adaptive Equalization and Diversity Combining for Mobile Radio using Interpolated Channel Estimates". *IEEE Transactions on Vehicular Technology*, pages 636–645, August 1991.
- [3] R. Ziegler, N. Al-Dhahir, and J.M.Cioffi. "Nonrecursive Adaptive Decision-Feedback Equalization from Channel Estimates". In *Proceedings of IEEE Vehicular Technology Society 42<sup>nd</sup> VTS Conference*, volume 2, pages 600–603, Denver, May 1992.
- [4] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1989. Second Edition.
- [5] N. Al-Dhahir and J.M.Cioffi. "Efficient Computation of the Delay-Optimized Finite-Length MMSE-DFE". *IEEE Transactions on Signal Processing*, 44(5):1288–1292, May 1996.
- [6] S. Qureshi. "Adaptive Equalization". *Proceedings of the IEEE*, 73(9), September 1985.
- [7] W.C.Jakes. *Microwave Mobile Communications*. IEEE Press, 1974.
- [8] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 1991. 2nd Edition.
- [9] Yu. Hu. "CORDIC-Based VLSI Architectures for Digital Signal Processing". *IEEE Signal Processing Magazine*, pages 16–35, July 1992.

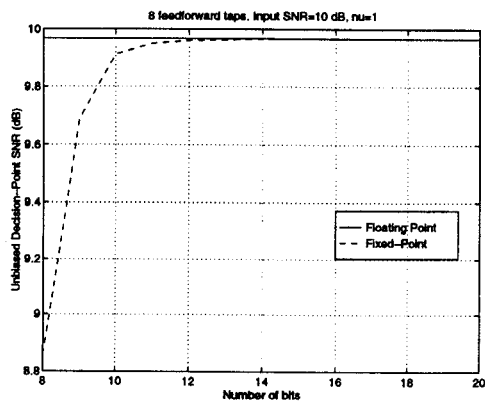


Figure 1: Effect of Fixed-Point Implementation on  $SNR_{MMSE-DFE,U}$  with Givens-Based QR Factorization

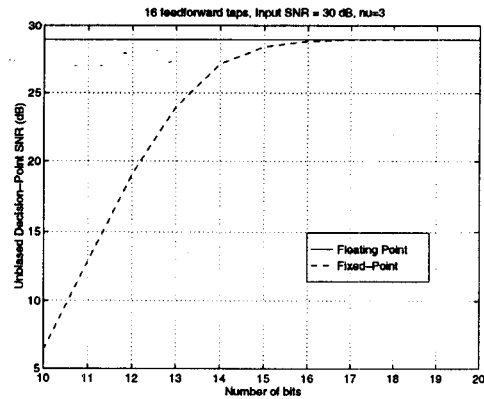


Figure 2: Effect of Fixed-Point Implementation on  $SNR_{MMSE-DFE,U}$  with Givens-Based QR Factorization

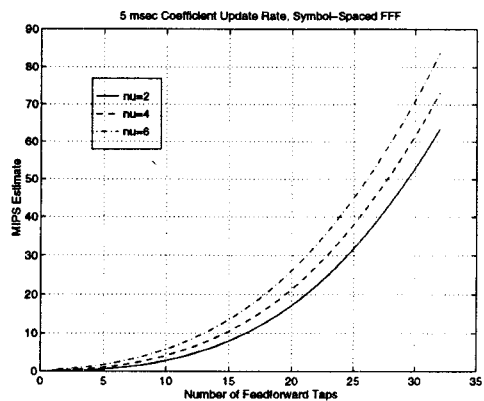


Figure 3: MIPS Estimate for Computing DFE Coefficients with 5 msec Update Rate

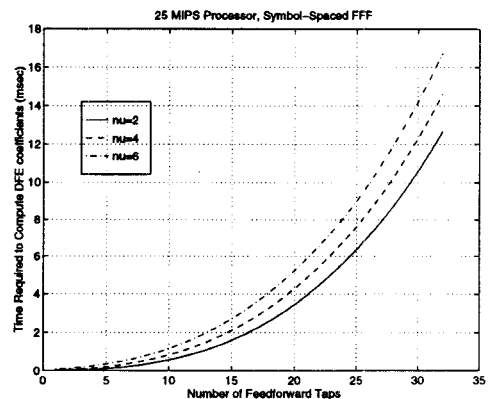


Figure 4: Time Required to Compute DFE Coefficients with 25-MIP Processor

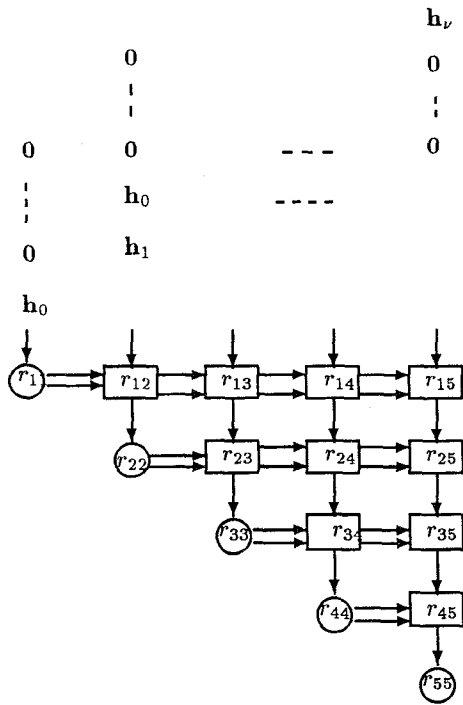


Figure 5: Systolic Array Implementation for QR Factorization of  $\hat{\mathbf{H}}$

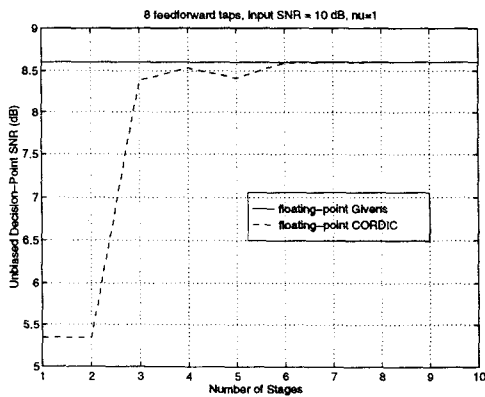


Figure 6: Variation of  $SNR_{MMSE-DFE,U}$  with the Number of CORDIC Stages

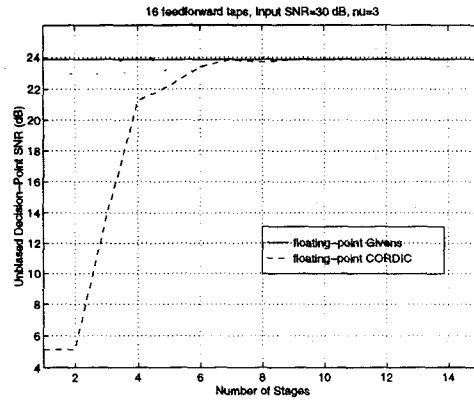


Figure 7: Variation of  $SNR_{MMSE-DFE,U}$  with the Number of CORDIC Stages

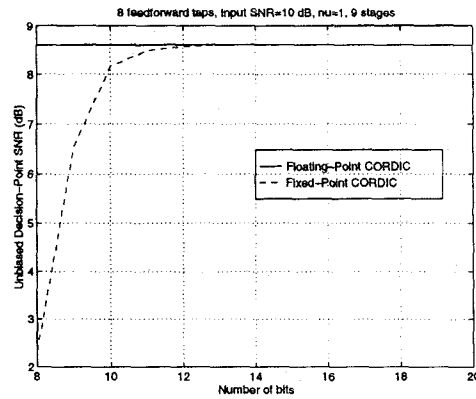


Figure 8: Effect of Fixed-Point Implementation on  $SNR_{MMSE-DFE,U}$  with CORDIC-Based QR Factorization

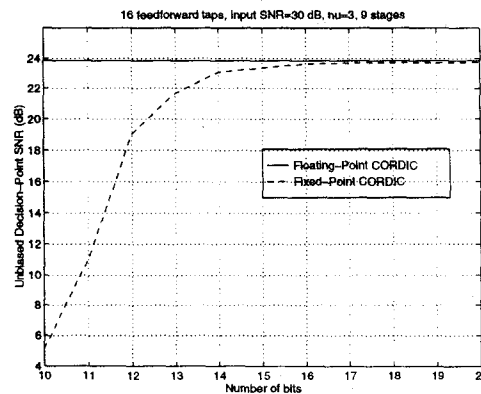


Figure 9: Effect of Fixed-Point Implementation on  $SNR_{MMSE-DFE,U}$  with CORDIC-Based QR Factorization