A Low Power Normalized-LMS Decision Feedback Equalizer for a Wireless Packet Modem

David Garrett, Chris Nicol Bell Laboratories Research, Lucent Technologies Sydney, Australia

{garrettd, chrisn}@lucent.com

ABSTRACT

This paper presents a decision feedback equalizer (DFE) for a high-speed packet modem utilizing the normalized least mean squared (NLMS) tap update algorithm. The equalizer supports up to 43.2 Mbps uncoded data over a wireless channel with a 10% training preamble (48 Mbps with no training). In this work the rapid convergence of the NLMS algorithm is combined a technique for early termination of the tap training process to yield a low power DFE implementation. The low power techniques result in a 43% power reduction over a baseline design. Furthermore, low power synthesis techniques result in an additional 30% power savings on top of the algorithmic power savings.

Categories and Subject Descriptors

B.2.4 [Arithmetic and Logic Structures]: High-speed arithmetic – *algorithms, cost/performance.*

General Terms

Algorithms, Performance, Design.

Keywords

Low power, NLMS, equalization, early termination.

1. INTRODUCTION

One of the challenges of transmitting information at high bits rates over a wireless channel is that frequency selective fading causes significant inter-symbol interference (ISI). In the time domain ISI is modeled by convolving the transmitted symbols with the channel impulse response. An equalizer is a filter that can be used at the received to compensate for the distortion introduced by the channel [1]. The goal of an equalizer is to reverse the effects of the channel, and approximate the originally transmitted symbols. The optimal equalizer structure is a maximum likelihood sequence estimator (MLSE). The complexity of a MLSE grows exponentially with the length of the channel impulse response.

A decision feedback equalizer (DFE) sub-optimally approximates

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*ISLPED*⁷⁰², August 12-14, 2002, Monterey, California, USA. Copyright 2002 ACM 1-58113-475-4/02/0008...\$5.00.

Andrew Blanksby, Chris Howland Agere Systems Holmdel, NJ {blanksby, howland}@agere.com

the MLSE but with a complexity that only grows linearly with the length of the channel and is generally more practical from an implementation perspective [2]-[6].

The DFE described in this paper forms part of a modem for a fixed wireless system. In this system, time division multiple access (TDMA) is used to allow multiple users to share the radio resource. Data and voice traffic is segmented into packets, modulated, and a small sequence of symbols is added as a preamble. Among other functions the preamble symbols allow the equalizer at the receiver to be trained to the instantaneous channel before the data symbols are equalized.

This paper presents a DFE that uses the normalized least mean square (NLMS) iterative algorithm to train the equalizer coefficients. The NLMS algorithm adds a normalization term to the regular LMS tap update equation, providing a variable step size that is inversely proportional to the energy in the filter. The NLMS algorithm corrects the problem of amplification of the gradient noise associated with the regular LMS algorithm, and can lead to more rapid convergence [7]. The DFE operates on T/2 fractionally spaced samples and supports symbol rates up to 8Msymbol/s. The packet format for the synthesized design is programmable with a maximum size of 312 symbols, including up to 32 training symbols. The five modulation formats supported are QPSK, 8PSK, 16QAM, 32QAM, and 64QAM.

The remainder of the paper is organized as follows. Section 2 describes the architecture for the equalizer including the low power arithemetic and terminations circuits. Section 3 presents performance simulations for the equalizer in its operating environment. Section 4 demonstrates implementation results from the synthesized DFE. Finally, section 5 present conclusions.

2. EQUALIZER STRUCTURE

Before describing the architecture of the DFE, it is instructive to discuss how the workload of the equalizer is divided between training and processing the data symbol payload. With an iterative equalizer, the filter must iterate many times over the training preamble in order to obtain satisfactory performance. The iterations need to continue until the filter coefficients have converged to values close to the optimal solution. To sustain a high throughput the equalizer must iterate over the training symbols at a much high rate than the rate at which the symbols are received. This concept is shown pictorially in Figure 1.

The maximum number of training iterations, L, that the architecture can support is related to the training preamble size, T, and the number of symbols the equalizer can process per received symbol. Equation (1) shows the maximum number of training iterations for the DFE architecture as a function of the

processing speed increase, K, the total length of a packet, L, and the length of the training sequence, T. A training iteration is defined as one pass through the training preamble updating the tap weights for each symbol.



$$\Lambda = \frac{KL - (L - T)}{T} \tag{1}$$

For example, with a processing speed increase of 8, a total packet length of 133 symbols, including training sequence of 13 symbols, the equalizer can provide over 72 training iterations. From system simulations it was found that an 8x processing speed increase allowed a sufficient number of iterations for satisfactory equalizer performance for the required range of channel conditions.

2.1 Decision Feedback Equalizer

Figure 2 shows the architecture of the DFE. For this application, the DFE structure uses up to 8 taps in the feed forward (FF) filter to compensate for the delay spread of the channel, a constellation slicer which generates the hard decisions for each symbol, and up to 6 taps in the feedback (FB) filter. When processing the payload the slicer outputs the closest constellation point to the symbol output of the filters, but during training, the slicer outputs the expected training symbol regardless of the slicer input. The slicer outputs a complex error signal that is used by the NLMS algorithm for the tap coefficient update. The DFE continues to train the filter taps during the payload using the decision-directed output of the slicer. In addition a termination algorithm monitors the mean square error (MSE) to measure the convergence of the training process.



Figure 2. Equalizer architecture

The structure of the iterative LMS equalizer is well known in the literature, and there are many standard techniques that have been employed [8]. This paper will specifically focus on the NLMS tap update equations and the early termination of the training process.

2.2 Normalized LMS Equations

The NLMS tap update algorithm provides excellent convergence properties, and can facilitate low power operation of the DFE by speeding up the convergence process. Rather than using a fixed step size as with the LMS algorithm, the NLMS tap equation, shown in equation (2), normalizes the tap update equations based on the magnitude of the energy in the filter (w represents the filter coefficients, x is the data in the filter, e is the error signal from the slicer, and u is the step size). The implementation challenge associated with an NLMS equalizer is that it requires both complex multiplications by the error signal and divisions by the filter energy, and therefore has been avoided in all but a few implementations [9][10].

$$w_{i}^{+} = w_{i} + x_{i} \left(\frac{ue^{*}}{0.5 + \sum |x|^{2}} \right)$$
(2)

However, because the filter is adaptive, the implementation of the tap update does not have to exactly realize equation (2) for satisfactory performance. Indeed, there have been many simplifications of the LMS algorithm including sign updating, which uses the sign of the error signal for the tap update, and power of two tap updates where the error is approximated by a number $e=2^{n}$ [10]-[11]. Tap updates by powers of two has been applied to the NLMS algorithm by dividing by the power of two number that approximates the energy in the filter [10]. The power of two division for NLMS captures the energy of filter and provides good convergence properties without the need for costly division circuits. This paper extends the prior work that concerned real samples, to include a formulation of the NLMS update equation when the filter uses complex samples and coefficients (that are required for a DFE using multi-dimensional constellations). In order to use the power of two updates for complex NLMS, the complex error term is converted into real and imaginary power of two terms. This maintains the NLMS style tap update while significantly reducing the hardware complexity of the complex multiplier and the division. Equation (3) shows the approximation of equation (2) with the power of two $floor(log_2)$ function applied to the complex error. The exponents of the error term, a and b, are calculated by subtracting the numerator and denominator exponents.

$$w_{i}^{+} = w_{i} + x_{x}(s_{r}2^{a} + s_{i}j2^{b}), where$$

$$a = floor(\log_{2}(\operatorname{Re}\{ue^{*}\})) - floor(\log_{2}(0.5 + \sum |x_{i}|^{2}))$$

$$b = floor(\log_{2}(\operatorname{Im}\{ue^{*}\})) - floor(\log_{2}(0.5 + \sum |x_{i}|^{2})) \quad (3)$$

$$s_{r} = sign(\operatorname{Re}\{ue^{*}\})$$

$$s_{i} = sign(\operatorname{Im}\{ue^{*}\})$$

Figure 3 shows a schematic representation of equation (3). The floor(log_2) function is implemented as a leading zeros count, yielding the bit location of the most significant nonzero bit. Because of the symmetry problem associated with 2's complement numbers, the log_2 function is applied to the absolute value of the error term.



Figure 3. Computing the NLMS log₂ numbers

Equation (4) shows the approximated tap updated equation, when the simplifications of equation (3) are applied. The implementation complexity advantage is that the complex multiplication and the division of the original NLMS equation have been replaced with a series of shifts and adds on the original delayed samples.

$$w_{i}^{+} = w_{i} + x_{i} \left(\frac{ue^{*}}{0.5 + \sum |x|^{2}} \right) \approx$$

$$w_{i} + \left[s_{r} 2^{a} \operatorname{Re}\{x_{i}\} - s_{i} 2^{b} \operatorname{Im}\{x_{i}\} + j (s_{i} 2^{b} \operatorname{Re}\{x_{i}\} + s_{r} 2^{a} \operatorname{Im}\{x_{i}\}) \right]$$
(4)

Figure 4 details the NLMS tap update equation that replaces the complex multiplier using the log₂ function. The result is that while the original NLMS update required both a complex multiplication and a division, the approximated NLMS tap update uses only shifts and adds.



Figure 4. Tap update schematic

2.3 Early Termination

Another important low power implementation technique is the use of early termination by predicting when the equalizer has converged to a good solution, and stopping the training of the equalizer at that point. In rare cases, the NLMS tap algorithm will require the maximum number of training iterations to guarantee good convergence. However the average number of training iterations is typically much smaller. The key to early termination of the training process is to have an accurate measure of when the filter has converged to a good solution.

One of the most basic convergence measurements is to set an absolute threshold on the MSE out of the DFE. Thus the NLMS algorithm continues to train on the taps until the MSE drops below the pre-determined threshold. The problem is that this threshold requires knowledge of the channel conditions. If the threshold is set too high, the equalizer will stop training earlier than required, degrading the system performance. On the other hand, if the threshold is too low, the equalizer performs needless training iterations. A better criterion is to monitor the relative change in the error term and find when the MSE starts to reach its asymptotic value. Figure 5 shows the schematic of a circuit used in this DFE design that measures the percentage change in the average MSE of the equalizer and flags a terminate command if the average change drops below a threshold. Using only two combinations of binary shifts on the current MSE, the termination circuit can provide 9.375%, 6.25% or 3.125% percentage change termination criteria. The circuit provides an effective early termination test that does not require knowledge of the channel and operating SNR.



Figure 5. Early termination circuit

3. ALGORITHMIC PERFORMANCE

The architectural optimizations for the adaptive NLMS DFE presented in section 2 significantly reduce the power dissipation and complexity, but it is important to measure their impact on the system performance. The DFE was simulated for an 8 dB Rician fading channel with an exponential delay power profile with a half symbol standard deviation. A packet length of 133 and a length 13 Barker preamble were used.

3.1 NLMS Tap Update

Figure 6 shows the bit error rate (BER) and packet error rate (PER) performance of the equalizer for an uncoded system with an 8PSK constellation using the NLMS tap update equations using both a floating point NLMS equalizer (denoted float) and a fixed point equalizer using equations (3) and (4) (denoted fixed). Furthermore the figure shows performance both with early termination enabled (6% MSE criteria, denoted 'w/ term') and when the equalizer trains for exactly 63 iterations for each packet (denoted 'w/o term'). The key importance of this figure is that the approximations used for the NLMS update have negligible impact on system performance compared to the full floating point NLMS equalizer. Secondly, the early termination has a negligible impact on the system performance when compared with the equalizer always running 63 iterations, indicating that it accurately predicts the convergence of the tap update process.

Figure 7 shows the BER curves for the equalizer with and without early termination enabled for the 8 dB Rician channel for various signal constellations. For both QPSK and 8PSK, the system performs slightly better with early termination enabled. The curves start to diverge near the error floor, which is a function of the signal constellation (around 10^{-3} for 32QAM and 64QAM). This error floor drops considerably once coding is introduced into the system.



Figure 6. Floating vs. fixed point 8PSK performance



Figure 7. Fixed point system BER w/ termination



Figure 8. Average iterations w/ termination

3.2 Early Termination

The average number of training iterations was measured at various geometries for each constellation as seen in Figure 8. The range of average iterations across all measured SNRs and constellations was between 5 to 12 iterations, translating into reductions of between 92% and 81% respectively over the maximum value of 63, substantially reducing the energy required to train the equalizer taps.

Figure 8 shows an interesting trend for the early termination algorithm, the average number of iterations needed increased slightly with increasing SNR. One hypothesis to explain this effect is that the higher SNR reduces the MSE out of the filter, therefore making the relative termination criteria from section 2.3 harder to meet. Intuitively, one might expect higher SNR to require fewer training iterations. This does not detract from the early termination performance because, most importantly, it still runs enough training iterations at the low SNR as seen in Figure 6, and the overall range of average iterations is not large across the range of input SNRs.

4. IMPLEMENTATION RESULTS

The DFE design was synthesized with 8 FF taps, 6 FB taps in TSMC 0.18 um, 1.8V, 6LM CMOS technology with an input sample precision of 10-bits. The coefficients are stored with 16-bit precision, while only the 10 MSBs are used in the filtering process. The entire design was coded in VHDL and synthesized using the Cadence physical synthesis tool, PKS. The equalizer was designed to run 8 times faster than the symbol rate with each complex multiplier in the FF filter shared between two taps. At 128 MHz, the equalizer dissipates an estimated 95 mW while performing over 7 billion effective multiplications per second (4 multipliers for each complex multiplication) at :

- 512 million, 10x10 complex multiplications/s (FF taps)
- 384 million 4x10 complex multiplications/s (FB taps)
- 896 million NLMS complex tap update/s

Figure 9 shows a layout of the placed design, with an active area of 2mm x 2mm, 180k effective gates, and 30 Kbits of memory.

4.1 Power Dissipation

To measure the impact of early termination on power dissipation, the power was estimated using the synthesis tool. The backannotated netlist was simulated with a set of operational vectors generated from a testbench modeling the system environment. The equalizer was run in a mixture of modes, such as enabling 4 or 8 FF taps, and 3 or 6 FB taps respectively (denoted as $\{4,3\}$ or $\{8,6\}$), and with and without early termination enabled. The signal activities from the design were fed back into the synthesis tool to perform low power RTL synthesis (LPS) to include clock gating, sleep mode, and gate level power optimizations [12]. Table 2 shows the power estimation results for the DFE when configured with QPSK modulation both for the original placed design and the design synthesized with LPS. The equalizer was run with four scenarios, $\{4,3\}$ and $\{8,6\}$ with early termination both disabled and enabled.

Table 1 shows the substantial power savings for early termination. In the original design, the early termination reduces the power dissipation by an average of 43% over the DFE running the full number of iterations across both $\{4,3\}$ and $\{8,6\}$ modes. Furthermore, in the early termination mode, the LPS resulted in a

further power savings of 51%. The overall power savings in the architecture from full iterations to the final LPS synthesized design is almost 73%.

Table 1. Power estimation results QPSK (1.8V)

Implementation	Power (mW) Original	Power (mW) after LPS
Equalizer w/o termation {4,3} (63 iterations)	337 mW	192 mW
Equalizer w/o termination {8,6} (63 iterations)	341 mW	202 mW
Equalizer w/ terminations {4,3} (6 iterations)	189 mW	92 mW
Equalizer w/ termination {8,6}	196 mW	95 mW



Figure 9. Chip Layout

Figure 10 shows a breakdown of the power dissipation in the design when early termination is enabled. The FF and FB filters dissipated 34% and 9% respectively of the total power consumption in the filter. The key item to note is that the termination block only contributes 3% of the total power, yet allows the equalizer to reduce its power dissipation by over 43%.

5. CONCLUSIONS

This paper has presented two main algorithmic techniques for power reduction in the implementation of an 8Msymbol/s NLMS DFE (supporting up to 43.2 Mbps uncoded data with 64QAM and a 10% training preamble). The first technique presented was an efficient implementation the NLMS algorithm with power of two tap updates. It was shown that the NLMS approximations still allow fast convergence in the filter. The second technique was a method for detecting when the equalizer had converged during the training period and then freezing the tap updates. The early termination and the LPS resulted in almost 73% power savings over the baseline DFE structure.

Power Dissipation Breakdown in DFE



Figure 10. Power dissipation breakdown

6. ACKNOWLEDGEMENTS

The authors would like to thank Richard Owen and Christina Chu from Cadence for their assistance with the low power synthesis.

7. REFERENCES

- [1] S. Qureshi, "Adaptive Equalization," *Proceedings IEEE*, vol. 73, pp. 1349-1386, September 1985.
- [2] F. Lu and H. Samueli, "A Reconfigurable Decision-Feedback Equalizer Chip Set Architecture for High Bit-rate QAM Digital Modems," *ICASSP*, 1991. pp. 1185-1188.
- [3] Y. Jemaa, S. Cherif, M. Jaidane, S. Marcos, "Design of Decision Feedback Equalizer with short training sequence," *VTC 2000*, pp. 2920-2925.
- [4] J. Liu, S. Gelfand, "Adaptively Optimized Decision Feedback Equalization for Convolutional Coding," *ICC 2001*, pp. 403-407.
- [5] M. Rupp, "On the Learning Behavior of Decision Feedback Equalizers," Asilomar Conference on Signals, Systems, and Computers, 1999, pp. 514-518.
- [6] D. Shin, H. Park, M. Sunwoo, "A DFE Equalizer ASIC chip using the MMA Algorithm," *IEEE International ASIC/SOC Conference*, 2000, pp. 70-74.
- [7] S. Haykin, *Adaptive Filter Theory (3rd Edition)*, Prentice-Hall: New Jersey, 1996. pp 432-439.
- [8] C. Nicol, P. Larsson, K. Azadet, and J. O'Neill, "A Low-Power 128-Tap Digital Adaptive Equalizer for Broadband Modems," *JSSC*, vol 32. No 11, pp. 1777-1789, Nov 1997.
- [9] H. Chi, "Efficient Computation Schemes and Bit-Serial Architectures for Normalized LMS Adaptive Filtering in Audio Applications," *ISCAS*, 2000, Vol. III, pp. 666-669.
- [10] A. Harada, K. Nishikawa, H. Kiya, "A Pipelined Architecture for the Normalized LMS Adaptive Digital Filters," *IEEE APCCAS*, 1998, pp. 73-76.
- [11] H. Samueli, "The design of Multiplierless digital data transmission filters with powers-of-two coefficient," IEEE International Telecommunications Symposium, 1990, pp. 425-429.
- [12] http://www.cadence.com/datasheets/low_power_synth_optio n.htm