A LOW-POWER VLSI DESIGN METHODOLOGY FOR HIGH BIT-RATE DATA COMMUNICATIONS OVER UTP CHANNEL

Manish Goel and Naresh R. Shanbhag

Coordinated Science Lab./ECE Department,
Univ. of Illinois at Urbana-Champaign,
1308 W. Main Street, Urbana IL-61801.
E-mail : [mgoel,shanbhag]@uivlsi.csl.uiuc.edu

ABSTRACT

Presented in this paper is a systematic methodology to design low-power integrated transceivers for broadband data communications over unshielded twisted-pair (UTP) channels. The design methodology is based upon two algorithmic low-power techniques referred to as *Hilbert transformation* and *strength reduction* and a high-speed pipelining technique referred to as *relaxed look-ahead transformation*. Finite-precision requirements and power savings are presented. The application of these techniques to design low-power and high-speed 155.52 Mb/s ATM-LAN and 51.84 Mb/s VDSL transceivers is illustrated.

1. INTRODUCTION

Numerous high-bit rate digital communication technologies are currently being proposed that employ unshielded twisted-pair (UTP) wiring. These include asymmetric digital subscriber loop (ADSL), high-speed digital subscriber loop (HDSL), very high-speed digital subscriber loop (VDSL), asynchronous transfer mode (ATM) LAN [2] and broadband access [3]. The above mentioned transmission technologies are especially challenging from both algorithmic and VLSI viewpoints. This is due to the fact that high data rates (51.84 Mb/s to 155.52 Mb/s) need to be achieved over severely bandlimited (less than 30 MHz) UTP channels which necessitates the use of highly complex digital communications algorithms. Low-cost solutions require an integrated approach whereby algorithmic concerns such as signal-to-noise ratio (SNR) and bit-error rate (BER) along with VLSI constraints such as power dissipation, area, and speed, are addressed in a joint manner.

In this paper, we present a systematic design methodology which incorporates algorithmic and VLSI architectural issues in the design of low-power transceivers for broadband data communications over UTP channels. The application of these techniques to the design of 155.52 Mb/s ATM-LAN and 51.84 Mb/s VDSL is demonstrated.

2. THE CHANNEL

In this section, we will describe the UTP-based channel for ATM-LAN and VDSL.

2.1. The Channel Response

In the LAN environment, the two major causes of performance degradation for transceivers operating over UTP wiring are *propagation loss* and *crosstalk* generated between adjacent wire pairs. The worst-case propagation loss is given in the TIA/EIA-568 draft standard for category 3 cable [1] as

$$L_P(f) = 2.320\sqrt{f} + 0.238f, \qquad (1)$$

where the propagation loss $L_P(f)$ is expressed in dB per 100 meters and the frequency f is expressed in MHz.

Similarly, the worst-case NEXT loss model for a single interferer is also given in the TIA/EIA draft standard [1]. This loss can be expressed as:

$$L_N(f) = 43 - 15\log f, \qquad (2)$$

where the frequency f is in megahertz.

In case of VDSL, the performance degradation is due to propagation loss and far-end crosstalk (FEXT). The propagation loss characteristics of a BKMA cable, which is typically employed in this environment, are similar to that of a category 5 cable specified in the TIA/EIA-568A Standard [1], as follows:

$$L_P(f) = 3.597\sqrt{f} + 0.043f + 0.0914/\sqrt{f}, \qquad (3)$$

where the propagation loss $L_P(f)$ is expressed in dB and the frequency f is expressed in MHz.

The equal-level FEXT ($EL - FEXT$) loss in a FEXT dominated environment can be written as:

$$EL - FEXT == \frac{K}{\Psi f^2 d} \qquad K = (49/N)^{0.6}, \qquad (4)$$

where Ψ is the coupling constant which equals 10^{-10} for 1% equal level 49 interferers, d is the distance in kilofeet, f is the frequency in kilohertz and N is the number of interferers.

2.2. The Carrierless Amplitude/Phase (CAP) Modulation Scheme

In the following, we describe a bandwidth-efficient two-dimensional passband transmission scheme referred to as carrierless amplitude/phase modulation (CAP). Note that 64-CAP modulation scheme is the standard for ATM-LAN over UTP-3 at 155.52 Mb/s and 16-CAP is the standard for 51.84 VDSL [3].

The block diagrams of a digital CAP transmitter and receiver are shown in Fig. 1. In the transmitter (see Fig. 1(a)), the bit stream to be transmitted is first passed through a scrambler in order to randomize the data. The scrambled bits are then fed into a CAP encoder, which maps blocks of m bits into one of $k = 2^m$ different complex symbols $a(n) = a_r(n) + ja_i(n)$. After the encoder, the symbols $a_r(n)$ and $a_i(n)$ are fed to digital shaping filters. The outputs of the filters are subtracted and the result is passed through a digital-to-analog converter (DAC), which is followed by an interpolating low-pass filter (LPF).

A typical CAP receiver (see Fig. 1(b)) consists of an analog-to-digital converter (ADC) followed by a parallel arrangement of two adaptive digital filters. The adaptive filters in Fig. 1 are referred to as a T/M fractionally spaced linear equalizers (FSLEs). In addition to the FSLEs, a CAP receiver can have a NEXT canceller and a decision feedback

0-7803-4455-3/98/$10.00 © 1998 IEEE

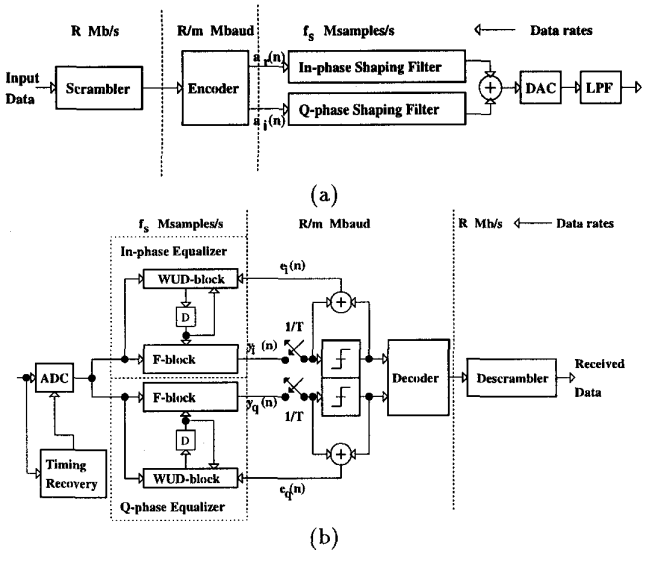


Figure 1: Carrierless Amplitude Phase (CAP) transceiver (a) transmitter and (b) receiver.

equalizer (DFE). The decision to incorporate a NEXT canceller and/or a DFE depends upon the channel impairments and the capabilities of an FSLE. For example, a NEXT canceller is required for 155.52 Mb/s ATM-LAN to cancel NEXT. Similarly, in case of 51.84 Mb/s VDSL, the presence of radio frequency interference (RFI) necessitates the use of a DFE.

3. LOW-POWER AND HIGH-SPEED EQUALIZER ARCHITECTURES

For high bit-rate applications, the equalizers in Fig. 1(b) need to be implemented as high sample-rate adaptive filters, which are typically power-hungry. In this section, we show how the techniques of Hilbert transformation [4] and strength reduction [5] can be employed to design low-power receivers. Also discussed is the technique of *relaxed look-ahead transformation* [7] pipelining the strength-reduced equalizer.

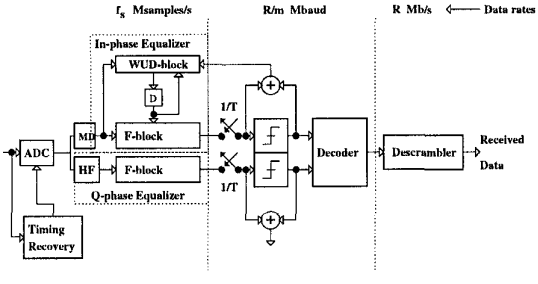


Figure 2: Hilbert-based CAP equalizer.

3.1. Low-Power FSLE Architecture via Hilbert Transformation

It can be shown that the in-phase and the quadrature-phase equalizers of the CAP receiver in Fig. 1 are Hilbert transforms of each other. In this subsection, we exploit this relationship to obtain a low-power structure.

If the in-phase and the quadrature-phase equalizer filter impulse responses are denoted by $f(n)$ and $\tilde{f}(n)$, respectively, then

$$\tilde{f}(n) = h_I(n) * f(n), \tag{5}$$

where the symbol "$*$" denotes convolution. Let $y_i(n)$ and $y_q(n)$ denote the in-phase and the quadrature-phase components of the receive filter output, respectively, and $x(n)$ denote the input. Employing (5), the equalizer outputs can be expressed as

$$\begin{aligned} y_i(n) &= f(n) * x(n), \\ y_q(n) = \tilde{f}(n) * x(n) &= f(n) * [h_I(n) * x(n)]. \end{aligned} \tag{6}$$

From (6), we see that $y_q(n)$ can be computed as the output of a filter, which has the same coefficients as that of the in-phase filter with the Hilbert transform of $x(n)$ as the input. Hence, the CAP receiver in Fig. 1(b) can be modified into the form [4] as shown in Fig. 2, where **HF** is the Hilbert filter. The ideal Hilbert filter is approximated by an M-tap windowed version. From a power dissipation perspective, the Hilbert filter length should be as small as possible. The power dissipation versus performance trade-off has been explored in [4] where it was shown that for 51.84 Mb/s ATM-LAN, assuming $N = 32$, we obtain SNR_o values of more than 25 dB, when the Hilbert transformer length M is more than 33. This results in a 21% power savings.

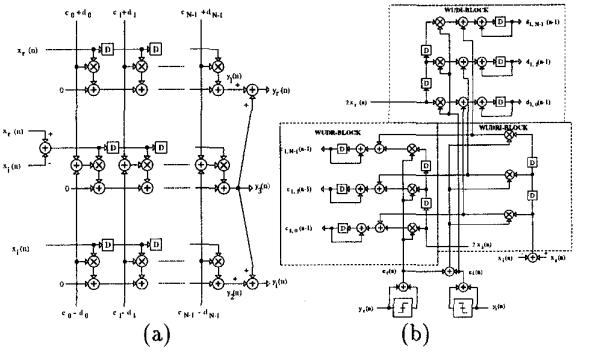


Figure 3: Strength-reduced equalizer (a) **F**-block and (b) **WUD**-block.

3.2. Low-power Complex Adaptive Filter via Strength Reduction

The product of two complex number $(a+jb)$ and $(c+jd)$ is given by $(a+jb)(c+jd) = (ac-bd) + j(ad+bc)$. A direct-mapped architectural implementation of this would require a total of four real multiplications and two real additions to compute the complex product. Application of strength reduction involves reformulating the above multiplication as follows

$$(a-b)d+a(c-d) = ac-bd, \quad (a-b)d+b(c+d) = ad+bc, \tag{7}$$

where we see that strength reduction reduces the number of multipliers by one at the expense of three additional adders. Typically, multiplications are more expensive than additions and hence we achieve an overall savings in hardware. The strength reduced **SR** architecture [5] is obtained by applying strength reduction transformation at the algorithmic level.

Consider an N-tap serial complex LMS filter described by the following equations

$$\begin{aligned} e(n) &= y_d(n) - \mathbf{W}^H(n-1)\mathbf{X}(n), \\ \mathbf{W}(n) &= \mathbf{W}(n-1) + \mu e^*(n)\mathbf{X}(n), \end{aligned} \tag{8}$$

where $\mathbf{W}(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^T$ is the weight vector with $\mathbf{W}^H(n)$ being the Hermitian (transpose and complex conjugate), $\mathbf{X}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T$ is the input vector, $e^*(n)$ is the complex conjugate of the adaptation error $e(n)$, μ is the step-size, and $y_d(n)$ is the desired signal.

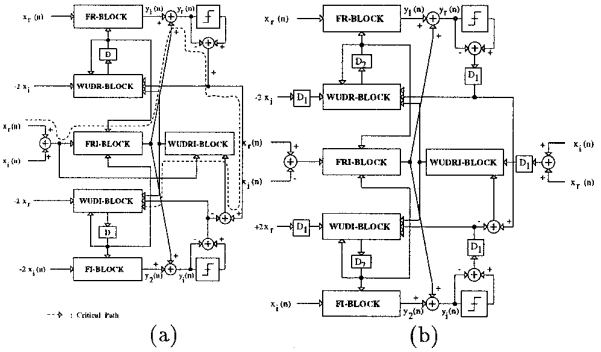
Traditionally, the complex LMS algorithm is implemented via the cross-coupled **CC** architecture, where four real **F**-blocks and four real **WUD**-blocks are employed requiring a total of $8N$ multipliers and $8N$ adders. Applying strength reduction to (8) results in the following equations, which describe the **F**-block computations of the **SR** architecture [5]:

$$y_1(n) = \mathbf{c}_1^T(n-1)\mathbf{X}_r(n),\ y_2(n) = \mathbf{d}_1^T(n-1)\mathbf{X}_i(n)$$
$$y_3(n) = -\mathbf{d}^T(n-1)\mathbf{X}_1(n),$$
$$y_r(n) = y_1(n) + y_3(n),\ y_i(n) = y_2(n) + y_3(n), \quad (9)$$

where $\mathbf{X}_1(n) = \mathbf{X}_r(n) - \mathbf{X}_i(n)$, $\mathbf{c}_1(n) = \mathbf{c}(n) + \mathbf{d}(n)$, and $\mathbf{d}_1(n) = \mathbf{c}(n) - \mathbf{d}(n)$. Similarly, the **WUD** computation is described by,

$$\mathbf{c}_1(n) = \mathbf{c}_1(n-1) + \mu[\mathbf{eX}_1(n) + \mathbf{eX}_3(n)]$$
$$\mathbf{d}_1(n) = \mathbf{d}_1(n-1) + \mu[\mathbf{eX}_2(n) + \mathbf{eX}_3(n)], \quad (10)$$

where $\mathbf{eX}_1(n) = 2e_r(n)\mathbf{X}_i(n)$, $\mathbf{eX}_2(n) = 2e_i(n)\mathbf{X}_r(n)$, $\mathbf{eX}_3(n) = e_1(n)\mathbf{X}_1(n)$, $e_1(n) = e_r(n) - e_i(n)$, $\mathbf{X}_1(n) = \mathbf{X}_r(n) - \mathbf{X}_i(n)$. It is easy to show that the **SR** architecture (see Fig. 3) requires only $6N$ multipliers and $8N+3$ adders. This is the reason why the **SR** architecture results in $21-25\%$ power savings [5] over the **CC** architecture.



(a) (b)

Figure 4: Strength-reduced equalizer (a) Serial and (b) Pipelined.

3.3. Pipelined Strength-reduced (PIPSR) Architecture

The block diagram of the **SR** architecture is shown in Fig. 4(a). The dotted line in Fig. 4(a) indicates the critical path of the **SR** architecture. As explained in [5], both the **SR** as well as **CC** architectures are bounded by a maximum possible clock rate due the computations in this critical path. This throughput limitation is eliminated via the application of the *relaxed look-ahead transformation* [7] to the **SR** architecture (see (9-10)). Application of relaxed look-ahead to the **SR** architecture in (9-10) results in the following equations that describe the **F**-block computations in the **PIPSR** architecture:

$$y_1(n) = \mathbf{c}_1^T(n-D_2)\mathbf{X}_r(n),\ y_2(n) = \mathbf{d}_1^T(n-D_2)\mathbf{X}_i(n)$$
$$y_3(n) = -\mathbf{d}^T(n-D_2)\mathbf{X}_1(n)$$
$$y_r(n) = y_1(n) + y_3(n),\ y_i(n) = y_2(n) + y_3(n), \quad (11)$$

where D_2 is the number of delays introduced before feeding the filter coefficients into the **F**-block. Similarly, the computation of the **WUD** block of the **PIPSR** architecture are given by

$$\mathbf{c}_1(n) = \mathbf{c}_1(n-D_2) + \mu \sum_{i=0}^{LA-1} \big[\mathbf{eX}_1(n-D_1-i) + \mathbf{eX}_3(n-D_1-i)\big] \quad (12)$$

$$\mathbf{d}_1(n) = \mathbf{d}_1(n-D_2) + \mu \sum_{i=0}^{LA-1} \big[\mathbf{eX}_2(n-D_1-i) + \mathbf{eX}_3(n-D_1-i)\big], \quad (13)$$

where $\mathbf{eX}_1(n)$, $\mathbf{eX}_2(n)$ and $\mathbf{eX}_3(n)$ are defined in the previous subsection, $D_1 \geq 0$ are the delays introduced into the error feedback loop and $0 < LA \leq D_2$ indicates the number of terms considered in the sum-relaxation. A block level implementation of the **PIPSR** architecture is shown in Fig. 4(b) where D_1 and D_2 delays will be employed to pipeline the various operators such as adders and multipliers at a fine-grain level. The high-throughput of the **PIPSR** architecture can also be traded-off with supply voltage reduction resulting in additional power savings [5] of $40-69\%$. Therefore, the **PIPSR** architecture results in $60-90\%$ power savings as compared to the serial **CC** architecture. In a similar manner, the relaxed look-ahead transformation can be employed to pipeline the real adaptive filters.

4. SYSTEMATIC DESIGN METHODOLOGY AND EXAMPLES

In this section, we present our design methodology for the high-speed communication system design. We determine the encoder and shaping filters in the transmitter and adaptive filter lengths, precisions and pipelining levels in the receiver.

4.1. Design Methodology and Applications

The design methodology consists of the following steps:

Step 1: Determine the specifications for the signal constellation and the spectrum based on the channel constraints.

Step 2: Determine adaptive filters lengths via the floating-point simulations. Apply low-power transformations given in section 3. Let N be the tap-length and $SNR_{o,fl}$ be the output SNR in dB of the floating point algorithm after this step.

Step 3: Determine the precision for the adaptive filters such that the fixed-point SNR is close to the floating-point SNR employing the following equations [6] for an **SR** architecture,

$$B_{F,SR} > \frac{1}{2}\log_2\left(\frac{N\sigma_x^2}{4\beta\sigma_d^2}\right) + \frac{SNR_{o,fl}(dB)}{6} \quad (14)$$

$$B_{WUD,SR} \geq \frac{1}{2}\log_2\left(\frac{1}{\mu^2\sigma_x^2\sigma_d^2}\right) + \frac{SNR_o(dB)}{6}, \quad (15)$$

where σ_x^2 is the input power to the **F**-block, σ_d^2 is the power of symbol constellation (or the desired signal) and $\beta \ll 1$ is the ratio between the quantization noise at the output and floating point MSE. Also, SNR_o is the desired SNR at the output. Typically, we choose $SNR_o = SNR_{o,fl}$ because we assume that the quantization error at the output is small. Similar expressions can be obtained for the real adaptive filters also.

Step 4: Calculate critical path delay for the fixed-point architectures, and pipeline (if needed) via relaxed look-ahead transformation (see section 3.3) by choosing appropriate values of D_1, D_2 and LA.

4.2. 155.52Mb/s ATM-LAN

In this subsection, we will employ the proposed methodology to design 155.52 Mb/s ATM-LAN transceiver for UTP-3 wiring. Given the data rate of $R = 155.52$ Mb/s and the FCC imposed limit of 30 MHz on the transmit spectrum, we choose a value of $m = 6$ to get the symbol rate $1/T$ to be 25.92 Mbaud. Choosing $0 - 30$ MHz spectrum, we obtain a value of $\alpha = 0.15$. Clearly, the sample rate f_s needs to be greater than 60 MHz. We choose the next nearest multiple of the symbol rate or 77.76 MHz as the value of f_s. As the excess bandwidth is less than 100%, a NEXT canceller is required to suppress NEXT as shown in Fig. 5.

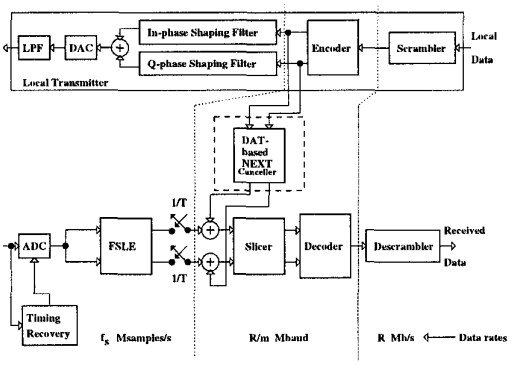


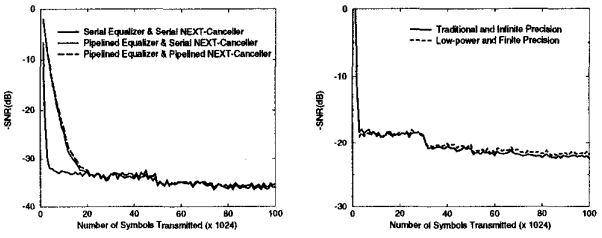
Figure 5: 155.52 Mb/s CAP transceiver.

Next, we decide the lengths of the adaptive filters according to **Step 2**. It has been observed in [2] that the NEXT canceller span is weakly dependent upon the equalizer and that it should be about $1\mu s$ or greater. Given the symbol rate of 25.92 Mbaud, we find that the number of NEXT canceller taps $N_{nextc} \geq 26$. We chose a value of $N_{nextc} = 32$ for our simulations. The equalizer length is determined via the simulations, which show that with symbol span equal to 40, $SNR_o = 36.1$ dB is obtained. Note that this is also the value of $SNR_{o,fl}$ employed in **Step 3** to determine the precisions. Employing (14) and (15), we choose the **F**-block precision in the NEXT canceller $B_{F,nextc} = 10$ bits and the **WUD**-block precision in the NEXT canceller $B_{WUD,nextc} = 14$ bits. A similar analysis can be applied to the equalizer. In doing so, we obtained the **F**-block precision $B_{F,eq} = 12$ bits and $B_{WUD,eq} = 18$ bits for the equalizer.

Given the tap lengths, the precisions (derived above) and the assumption of a 1 bit adder delay of 1 ns, it can be shown that pipelined architectures for the NEXT canceller and the equalizer becomes necessary. Hence, we employ the pipelined strength reduced adaptive filter architecture for the NEXT canceller with $D_1 = 16$, $D_2 = 4$, and $LA = 1$ and the pipelined FSLE architecture with $D_1 = 84$, $D_2 = 4$, and $LA = 1$. The simulation results for the finite-precision, pipelined 155.52 Mb/s ATM-LAN receiver is shown in Fig. 6(a), where we see that the final $SNR_o = 35.3$ dB provides a noise margin of 5.8 dB, which is quite sufficient for practical purposes.

4.3. 51.84 Mb/s VDSL

For VDSL, we need a DFE with feedforward equalizer (FFE) span of 16 symbol periods and a feedback equalizer (FBE) with 8 taps. Clearly, other combinations of FFE and FBE spans are also possible. The precisions of the equalizer were: $B_{F,ffe} = 10$ bits and $B_{WUD,ffe} = 16$ bits. The low-power strength reduced DFE had precisions: $B_{F,fbe} = 9$ bits and $B_{WUD,fbe} = 13$ bits. A Hilbert filter with length $M = 65$ and coefficient precision of 9 bits was chosen. With these parameters, simulation results (see Fig. 6(b)) indicate that the finite-precision VDSL receiver can achieve an $SNR_o = 21.8$ dB. This is 0.3 dB lower than the SNR_o of the floating point receiver model. Note that with $SNR_{o,ref} = 21.5$ dB (corresponding to a $BER = 10^{-7}$ for a 16-CAP transceiver), we have a noise margin of 0.3 dB. Clearly, it is harder to achieve the desired data rates with comfortable noise margins in a VDSL environment as compared to the ATM-LAN environment.



(a) (b)

Figure 6: Simulation results for (a) 155.52 Mb/s ATM-LAN and (b) 51.84 Mb/s VDSL.

5. ACKNOWLEDGMENTS

This work was supported via National Science Foundation CAREER Award MIP-9623737.

6. REFERENCES

[1] *Commercial Building Telecommunications Cabling Standard*, TIA/EIA-568-A Standard, 1994.

[2] G. H. Im and J. J. Werner, "Bandwidth-efficient digital transmission up to 155 Mb/s over unshielded twisted-pair wiring," *IEEE J. Select. Areas Comm.*, vol. 13, no. 9, pp. 1643-1655, Dec. 1995.

[3] D. D. Harman, G. Huang, G. H. Im, M.-H. Nguyen, J.-J. Werner, and M. K. Wong, "Local distribution for interactive multimedia TV," *IEEE Multimedia Magazine*, pp. 14-23, Fall, 1995.

[4] R. Hegde and N. R. Shanbhag, "A low-power phase-splitting adaptive equalizer for high bit-rate communications systems," *Proc. of IEEE Workshop on Signal Processing Systems*, Nov. 1997, Leicester, U.K..

[5] N. R. Shanbhag and M. Goel, "Low-power adaptive filter architectures and their application to 51.84 Mb/s ATM-LAN," *IEEE Trans. on Signal Processing*, vol. 45, no. 5, pp. 1276-1290, May 1997.

[6] M. Goel and N. R. Shanbhag, "A pipelined strength reduced adaptive filter: Finite precision analysis and application to 155.52 Mb/s ATM-LAN," *1997 Proc. Midwest Symposium on Circuits and Systems*, Sacramento, CA, August 1997.

[7] N. R. Shanbhag and K. K. Parhi, *Pipelined Adaptive Digital Filters*. Kluwer Academic Publishers, 1994.

[8] B. R. Petersen and D. D. Falconer, "Minimum mean square equalization in cyclostationary and stationary interference: Analysis and subscriber line calculations", *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 6, pp. 931-940, Aug. 1991.