DYNAMIC ALGORITHM TRANSFORMATIONS (DAT) FOR LOW-POWER ADAPTIVE SIGNAL PROCESSING[†]

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 are algorithm transformation techniques for adaptive signal processing, which allow dynamic alteration of algorithm properties in response to signal nonstationarities. These transformations, referred to as *dynamic algorithm transformations* (DAT), jointly optimize algorithm and circuit performance measures such as signal-to-noise ratios (SNR) and power dissipation (P_D), respectively. A DAT-based signal processing system is composed of a signal monitoring algorithm (SMA) block and a signal processing algorithm (SPA) block. First, computation of the theoretical power-optimum SPA configuration incorporating signal transition activity is presented. Next, practical SMA schemes are developed, which achieved power reduction by a combination of powering down the filter taps *and* modifying the coefficients. The DAT-based adaptive filter is then employed as a near-end cross-talk (NEXT) canceller in 155.52 Mb/s ATM-LAN over category 3 wiring. Simulation results indicate that the power savings for the NEXT canceller range from 21% – 62% as the cable length varies from 100 meters to 70 meters.

I. Introduction

Modern day communication systems are characterized by high-bit rates over severely bandlimited channels. This requires the implementation of complex signal processing algorithms which result in high power dissipation. Therefore, low-power design techniques [2] are of great interest in such applications and have been proposed at all levels of the VLSI design hierarchy including those at the logic level [7]. Algorithm transformation techniques have been proposed [3] for power optimization at the algorithmic level. Originally developed for high-throughput applications [12], algorithm transformation techniques modify the algorithm structure and/or performance in order to introduce 'VLSI-friendly' features. These techniques include look-ahead pipelining [12], relaxed look-ahead [17], associativity [14], strength-reduction [16], [3] and block-processing [13] have been employed to design low-power and high-throughput systems.

We refer to the existing algorithm transformations mentioned above as *static algorithm transformations* (SAT), because these are applied during the algorithm design phase assuming a worst-case scenario and their implementation is time-invariant. Most real-life signal environments are non-stationary and hence significant power savings can be expected if the algorithm and architecture can be dynamically tailored to the input. This gives rise to the general concept of 'data-driven signal processing' [4], where the algorithm workload [10] and the voltage supply are varied in real-time to optimize the power dissipation.

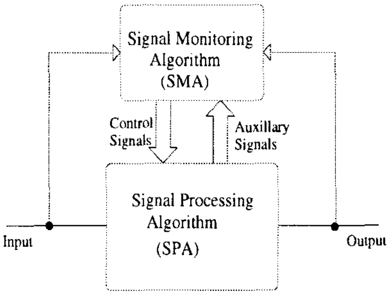
Signal Monitoring Algorithm (SMA)
Control Signals
Auxillary Signals
Signal Processing Algorithm (SPA)
Input
Output

Fig 1. DAT based Implementation

In this paper, we are proposing *dynamic algorithm transformations* (DAT) as another approach to data-driven signal processing, whereby the theoretical power-optimum signal processing architecture is first determined and then practical methods to realize this optimum are developed. As adaptive filters [5] are inherently data-driven filters therefore it is quite natural to develop DAT techniques for these filters. We calculate the power-optimum adaptive filter configuration and then propose the DAT-based structure shown in Fig. 1 to approach this optimum. The system in Fig. 1 consists of two major blocks - the signal processing algorithm (**SPA**) block and the signal monitoring algo-

[†] This work was supported by National Science Foundation CAREER award MIP-9623737.

rithm (**SMA**) block. The **SPA** block implements the main signal processing function, which would vary over time. The **SMA** block decides the instant and the extent of change to the **SPA** block so as to optimize a circuit performance measure such as power dissipation while maintaining the algorithm performance such as the mean-squared error.

The dynamic power dissipation in digital CMOS VLSI systems are a function of input signal statistics and the filter topology. We show that the proposed DAT-based adaptive filters achieve the power-optimum configuration for a white input signal if the signal statistics are ignored and approach the optimum configuration otherwise. A key feature of DAT-based adaptive filters presented here is that filter taps are dynamically powered down *and* the remaining coefficients are adapted to maintain a satisfactory algorithm performance. This makes the proposed adaptive equalizers cover a more general class of non-uniformly spaced equalizers.

Related work includes a 128 tap adaptive equalizer proposed in [11], which has varying filter order and and precision. In this paper, we provide simulation results to show the performance of the proposed DAT-based filter when employed as a near-end cross-talk (NEXT) canceller in 155.52 Mb/s ATM-LAN [6] over category 3 wiring. Simulation results indicate that the power savings for a NEXT canceller range from 21% – 62% as the cable length varies from 100 meters to 70 meters.

Section II provides some preliminaries regarding power dissipation and adaptive filters. In section III, we present an algorithm to calculate the power-optimum adaptive filter configuration for a given level of performance. The DAT-based adaptive filter is presented in section IV. In section V, we present simulation results demonstrating the performance of the proposed DAT-based adaptive filters.

II. Preliminaries

In this section, we present the relevant preliminaries regarding power dissipation in the CMOS circuits [18] and adaptive filters [5].

A. Dynamic Power Dissipation

The dynamic power dissipation [18] P_D is a major component of total power dissipation in digital CMOS VLSI and can be approximated by,

$$P_D = \sum T_i C_i V_{dd}^2 f_s, \tag{2.1}$$

where T_i is the average transition activity, C_i is the switching capacitance of hardware unit i, V_{dd} is the supply voltage and f_s is the frequency of operation. Most of the existing power reduction techniques [2] involve reducing one or more of the four quantities T_i, C_i, V_{dd} and f_s.

B. Adaptive Filtering

Let $\mathbf{X}(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T$ be the input signal vector to the adaptive filter, and $\mathbf{W}(n) = [w_1(n), w_2(n), \cdots, w_N(n)]^T$ be the filter coefficient vector. We want to find the vector which minimizes the mean squared error (MSE), $J(n) = E[e^2(n)]$, where $e(n) = d(n) - \mathbf{W}^T(n-1)\mathbf{X}(n)$ and $E[]$ is the expectation operator. The least mean square (LMS) algorithm[19] has been extensively employed for this purpose. This algorithm is based on the stochastic gradient approach and is given by,

$$y(n) = \mathbf{W}^T(n-1)\mathbf{X}(n) \tag{2.2(a)}$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mu e(n)\mathbf{X}(n) \tag{2.2(b)}$$

where $e(n) = d(n) - y(n)$ is the error signal, $d(n)$ is the desired signal, $y(n)$ is the filter output, μ is the step-size. If the step-size is chosen to be small enough, the LMS algorithm converges very close to the optimum solution.

In this paper, we present DAT techniques for adaptive filters to dynamically alter T_i in (2.1) by powering down taps in (2.2) such that the average power dissipation P_D is minimized while maintaining satisfactory value of MSE.

III. Power Optimization Problem

In this section, we describe the problem of joint optimization of power dissipation and algorithm performance for an adaptive signal processing system.

A. Power Dissipation

In computing the optimum configuration, we will assume that in an N-tap adaptive filter, any filter tap can be powered up or down (and not just the trailing/leading taps [10], [11]). This feature can be algorithmically characterized by defining control signals, $\alpha_i \in \{0,1\}$, $i = 1, \ldots N$ for each of the filter taps. Here, $\alpha_i = 0$ implies that the i^{th} tap has been powered down and $\alpha_i = 1$ implies that it is not powered down. The power dissipation P_D, for this adaptive filter, can be obtained from (2.1) as follows,

$$P_D = \left(\sum_{i=1}^{N} \alpha_i \left(\sum_{j=1}^{M} t_{ij} C_j \right) + T_{oh} C_{oh} \right) V_{dd}^2 f_s, \tag{3.1}$$

where M is the number of the hardware units in each tap, C_j is the average switching capacitance for j^{th} hardware unit in any tap, and C_{oh} is the overhead capacitance not considered in C_j's. Also, t_{ij} is the transition activity in the j^{th} hardware unit in the i^{th} tap, and T_{oh} is the average transition activity for the overhead capacitance C_{oh}. In order to minimize power dissipation it is desirable to power down those taps which maximize the expression $\left(\sum_{j=1}^{M} t_{ij} C_j\right)$.

The product $t_{ij}C_j$ in (3.1) is difficult to estimate at the architectural level. Recent work [8], [15] have addressed this problem. In [8], a B-bit input signal $x(n)$ is modeled as having spatio-temporally uncorrelated bits from the least significant bit (LSB) to bit BP_0. Bits from BP_1 ($BP_1 > BP_0$) to the most significant bit (MSB) are assumed to be perfectly correlated spatially, while their temporal correlation is close to that of $x(n)$ [15]. Bits lying between BP_0 and BP_1 are said to comprise a transition region. Based upon the values of BP_0 and BP_1 in the input signals, capacitive coefficients for arithmetic modules such as adders and multipliers were computed in [8]. In this paper, we will assume that such models are available.

Furthermore, in order to simplify the problem and to come up with practical **SMA** strategies, we will assume throughout this paper that the input signal $x(n)$ is uncorrelated. We will see later that the **SMA** strategies resulting from this assumption are simple enough to be implemented and also result in substantial power savings. It can be shown that the power dissipation of a $B_x \times B_c$ bit multiplier, which multiplies a B_x-bit uncorrelated input $x(n)$ with a B_c-bit coefficient w_k, is given by

$$P_m = B_x \lceil log_2(|w_k|) \rceil C_b V_{dd}^2 f_s, \qquad (3.2)$$

where C_b is the switching capacitance of a primitive block of the array. This observation has been supported via simulation results in [1] obtained via the gate-level simulator MED [20]. Note that the term $B_x\lceil log_2(|w_k|)\rceil$ represents the number of primitive blocks in the multiplier that are needed to perform the multiplication.

B. Algorithm Performance

The mean square error (MSE) can be formulated for the given set of α_i's. The output error of the adaptive filter can be written as,

$$e(n) = d(n) - \sum_{i=1}^{N} \alpha_i w_i x(n-i+1), \qquad (3.3)$$

where w_i and $x(n-i+1)$ is the coefficient and input signal for i^{th} tap. For an uncorrelated/white input $x(n)$, it can be shown that [5] the minimum MSE (J_{min}), is given by,

$$J_{min} = \sigma_d^2 - \sum_{i=1}^{N} \alpha_i |w_i|^2 r(0), \qquad (3.4)$$

where σ_d^2 and $r(0)$ is the power in the desired signal $d(n)$ and input signal $x(n)$, respectively.

C. Joint Optimization

From (3.4), we note that powering down taps with small values of w_k result in a small increase in J_{min} which is desirable. However, from (3.2), we also see that a tap with a small value of w_k consumes lesser power as well and hence powering down such a tap will not provide substantial power savings. Clearly, the power-optimum configuration will be the one that powers down those taps which result in maximal power savings and at the same time results in a J_{min}, which is less than the desired value J_o. This is formally stated as follows,

$$\min_{\alpha_i, i\in\{1,\cdots,N\}} \sum_{i=0}^{N-1} \alpha_i \lceil log_2(|w_i|) \rceil,$$
$$s.t. \sum_{i=0}^{N-1} \alpha_i |w_i|^2 r(0) > \sigma_d^2 - J_o, \qquad (3.5)$$

where $\alpha_i \in \{0,1\}$ and J_o is the desired value of MSE dictated by the application. Note that (3.5) assumes that the multipliers in (2.2(b)) are powered-down after the adaptive filter has converged, an assumption that is usually true in practice. The optimization problem in (3.5) can be solved via standard mixed integer linear programming (MILP) approaches. In the next section, we will present practical **SMA** strategies that approach the solution of (3.5).

If input statistics are ignored, then the objective function in (3.5) reduces to $\sum \alpha_i$. Minimization of $\sum \alpha_i$ is equivalent to powering down the maximum number of taps in the filter subject to the constraint in (3.5).

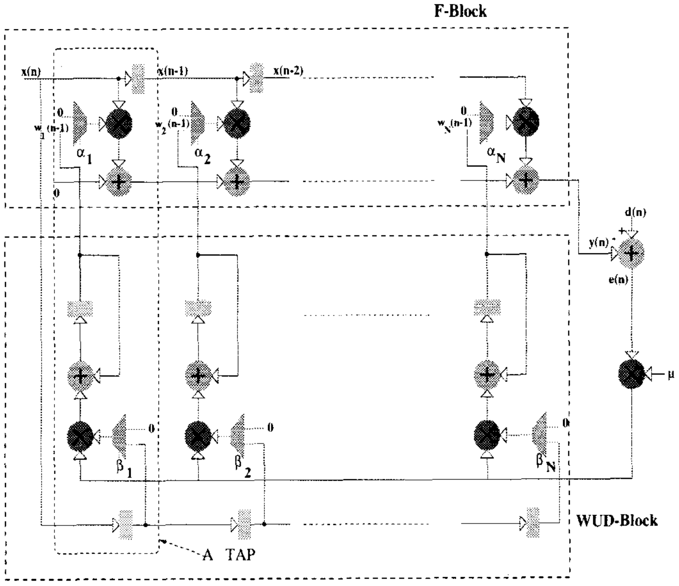

Fig 2. **SPA** Block of the DAT-based Adaptive Filter

IV. DAT-based Low-Power Adaptive Filters

In this section, we present the DAT-based adaptive filter formulated according to Fig. 1. The goal here is to develop **SMA** block strategies to power down the set of taps, which optimizes the power dissipation while maintaining a specified MSE.

A. SMA Strategy 1

In this subsection, we will present an algorithm for dynamically controlling the α_k's, while maintaining $J_{min} < J_o$ and reaching the $min\ P_D$ solution.

SMA Strategy 1

Step 1. *Start with* $\alpha_k = 1,\ \forall k$.

Step 2. *Allow the adaptive filter to converge to the optimum solution. Check* J_{min}*, which is the converged value of the* MSE*,* $E[e^2(n)]$.

Step 3. *If* $J_{min} < J_o$*, goto* **3a** *else goto* **3b**.

3a. *Determine* j *such that* $|w_{o,j}| = min\{|w_{o,k}|, \forall k : \alpha_k = 1\}$. *Assign* $\alpha_j = 0$ *and go to* **Step 2**.

3b. *Determine* j *such that* $|w_{o,j}| = max\{|w_{o,k}|, \forall k : \alpha_k = 0\}$. *Assign* $\alpha_j = 1$ *and go to* **Step 2**. ◇

Therefore, **SMA Strategy 1** *approaches* the power-optimum configuration (obtained as a solution of (3.5)) by assigning $\alpha_k = 0$ starting with coefficients with the lowest magnitude until the $J_{min} < J_o$. Equivalently, it minimizes $\sum \alpha_i$ thus *achieving* the solution of (3.5) if the input signal statistics are not accounted for. We should mention here that in the power-optimum configuration, some of the internal taps may also be powered down leading to non-uniformly spaced samples.

B. SMA Strategy 2

In this subsection, we present an **SMA** strategy that takes into account input signal statistics for calculating power dissipation. Therefore it optimizes objective function in (3.5), and not just $\sum \alpha_i$ as done by **SMA Strategy 1**. This algorithm is based on the idea that for a fixed value of $\sum |w_i|^2$, the expression $\sum \log_2(|w_i|)$ is maximized when all w_i's have the same value. Thus, an intelligent algorithm will be to sort the coefficients, and power down taps whose values are very close to each other. This is done by partitioning the set of powered-up taps into two parts both having all the taps with similar valued coefficients.

SMA Strategy 2

Step 1. *Start with* $\alpha_k = 1,\ \forall k$.

Step 2. *Allow the adaptive filter to converge to the optimum solution. Check* J_{min}*, which is the converged value of the* MSE*,* $E[e^2(n)]$.

Step 3. *If* $J_{min} < J_o$*, goto* **Step 4** *else goto* **Step 5**.

Step 4. *Form* $\Omega = \{w_k, \alpha_k = 1\}$ *and sort it in the increasing order of absolute value of the coefficients,* w_k*'s.*

4a. *Let* $\alpha_k = 0, \forall w_k \in \Omega$*, and find* J_{min} *using (3.4). If* $J_{min} < J_o$*, store the* α *as a possible candidate set for powering down the taps, otherwise let* $\alpha_k = 1, \forall w_k \in \Omega$*. Partition* Ω *into two disjoint sets of equal length* Ω_0 *and* Ω_1 *i.e.* $\Omega = \Omega_0 \bigcup \Omega_1$*, and employ* **4a** *for both* Ω_0 *and* Ω_1.

4b. *Evaluate objective function in (3.5) for each of the solutions in* **4a***, and choose the one which gives minimum value of objective function. Goto* **Step 2**

Step 5. *Power up all the taps i.e. use* $\alpha_k = 1\ \forall k$ *and goto* **Step 2**. ◇

The worst case complexity for one run of this algorithm is $2N - 1$, and is a sub-optimum algorithm. In a practical implementation, we will get the converged solution after a few runs of this algorithm. Whenever, MSE degrades ($J_{min} > J_o$), some taps must be powered up. In this case, to maintain the optimality of the solution, all the taps are first powered up (**Step 5**), and then the optimum tap selection is found. As will be shown by an example in section V(A), the solution via **SMA Strategy 2** approaches the optimum solution (see (3.5)).

To reduce undesirable glitching near the final solution, one can employ a window, δ around J_o rather than a single desired value in **Step 3**.

C. Implementation of DAT-based Adaptive Filter

We present architectural level implementation of the DAT-based adaptive filter derived in last section. Fig. 2 shows the **SPA** block of the DAT-based adaptive filter, where each tap is enclosed in a dotted box and is composed of two multiply-adds. The control signals, α_k's are employed to force a static value of '0' into one of the inputs of the filtering (**F**-block) multipliers in the k^{th} tap if $\alpha_k = 0$. The signals β_k's in the weight-update (**WUD**) block equal zero if either the filter has converged or if the tap is powered down ($\alpha_k = 0$). For array multipliers, if one of the inputs to the multipliers is zero then the switching power consumption of the multiplier is close to zero. Thus, for $\alpha_k = 0$, the **F**-block multiplier in tap k is powered down. Similarly for $\beta_k = 0$, the **WUD**-block multiplier in tap k is powered down and the two inputs to the lower adder are constant. Therefore, the switching activity for this adder will also be zero. If needed, the latch in the weight update block can be powered down by disabling the clock.

V. Simulation Results

In this section, we employ the DAT-based adaptive filter of section IV in three experiments. First, in experiment A, we consider a simple channel and compute the optimal configuration $\alpha_{\mathbf{opt}}$ and the final configuration $\alpha_{\mathbf{final}}$ achieved by the proposed algorithm. Then, we consider a practical communication system scenario. We employ the DAT-based adaptive filter as the near-end crosstalk (NEXT) canceller for 155.52 Mb/s ATM-LAN over category 3 wiring, and discuss the power savings.

A. System Identification

In this experiment, we assume that the DAT-based filter presented in section IV is being used in a system identification scenario. Such a set-up emulates those communications systems, which employ echo cancellers [9] or near-end cross talk (NEXT) cancellers [6]. The system to be identified was a 8^{th} order FIR filter and the proposed adaptive filter had $N = 8$ taps. In addition, we assume a measurement noise of $-60dB$. Further, assume that the desired MSE at the output is, $J_o = -48dB$ and $\delta = -2dB$. Let's assume that the input signal is white with power $r(0) = 1.0$, and $B_x = 8$. It can be shown from the finite precision analysis that $B_c = 8$ bits are sufficient for keeping the quantization noise power, σ_q^2 less than J_o.

We find optimum solution for this case by solving optimization problem (3.5). First, the Wiener solution [5], W_o for the 8-tap adaptive filter is found. These coefficients are then employed in (3.5) along with the desired performance level J_o to determine the optimum set of α_k's. For the given MSE, the optimum solution was obtained via mixed integer linear programming (MILP) technique, and was given by the vector, $\alpha_{\mathbf{opt}} = [\mathbf{01101111}]$. This corresponds to a power dissipation value (see (3.2)) of $288 C_b V_{dd}^2 f_s$ for the **F**-block of the adaptive filter.

Next, the **SMA Strategy 1** was simulated. Fig. 3 shows the learning curve for the adaptive filter. The filter first converges to MSE of $-60dB$. Then, **SMA** is brought up, and taps 1 and 4 are powered down, thus converging to the optimum solution, $\alpha_{\mathbf{opt}}$. Thus, in this case **SMA Strategy 1** was able to achieve the optimum solution.

The **SMA Strategy 2** can also be employed to converge to the optimum solution. Using **SMA Strategy 2** with the same specifications, the filter converges to the final configuration, $\alpha_{\mathbf{final}} = [\mathbf{01101111}]$, which is same as the optimum solution, $\alpha_{\mathbf{opt}}$.

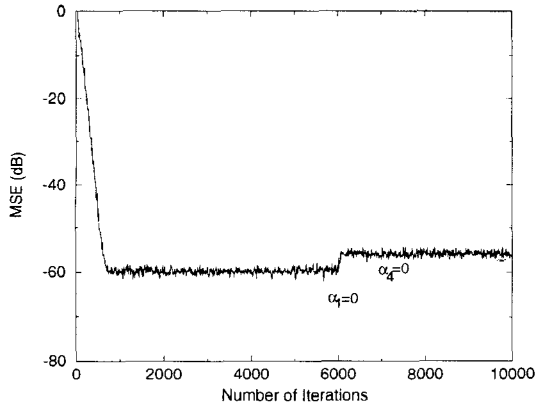

Fig 3. MSE Convergence for System Identification (**SMA Strategy 1**)

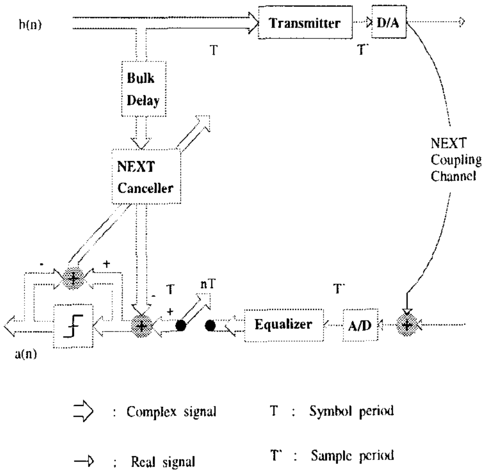

Fig 4. Block Diagram of the ATM-LAN Transceiver

B. NEXT Canceller for 155.52 Mb/s ATM-LAN

In this experiment, we demonstrate the performance of the DAT-based adaptive filter as a NEXT canceller. A block diagram of the system is presented in Fig. 4. The transmitter consists of a 64-CAP (6 bits/symbol) encoder which results in a symbol rate of 155.52/6 = $25.92 Mbaud/s$. The sampling frequency of the D/A and A/D is 3 times the baud-rate of 25.92 $Mbaud$ or 77.76 MHz. At the receiver (see Fig. 4), the received signal is distorted further due to the superimposition of the NEXT signal. This composite signal is processed by the fractionally space linear equalizer (FSLE), which is a pair of adaptive filters. In addition, the local transmitted symbols are passed through a complex adaptive NEXT canceller, which tries to cancel the effect of the NEXT in the received signal. The algorithmic performance measure in this case is signal-to-noise ratio, SNR_o which is equal to the ratio of signal power to MSE at the slicer. For 64-CAP, SNR_o of $29.45dB$ is sufficient to obtain a probability of error of 10^{-10}.

The complexity requirements for the NEXT-canceller increase as the cable length increases. Traditionally, the NEXT-canceller is designed for the worst case scenario i.e., the longest cable length, which in this case is given by EIA/TIA model for $100m$. However, for shorter cable lengths, the complexity requirements can be relaxed so that the power dissipation can be lowered. In this experiment, we illustrate how much power savings result from a DAT based NEXT-canceller due to different cable lengths.

The DAT-based NEXT-canceller was employed for two different cable lengths of $100m$ and $70m$. We specify $J_o = -29.45dB$, $\delta = -1.5dB$, and $N = 32$. The coefficient precision such that $SQNR_o < SNR_o$ was

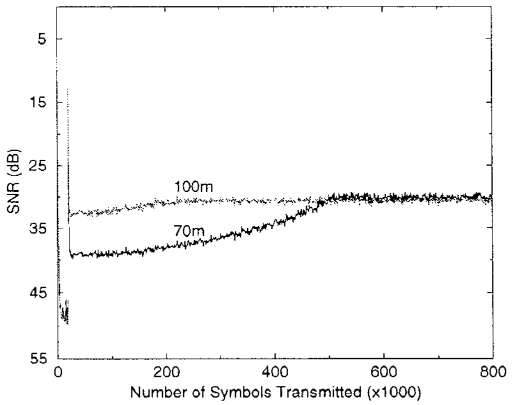

Fig 5. DAT-based NEXT Canceller for 155.52 Mb/s ATM-LAN

calculated to be $B_c = 12$ from the finite precision analysis of the adaptive filter. The convergence plot is shown in Fig. 5 for the cable lengths of $100m$ and $70m$. We average SNR_o over 1024 iterations before plotting Fig. 5. For $100m$ cable length, the NEXT-canceller first converges to an SNR_o of $32.5dB$ as shown in Fig. 5. Clearly, $SNR_o = 32.5dB$ is higher than $30.95dB$. Therefore, we employ the **SMA Strategy 1** to power down some of the taps. The filter converges to $\alpha_{\mathbf{final}} = [\mathbf{01111111111111111111111100001000}]$, thus powering down 8 taps and achieving 21% power savings in the filter block. The optimum solution obtained via MILP is given by $\alpha_{\mathbf{opt}} = [\mathbf{00111111111111111111111000001000}]$, which corresponds to 22% power savings. It is clear that the solution from **SMA Strategy 1** is very close to the optimum solution.

Similarly, for a $70m$ cable, the NEXT-canceller initially converges to an SNR_o of $40dB$, which is greater than the desired value of $30.95dB$. Employing **SMA Strategy 1** results in the NEXT canceller to achieve SNR_o just above $29.45dB$, with $\alpha_{\mathbf{final}} = [\mathbf{00000000111111111000000000000000}]$ as the final configuration. This implies approximately 62% power savings for the **F**-block. The optimum solution for this set of specifications is given by, $\alpha_{\mathbf{opt}} = [\mathbf{00000000111111111100000000000000}]$, which is close to the solution, $\alpha_{\mathbf{final}}$. For this particular experiment, optimum solution ($\alpha_{\mathbf{opt}}$) shows that 22 taps should be powered down as compared to 23 achieved by employing **SMA Strategy 1**. This is due to the noisy values of converged filter coefficients and J_{min} being passed to MILP for finding the optimum solution. Thus, we conclude that a DAT-based NEXT-canceller can result in substantial power reductions.

REFERENCES

[1] S. Bobba, *Power Characterization of Adders, Multipliers and DSP Circuits.* Univ. of Illinois, ECE497NS Project Report, Dec 1996.

[2] A. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," in *Proceedings of the IEEE*, vol. 83, pp. 498–523, April 1995.

[3] A. Chandrakasan *et al.*, "Minimizing power using transformations," *IEEE Trans. Comp.-Aided Design*, vol. 14, no. 1, pp. 12–31, Jan. 1995.

[4] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data driven signal processing : An approach for energy efficient computing," in *Proceedings of International Symposium on Low Power Electronics and Design*, (Monterey, CA), pp. 347–352, August 1996.

[5] S. Haykin, *Adaptive Filter Theory.* Englewood Cliff, NJ: Prentice Hall, 1991.

[6] G. H. Im and J. J. Werner, "Bandwidth-efficient digital transmission up to 155 Mb/s over unshielded twisted-pair wiring," *IEEE Conf. on Comm.*, vol. 3, pp. 1797–1803, May 1993.

[7] S. Iman and M. Pedram, "An approach for multi-level logic optimzation targetting low-power," *IEEE Trans. Comp.-Aided Design*, vol. 15, no. 8, pp. 889–901, Aug. 1996.

[8] P. Landman and J. M. Rabaey, "Architectural power analysis: the dual bit type method," *IEEE Trans. on VLSI Systems*, vol. 3, pp. 173–187, June 1995.

[9] E. A. Lee and D. G. Messerschmitt, *Digital Communication.* Boston, MA: Kluwer Academic Publishers, 1994.

[10] J. T. Ludwig, S. H. Nawab, and A. P. Chandrakasan, "Low-power digital filtering using approximate processing," *IEEE J. of Solid State Circuits*, vol. 31, no. 3, pp. 395–400, March 1996.

[11] C. J. Nicol, P. Larsson, K. Azadet, and J. H. O'Neill, "A low power 128-tap digital adaptive equalizer for broadband modems," in *IEEE International Solid-State Circuits Conference*, pp. 94–95, Feb 1997.

[12] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *Proceedings of the IEEE*, vol. 77, no. 12, pp. 1879–1895, Dec. 1989.

[13] K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters - part II : Pipelined incremental block filtering," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 1118–1134, July 1989.

[14] M. Potkonjak and J. Rabaey, "Fast implementation of recursive programs using transformations," in *Proc. ICASSP*, (San Francisco), pp. 569–572, March 1992.

[15] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "Analytical estimation of transition activity from word-level statistics," in 34^{th} *Design Automation Conference*, June 1997.

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

[17] N. R. Shanbhag and K. K. Parhi, "Relaxed look-ahead pipelined LMS adaptive filters and their application to ADPCM coder," *IEEE Trans. Circuits and Systems*, vol. 40, pp. 753–766, Dec. 1993.

[18] N. Weste and K. Eshraghian, *Principles of CMOS VLSI design : A Systems Perspective.* MA: Addison-Wesley, 1988.

[19] B. Widrow *et al.*, "Stationary and non-stationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 1964, pp. 1151–1162, Aug. 1976.

[20] M. G. Xakellis and F. N. Najm, "Statistical estimation of the switching activity in digital circuits," in *Design Automation Conference*, pp. 728–733, June 1994.