

# Dynamic Algorithm Transforms For Low-power Reconfigurable Adaptive Equalizers

Manish Goel and Naresh R. Shanbhag, *Senior Member, IEEE*

## Abstract

Presented in this paper are low-power reconfigurable adaptive equalizers derived via *dynamic algorithm transforms* (DAT). The principle behind DAT is that conventional signal processing systems are designed for the worst-case and are not energy-optimum on average. Therefore, significant energy savings can be achieved by optimally reconfiguring the hardware in these situations. Practical reconfiguration strategies for adaptive filters are presented. These strategies are derived as a solution to an optimization problem. The optimization problem has energy as the objective function and a constraint on the algorithm performance (specifically the *SNR*). The DAT-based adaptive filter is employed as an equalizer for  $51.84\text{Mb/s}$  very high-speed digital subscriber loop (VDSL) over 24-pair BKMA cable. The channel non-stationarities are due to variations in cable length and number of far-end crosstalk (FEXT) interferers. For this application, the traditional design is based on  $1\text{kft}$  cable length and 11 FEXT interferers. It was found that upto 81% energy savings can be achieved when cable length varies from  $1\text{kft}$  to  $0.1\text{kft}$ , and the number of FEXT interferers varies from 11 to 4. On the average, 53% energy savings are achieved as compared to the conventional worst case design.

## I. Introduction

The power dissipation of CMOS circuits [1, 2] is a grave concern in the VLSI industry. This concern is mainly driven by the limited battery life in mobile applications, reliability as well as packaging costs in both mobile and tethered applications. Several low-power techniques [1] have been proposed for general VLSI as well DSP-specific systems. General low-power techniques include logic minimization [3]-[4] and precomputation [5] (at the logic level); reduced

---

\*The authors are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. E-mail: [mgoel,shanbhag]@uivlsi.csl.uiuc.edu.

†This research was supported by the DARPA contract DABT63-97-C-0025 and NSF CAREER award MIP-9623737.

voltage swing [6] and adiabatic logic [7] (at the circuit level); and CMOS scaling [8] (at the technological level). The low-power techniques specific to DSP systems include strength reduction [9]-[11] and DECOR [12] (at the algorithmic level); and pipelining [13]-[14] and parallel processing [14] (at the architecture level). Moreover, *algorithm transformation techniques* [15] such as *look-ahead* [14], *relaxed look-ahead* [16], *algebraic transformations* [17] and *retiming* [18] have been employed in high-speed and more recently low-power DSP system design. All of the above mentioned techniques are applied during the VLSI design phase and their implementation is time-invariant. Therefore, we refer to this class of low-power methods as *static techniques*.

Recently, *dynamic techniques* both at the circuit level and the algorithmic level have been proposed. Dynamic techniques can be applied after static techniques to obtain even greater energy savings. These techniques are based upon the principle that the input is usually non-stationary and hence, it is better (from an energy perspective) to adapt the algorithm and architecture to the input. Such systems are referred to as *reconfigurable signal processing* systems [19]-[29]. In [19], reconfigurability is employed to map a wide class of signal processing algorithms to an appropriate architectural template. Field programmable gate array (FPGA) based devices and their reconfiguration schemes are discussed in [20]-[22]. Hybrid architectures based on FPGAs and general-purpose DSPs is the topic of research in [23]-[24]. Other dynamic techniques include approximate signal processing [25]-[26], where just the right amount of computational resources, needed at a specific instant/period to meet the algorithm performance requirements, is allocated. In addition to dynamic techniques at circuit and architecture level, techniques at the algorithmic level [27]-[29] are also being developed. The key goal of these techniques is to improve the algorithm performance (such as convergence rate [27], data-rate [28] and image distortion [29]) by exploiting variabilities in the data and channel. Thus, there are several emerging dynamic techniques at the circuit, architecture and algorithmic levels.

Maximum benefits of dynamic techniques can be obtained if the circuit and architecture issues (such as energy) and algorithmic issues (such as  $SNR$ ) are addressed jointly. For this reason, we have recently proposed dynamic algorithm transforms (DAT) [30]-[31] (see Fig. 1) for a joint real-time optimization of energy consumption and  $SNR$  in digital filters. From an implementation perspective, a DAT-based reconfigurable DSP system has the signal processing algorithm (**SPA**) implemented in a reconfigurable hardware (see Fig. 1) (such as FPGA, certain DSPs or ASICs), while the input state and state transitions are monitored

by a signal monitoring algorithm (**SMA**) block (or a controller). In [30], DAT techniques are studied in the context of a system-identification scenario with application to a near-end crosstalk (NEXT) canceller for  $155.52\text{Mb/s}$  ATM-LAN. A general framework for DAT is

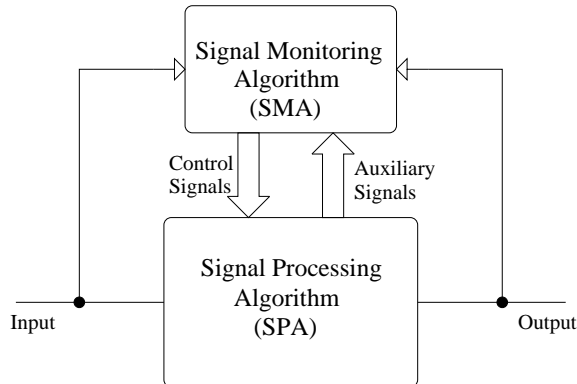


Figure 1: A DAT based reconfigurable signal processing system.

presented in [31], where the variabilities in the input are modeled as transitions in an *input state-space*, and the reconfigurations are modeled as transitions in the *configuration space* of the reconfigurable hardware fabric. In [31], the energy savings are achieved by employing the energy-optimal reconfiguration strategies and energy-optimal partitioning of the **SPA** block into fixed and configurable portions, and the **SMA** block into datapath and memory portions. It has been shown in [30]-[31] that substantial energy savings can be achieved via DAT-based reconfigurations.

The main contribution of this paper is to rigorously derive the energy-optimum reconfiguration strategy for an equalization scenario and demonstrate its performance in  $51.84\text{Mb/s}$  very high-speed digital subscriber loop (VDSL) [32]-[34]. Preliminaries results of this work have appeared in [35], where we extended the strategy in [30] to include the equalization scenario along with a precise energy modeling. Related works include a non-uniformly spaced equalizer [36], where a technique for choosing the best  $K$  taps (in terms of  $MSE$ ) out of total  $N$  taps is presented. This technique however is very complex, and is not suitable for the real-time implementation. In contrast, our reconfiguration strategy is much simpler, and is employed for real-time reconfigurations. In [37], a 128 tap adaptive equalizer was proposed, in which the tap length and precision are varied to maintain a certain  $SNR$ . Our approach is systematic as we employ *Lagrange multiplier method* [38] to find optimum set of powered-up taps, which are not necessarily the end taps. Further, we consider a fractionally-spaced linear equalizer (FSLE) and a complex-valued strength-reduced [11] feedback equalizer.

The rest of the paper is organized as follows. In section II, we present some preliminaries related to the adaptive filter and multiplier energy models. The dynamic algorithm transforms are discussed in section III, and energy-optimum reconfiguration strategy for adaptive filters is presented in section IV. Finally, in section V, we employ the DAT-based equalizer in 51.84Mb/s VDSL transceiver and present simulation results.

## II. Preliminaries

In this section, we present some preliminaries regarding the reconfigurable architecture of the adaptive filter, and energy models for the multipliers. Later sections will employ these energy models to determine the energy-optimum configuration for the adaptive filter architecture.

### A. Reconfigurable Adaptive Filter

Let  $x(n)$  be the input signal to an  $N$ -tap adaptive filter, and  $w_k(n)$ ,  $k = 1, 2, \dots, N$  be the real-valued filter coefficients. The least mean square (LMS) algorithm [39] is then given by,

$$e(n) = d(n) - \sum_{k=1}^N w_k(n-1)x(n-k+1) \quad (2.1)$$

$$w_k(n) = w_k(n-1) + \mu e(n)x(n-k+1), \quad k = 1, 2, \dots, N \quad (2.2)$$

where  $e(n)$  and  $d(n)$  are the output error and the desired output, respectively, and  $\mu$  is the step-size. If the correlation sequence  $r_x(k) = E[x(n)x(n-k)]$  of the input signal is known, then the mean squared error (*MSE*)  $\mathcal{J} = E[e^2(n)]$  is computed as [40],

$$\mathcal{J} = \sigma_d^2 - \sum_{k=1}^N \sum_{j=1}^N w_k w_j r_x(k-j), \quad (2.3)$$

where  $\sigma_d^2 = E[d^2(n)]$  is the desired signal power and  $w_k$  are the optimum filter coefficients.

A direct implementation of the LMS algorithm is shown in Fig. 2(a) where each tap consists of two multipliers and two adders. The filter (**F**) block implements (2.1) and the weight-update (**WUD**) block implements (2.2). The architecture in Fig. 2(a) can be modified to obtain a reconfigurable architecture in Fig. 2(b), where we have introduced additional control signals  $\alpha_k$  and  $\beta_k$  which are employed to power up/down the  $k^{th}$  tap. For example, setting  $\alpha_k = 0$  forces a zero at the input to the **F**-block multiplier of the  $k^{th}$  tap and bypasses the **F**-block adder thereby powering down the  $k^{th}$  tap in the **F**-block. Similarly,  $\beta_k = 0$  powers down the  $k^{th}$  tap in the **WUD**-block.

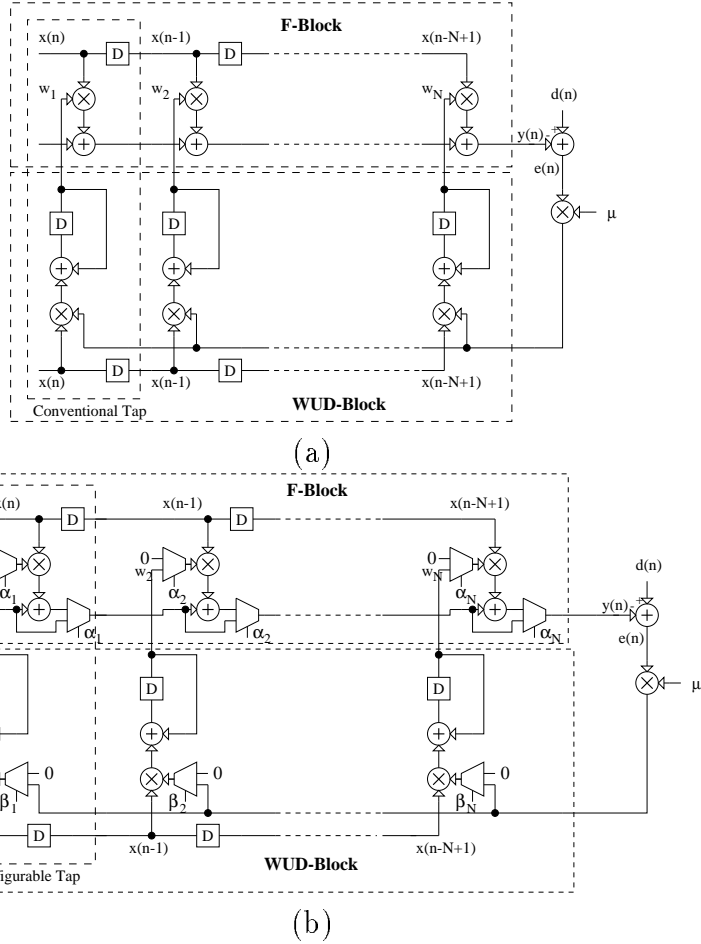


Figure 2: An LMS adaptive filter: (a) fixed architecture and (b) reconfigurable architecture.

Additional energy savings can be achieved by adapting the precisions of the input signal and the coefficients. The input precision  $B_x$  is chosen to achieve a specified signal-to-quantization noise ratio ( $SQNR$ ). It can be shown that,  $SQNR$  at the input is given by,

$$SQNR(dB) = 6B_x + 4.8 - PAR(dB), \quad (2.4)$$

where  $PAR$  is the peak-to-average ratio at the input, and is computed by dividing the maximum value of the input signal with its root-mean squared ( $RMS$ ) value. It can be seen from (2.4) that a  $6dB$  reduction in  $PAR$  results in 1 bit reduction in  $B_x$ .

It can be shown [41] that, to achieve a specified  $SQNR$  at the output, the coefficient precision  $B_w$  is given by,

$$B_w = B_{w,max} + \frac{1}{2} \log_2 \left( \frac{1}{N} \sum_{k=1}^N \alpha_k \right), \quad (2.5)$$

where  $B_{w,max}$  is the maximum value of  $B_w$  (for  $\alpha_k = 1$ ,  $k = 1, 2, \dots, N$ ). Thus, (2.5) indicates

that 1 bit reduction in the precision is achieved for each 4-fold reduction in filter length. In section IV, we will present reconfiguration strategies to choose the parameters  $\alpha_k$  and  $\beta_k$  in an energy-optimum manner. In the next subsection, we present a multiplier energy model that will be employed to derive the reconfiguration strategies presented in section IV.

## B. Multiplier Energy Model

We focus on energy models for the multipliers as these consume a large percentage of the total energy. There is a significant on-going effort [42, 43] in the computer-aided design (CAD) community to find accurate estimates of the energy dissipation. Our interest here is to determine accurate and simple relative power dissipation models which can be employed by the **SMA** block to determine an energy-optimum configuration in real-time. It is well-known that energy dissipation is a function of the input statistics in CMOS circuits.

For a direct-form FIR filter, the input  $x(n - k + 1)$  into the  $k^{\text{th}}$ -tap multiplier is a delayed copy of  $x(n)$ . Thus, the statistics of the data input are the same for all taps. Therefore, we present an energy dissipation model of a multiplier, which is a function of the coefficient input only. We will assume that a  $B_x$ -bit signal  $x(n)$  is being multiplied by a  $B_w$ -bit constant coefficient  $w_k$ . The constant coefficient assumption is valid for adaptive filters if we assume that the **WUD**-block is powered-down after convergence. Assuming a two's complement representation, we have

$$w_k = -w_k^{(0)}2^{B_w} + \sum_{j=1}^{B_w-1} w_k^{(j)}2^{B_w-j}, \quad (2.6)$$

where  $w_k^{(j)}$  is the value of the  $j^{\text{th}}$  bit of coefficient  $w_k$ . In [35], we define  $\mathcal{N}_1(w_k)$  as the number of non-zero bits in the coefficient  $w_k$  given in (2.6). Similarly,  $\mathcal{N}_2(w_k)$  is defined as the difference of  $B_w$  and number of zeros at least significant bit (LSB) positions, and is given as,

$$\mathcal{N}_2(w_k) = B_w - \sum_{j=0}^{B_w-1} \prod_{i=j}^{B_w-1} (1 - w_k^{(i)}). \quad (2.7)$$

It was found via a real-delay gate-level simulations [44] that the linear energy model based on  $\mathcal{N}_1(w_k)$  and  $\mathcal{N}_2(w_k)$ , underestimates and overestimates the energy consumption of the multiplier, respectively. Therefore, a better energy model can be obtained by taking a weighted sum of  $\mathcal{N}_1(w_k)$  and  $\mathcal{N}_2(w_k)$  as follows:

$$\mathcal{E}_m(w_k) = \mathcal{E}_{max} \frac{\eta \mathcal{N}_1(w_k) + (1 - \eta) \mathcal{N}_2(w_k)}{B_w}. \quad (2.8)$$

The constant  $\eta$  was chosen to equal 0.9 in order to minimize the error between  $\mathcal{E}_m(w_k)$  and the real-delay energy values obtained via a gate-level simulation tool MED [44]. Standard cells based on  $0.18\mu m$ ,  $2.5V$  technology are assumed for the real-delay simulations. It was found that the model in (2.8) is accurate with less than 9% estimation error as compared to a real-delay gate-level simulation. Note that, models based on closed form expressions such as (2.8) are useful in computing the energy-optimum configurations.

### III. Dynamic Algorithm Transforms (DAT)

In this section, we present the general framework for dynamic algorithm transforms (DAT). The motivation for DAT is that the conventional signal processing system designed for the worst case is usually not optimum (from energy perspective) for the best and the nominal cases. Hence, significant energy efficiencies can be gained by having a signal monitoring algorithm or the **SMA** block (see Fig. 1) that monitors the input state and then reconfigures the **SPA** block to match the input. This naturally leads to the definition of input state and configuration, which are presented in sections III(A) and III(B), respectively. In section III(C), we show how energy savings can be calculated.

#### A. Input State-Space

We employ the *input state-space* to distinguish between different scenarios at the input. In general, the input state-space depends upon the input non-stationarity and the hardware granularity. For example, the input state-space needs to have more states for a hardware platform with fine granularity of reconfiguration as compared to the one with coarse granularity. The *input state* is formally defined as follows:

**Definition 1 :** *The input state  $\mathbf{s}(n) \in \mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{N_s}\}$  (where  $\mathcal{S}$  is the **input state-space**), at time instant  $n$  is a vector of input-dependent parameters where  $\mathbf{s}(n) = \mathbf{s}_i$  with a probability  $p(\mathbf{s}_i)$ .*

For example, assume that a  $51.84Mb/s$  VDSL network has 100 connections, out of which 80 are approximately  $0.6kft$  from the transmitter and the remaining are distributed equally between  $0.1kft$  and  $1kft$ . Assume further that the equalizer complexities of these three cable lengths are substantially different so as to warrant reconfiguration. In that case, we can define the input state-space  $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3\}$ ,  $p(\mathbf{s}_1) = 0.1$ ,  $p(\mathbf{s}_2) = 0.8$  and  $p(\mathbf{s}_3) = 0.1$ , where  $\mathbf{s}_1$ ,  $\mathbf{s}_2$  and  $\mathbf{s}_3$  are the input states corresponding to the cable length of  $0.1kft$ ,  $0.6kft$

and  $1kft$ , respectively. For this example,  $\mathbf{s}_3$  corresponds to the worst case while  $\mathbf{s}_1$  and  $\mathbf{s}_2$  represent the best and the nominal cases. Proceeding further, we can define each state  $\mathbf{s}_i = [\sigma_{x_i}^2, PAR_i, SNR_{in_i}]$ , where  $\sigma_{x_i}^2$ ,  $PAR_i$  and  $SNR_{in_i}$  are input signal energy, input peak-to-average ratio ( $PAR$ ) and input signal-to-noise ratio, respectively. Thus, we an input state-space with three states ( $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ ) will suffice where each state is a three element vector.

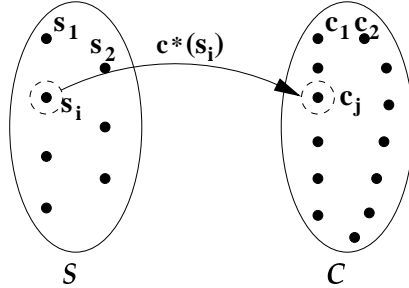


Figure 3: Reconfiguration as a mapping from the input state-space to the configuration-space.

## B. Configuration-Space

A reconfigurable hardware fabric is characterized by its *configuration vector* as defined below:

**Definition 2 :** *The configuration vector  $\mathbf{c}(n) \in \mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{N_c}\}$  (where  $\mathcal{C}$  is the configuration-space) at time instant  $n$  is defined as a vector of reconfiguration control signals. Each configuration vector  $\mathbf{c}_i$  corresponds to a particular value of the control signals.*

For example, for the  $N$ -tap reconfigurable adaptive filter in Fig. 2(b), the configuration vector is defined as:

$$\mathbf{c}(n) = [\alpha_1(n), \dots, \alpha_N(n), \beta_1(n), \dots, \beta_N(n)]$$

where  $\alpha_i(n)$  and  $\beta_i(n)$  are 1-bit control signals indicating whether the filtering and weight-update portions of the  $i^{th}$  tap are powered-up or not. Thus, for  $N = 48$ ,  $\mathbf{c}(n)$  is a 96-bit control signal. Therefore, the configuration space  $\mathcal{C}$  corresponds to  $N_c = 2^{96}$  binary tuples of dimension 96. The above example indicates how easily the number of configurations explodes with increase in reconfigurability options. We are interested in determining the energy-optimum configuration  $\mathbf{c}^*(\mathbf{s}_i)$  for every input state  $\mathbf{s}_i$  from a total of  $N_c$  possible configurations while satisfying a mean squared error ( $MSE$ ) constraint. Due to the large number of possible configurations  $N_c$ , it becomes important to develop a systematic approach to determining the energy-optimum configuration  $\mathbf{c}^*(\mathbf{s}_i)$ . For every state  $\mathbf{s}_i \in \mathcal{S}$ , there exists an energy-optimum configuration  $\mathbf{c}^*(\mathbf{s}_i)$  defined as follows:

**Definition 3** : The energy-optimum configuration  $\mathbf{c}^*(s_i) \in \mathcal{C}$  for a given input state  $s_i \in \mathcal{S}$  is defined as:

$$\begin{aligned} \mathbf{c}^*(s_i) &= \arg \min_{\mathbf{c} \in \mathcal{C}} \mathcal{E}_{SPA}(\mathbf{c}), \\ \text{s.t. } \mathcal{J}_{SPA}(s_i, \mathbf{c}) &\leq \mathcal{J}_o, \end{aligned} \quad (3.1)$$

where  $\mathcal{E}_{SPA}(\mathbf{c})$  is energy dissipated by the **SPA** block in configuration  $\mathbf{c}$ ,  $\mathcal{J}_o$  is the specified MSE and  $\mathcal{J}_{SPA}(s_i, \mathbf{c})$  is the MSE achieved by the **SPA** block when the input is in state  $s_i$  and the **SPA** block is in configuration  $\mathbf{c}$ .

The optimum **SPA** configuration is illustrated in Fig. 3. For each state  $s_i \in \mathcal{S}$ , we need to compute the optimum configuration  $\mathbf{c}_j \in \mathcal{C}$  such that the energy dissipation is minimized while satisfying the constraint on the algorithm performance.

### C. Energy Savings

The average energy savings ( $\mathcal{E}_{sav,ave}$ ) of a DAT-based system as compared to the worst-case design is given as,

$$\mathcal{E}_{sav,ave} = \frac{\mathcal{E}_{WC,ave} - \mathcal{E}_{DAT,ave}}{\mathcal{E}_{WC,ave}} \times 100\%, \quad (3.2)$$

where  $\mathcal{E}_{DAT,ave}$  and  $\mathcal{E}_{WC,ave}$  are the average energy dissipation of the DAT-based system and worst-case design, respectively. Large energy savings can be expected for situations where  $\mathcal{E}_{WC,ave} \gg \mathcal{E}_{DAT,ave}$ . Such situations arise if the energy dissipation requirements for the worst and the nominal cases are considerably different and the probability of occurrence of worst-case input is sufficiently small. Note that,  $\mathcal{E}_{DAT,ave}$  includes the energy of the **SPA** datapath  $\mathcal{E}_{SPA,ave}$  and that of the **SMA** controller  $\mathcal{E}_{SMA,ave}$ . The **SPA** energy consumption  $\mathcal{E}_{SPA,ave}$  can be computed by averaging  $\mathcal{E}_{SPA}(\mathbf{c}^*(s_i))$  over all the states  $s_i \in \mathcal{S}$ . Most of the **SMA** block is activated after  $L$  samples only if there is a transition in the state. Therefore, the energy dissipation of the **SMA** block will be negligible for  $L$  sufficiently large. However, we do include a fixed constant value for the **SMA** energy consumption to reflect the fact that the state monitor in the **SMA** is always active. This was found to be 2% of the worst-case energy consumption for the VDSL application. In next section, we derive energy-optimum reconfiguration strategies for adaptive filters as a solution to (3.1).

## IV. Reconfiguration Strategy For Adaptive Filters

In this section, we employ *Lagrange multiplier Method* to derive an energy-optimum reconfiguration strategy for the adaptive filter architecture in Fig. 2(b). The reconfigurable parameters in this architecture are the control signals  $\alpha_i$ s and  $\beta_i$ s, which indicate the powered-up taps, and precisions  $B_w$  and  $B_x$ . The choice of precisions  $B_w$  and  $B_x$  were presented in section II(A). Once the optimum value of  $\alpha_i$ s are obtained, then the coefficient precision  $B_w$  can be computed via (2.5) and the input precision  $B_x$  can be computed from (2.4). In this section, we present strategies for determining energy-optimal values of  $\alpha_i$ s and  $\beta_i$ s. In section IV(A), we formulate the energy optimization problem and derive the reconfiguration strategy in section IV(B).

### A. Lagrange Formulation

The optimization problem in (3.1) can be rewritten as,

$$\begin{aligned} & \min_{\mathbf{c} \in \mathcal{C}} \mathcal{E}_{SPA}(\mathbf{c}), \\ & s.t. \mathcal{J}_{SPA}(\mathbf{c}) - \mathcal{J}_o \leq 0, \end{aligned} \quad (4.1)$$

where we have dropped the state  $\mathbf{s}_i$  to simplify notation and with the understanding that we will now compute the optimum configuration for a given state. We refer to the optimization problem in (4.1) as a *primal problem*. The Lagrange multiplier method [38] requires the definition of the *Lagrangian function*  $\mathcal{L}(\mathbf{c}, \lambda)$ , which is defined as,

$$\mathcal{L}(\mathbf{c}, \lambda) = \mathcal{E}_{SPA}(\mathbf{c}) + \lambda(\mathcal{J}_{SPA}(\mathbf{c}) - \mathcal{J}_o), \quad (4.2)$$

where  $\lambda \geq 0$  is a real-valued Lagrange multiplier. Now, the problem is to find the pair  $(\mathbf{c}^*, \lambda^*)$  that optimizes  $\mathcal{L}(\mathbf{c}, \lambda)$ . This can be done by defining the *saddle point* [38] as follows.

**Definition 4 :** A **saddle-point**  $(\mathbf{c}^*, \lambda^*)$  of Lagrangian function  $\mathcal{L}(\mathbf{c}, \lambda)$  is defined as one that satisfies the following condition,

$$\mathcal{L}(\mathbf{c}^*, \lambda) \leq \mathcal{L}(\mathbf{c}^*, \lambda^*) \leq \mathcal{L}(\mathbf{c}, \lambda^*) \quad (4.3)$$

for all  $(\mathbf{c}^*, \lambda)$  and  $(\mathbf{c}, \lambda^*)$  sufficiently close to  $(\mathbf{c}^*, \lambda^*)$ .

The above definition implies that a saddle point is a local minimum of  $\mathcal{L}(\mathbf{c}, \lambda)$  in the  $\mathbf{c}$ -space and a local maximum in  $\lambda$ -space. Thus, a natural way for finding saddle point is to descend in the  $\mathbf{c}$ -space and ascend in the  $\lambda$ -space. The Lagrange multiplier  $\lambda$  can also

be viewed as the penalties associated with constraints. Therefore, ascents of  $\mathcal{L}$  in  $\lambda$ -space corresponds to increasing the penalties for the unsatisfied constraints. Similarly, a descent in the  $\mathbf{c}$ -space corresponds to the minimization of the objective function while satisfying the constraints. Based on this understanding, we can redefine the optimization problem in (4.1) as,

$$\max_{\lambda \geq 0} \min_{\substack{\mathbf{c} \in \mathcal{C} \\ \mathcal{J}_{SPA}(\mathbf{c}) \leq \mathcal{J}_o}} \mathcal{L}(\mathbf{c}, \lambda). \quad (4.4)$$

It can be shown [38] that  $\mathbf{c}^*$  in the saddle point  $(\mathbf{c}^*, \lambda^*)$  of (4.2) obtained as solution to (4.4) is also the optimum solution for the primal problem in (4.1). This is also called *Saddle Point Theorem* and the reader is referred to [38] for a detailed proof.

## B. Energy-Optimum Reconfiguration Strategy For Adaptive Filters

We solve (4.4) for adaptive filters under the following assumptions:

1. The weight-update (**WUD**) block in Fig. 2(b) is switched off (i.e.  $\beta_i = 0$ ,  $i = 1, 2, \dots, N$ ) after the filter has converged.
2. The input  $x(n)$  is uncorrelated. In other words, we will assume that the correlation sequence  $r_x(k-j) = E[x(n-k)x(n-j)]$  of the input signal  $x(n)$  is non-zero (and equal to  $\sigma_x^2$ ) only if  $k = j$ .

Under these two assumptions, the energy dissipation of the adaptive filter in Fig. 2(b) is given by,

$$\mathcal{E}_{SPA}(\underline{\alpha}) = \sum_{k=1}^N \alpha_k \mathcal{E}_m(w_k), \quad (4.5)$$

where  $\mathcal{E}_m(w_k)$  is the energy dissipated by a multiplier with coefficient  $w_k$ , and  $\underline{\alpha} = [\alpha_1, \dots, \alpha_N]$  is a vector representation of  $\alpha_i$ s. Note that we have ignored the energy dissipation of the adder in each tap. This is a reasonable assumption since multipliers are the power-hungry blocks in digital filters. We have also employed assumption 1 so that the energy consumption of the **WUD** block can be ignored. Substituting  $\alpha_k w_k$  for  $w_k$  in (2.3) and employing assumption 2 above, we obtain,

$$\mathcal{J}_{SPA}(\underline{\alpha}) = \sigma_d^2 - \sum_{k=1}^N \alpha_k w_k^2 \sigma_x^2, \quad (4.6)$$

where  $\sigma_d^2$ ,  $r_x(k-j)$ ,  $w_k$  and  $\alpha_k$  are defined in section II(A). The Lagrangian function  $\mathcal{L}(\underline{\alpha}, \lambda)$  can now be derived by employing (4.2), (4.5) and (4.6) as follows:

$$\mathcal{L}(\underline{\alpha}, \lambda) = \sum_{k=1}^N \alpha_k \left[ \mathcal{E}_m(w_k) - \lambda |w_k|^2 \sigma_x^2 \right] + \lambda \left( \sigma_d^2 - \mathcal{J}_o \right). \quad (4.7)$$

Employing the *saddle-point theorem* (see (4.3)), we define the Lagrangian optimization problem as,

$$\max_{\lambda \geq 0} \min_{\substack{\alpha \in \{0,1\}^N \\ \mathcal{J}(\underline{\alpha}) \leq \mathcal{J}_o}} \left[ \sum_{k=1}^N \alpha_k \left[ \mathcal{E}_m(w_k) - \lambda |w_k|^2 \sigma_x^2 \right] + \lambda \left( \sigma_d^2 - \mathcal{J}_o \right) \right]. \quad (4.8)$$

We show in Appendix A that the solution to (4.8) is given by:

$$\alpha_{k,opt} = \begin{cases} 1, & \text{if } \frac{|w_k|^2}{\mathcal{E}_m(w_k)} > \tilde{\lambda} \\ 0, & \text{if } \frac{|w_k|^2}{\mathcal{E}_m(w_k)} \leq \tilde{\lambda}, \end{cases} \quad (4.9)$$

where  $\tilde{\lambda} = 1/(\sigma_x^2 \lambda^*)$  is a constant and  $\lambda^*$  is the optimum value of  $\lambda$ . In practice, we do not need to compute the constant  $\lambda^*$  if we employ the reconfiguration strategy of powering down the taps starting with the smallest value of  $|w_k|^2/\mathcal{E}_m(w_k)$  until the *MSE* constraint (see (4.1)) is violated.

The reconfiguration strategy derived from (4.9) indicates that it is better to power down taps with small values of  $|w_k|^2/\mathcal{E}_m(w_k)$ . Intuitively, this makes sense as small values of  $|w_k|^2/\mathcal{E}_m(w_k)$  imply that the  $k^{th}$  tap contributes less to the performance measure (as  $w_k$  is small) but consumes more energy ( $\mathcal{E}_m(w_k)$  is large). Note also that different multiplier models can easily be accommodated by redefining  $\mathcal{E}_m(w_k)$ . Furthermore, if  $\mathcal{E}_m(w_k)$  is assumed to be independent of  $w_k$  then the energy-optimum reconfiguration strategy would be to switch off taps with the smallest coefficients. This is the strategy employed in [25, 37].

The optimum value for  $\beta_k$  ( $k = 1, 2, \dots, N$ ) is chosen as 0 if either  $\alpha_{k,opt} = 0$  or the filter has converged. The justification for this is that we don't need to update the  $k^{th}$  tap if it is not being employed in **F**-block computation. Also, if the filter has converged, then the weight-update (**WUD**) portion of all the taps can be powered down. Thus, we have presented a practical reconfiguration strategy that determines the configuration parameters  $\alpha_{k,opt}$  and  $\beta_{k,opt}$  for an adaptive filter. In the next section, we employ this reconfiguration strategy for 51.84Mb/s very high-speed digital subscriber loop (VDSL).

## V. Application to 51.84Mb/s VDSL

In this section, we employ DAT-based adaptive equalizer for 51.84Mb/s VDSL. First, we present an overview of the VDSL environment and the VDSL transceiver.

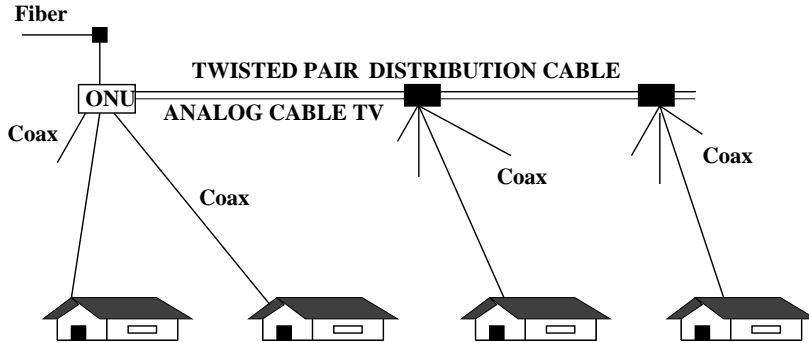


Figure 4: The VDSL environment.

### A. The VDSL Environment

The VDSL application assumes a fiber-to-the-curb (FTTC) [34] network architecture. In this architecture shown in Fig. 4, the optical fiber goes to a curbside pedestal which serves a small number of homes. At the pedestal, the optical signal is converted into an electrical signal and then demultiplexed for delivery to individual homes on copper wiring. These functions are performed in an optical network unit (ONU). The ONU also performs the multiplexing and signal conversion functions required in the opposite direction, i.e. from the homes to the network. In the VDSL system considered here, the downstream channel (from the optical network unit (ONU) to the home) operates at a data rate of 51.84Mb/s. A receiver for this data rate is conventionally designed for 1kft cable length and 11 far-end crosstalk (FEXT) interferers. However, in practice, the cable length and the number of interferers may vary. A DAT-based receiver can exploit these variations to achieve energy savings.

Next, we briefly discuss channel and FEXT characteristics of a BKMA cable, which is employed for twisted pair distribution cable in Fig. 4. The propagation loss of a BKMA cable is similar to that of a category-5 cable specified in the TIA/EIA-568A Standard [45] and is given by,

$$L_P(f) = (6.4\sqrt{f} + 0.091f)d, \quad (5.1)$$

where the propagation loss  $L_P(f)$  is expressed in dB, the frequency  $f$  is expressed in MHz, and  $d$  is the length of the cable in kft. As far as FEXT is concerned, a quantity of interest is

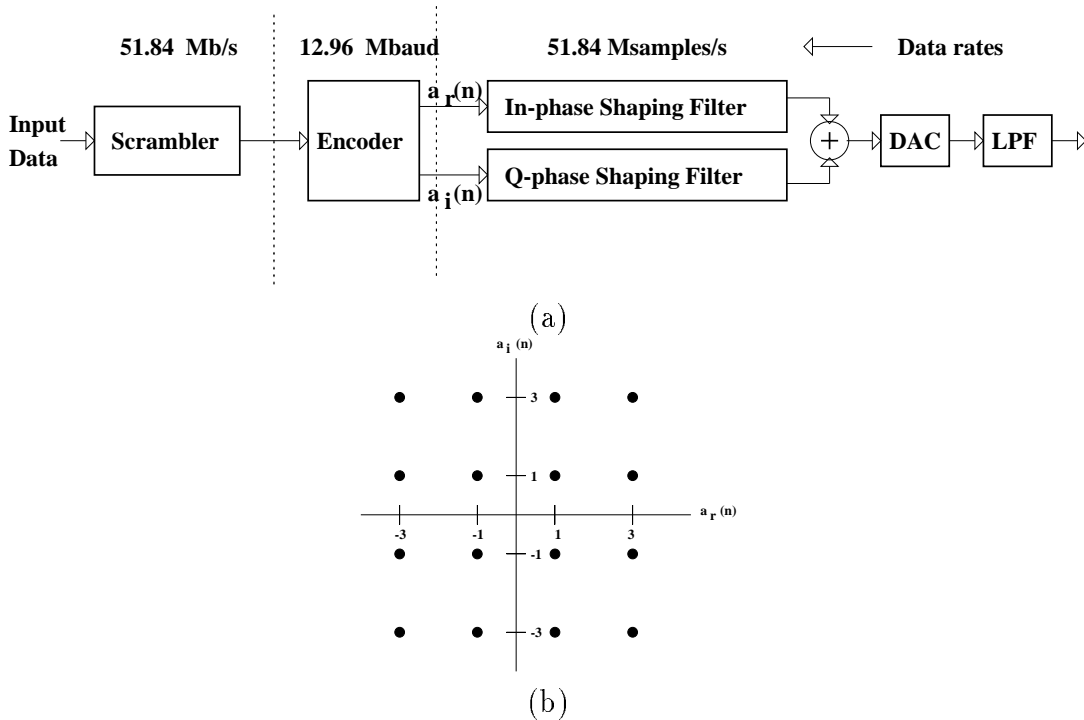


Figure 5: Transmitter for 51.84Mb/s VDSL: (a) block diagram and (b) signal constellation.

the ratio  $V_r^2/V_{f_{ext}}^2$ , where  $V_r$  and  $V_{f_{ext}}$  are the received signal and FEXT signal, respectively. This ratio (also called equal-level FEXT ( $EL - FEXT$ ) loss or the input signal-to-noise ratio  $SNR_i$  in a FEXT dominated environment) can be written as:

$$EL - FEXT = \frac{V_r^2}{V_{f_{ext}}^2} = 40.75 - 20 \log f - 10 \log d + 6 \log m/n, \quad (5.2)$$

where the  $EL - FEXT$  is expressed in  $dB$ , the frequency  $f$  is expressed in  $MHz$ ,  $d$  is the length of the cable in  $kft$ ,  $m$  is the maximum number of crosstalk interferers in the cable, and  $n$  is the number of active crosstalk interferers. We assume  $m = 24$  for this work. The FEXT impairment can be modeled as a Gaussian source because the FEXT sources are independent of each other.

## B. 51.84Mb/s DAT-based VDSL Transceiver

In this subsection, we describe the transmitter and the receiver for 51.84Mb/s VDSL. We will assume that the carrierless amplitude phase (CAP) [46] modulation scheme is being employed. The block diagram of a digital CAP transmitter is shown in Fig. 5(a). The bit stream to be transmitted is first passed through a scrambler. The scrambled bits are then fed into an encoder, which maps blocks of 4 bits onto one of 16 different complex

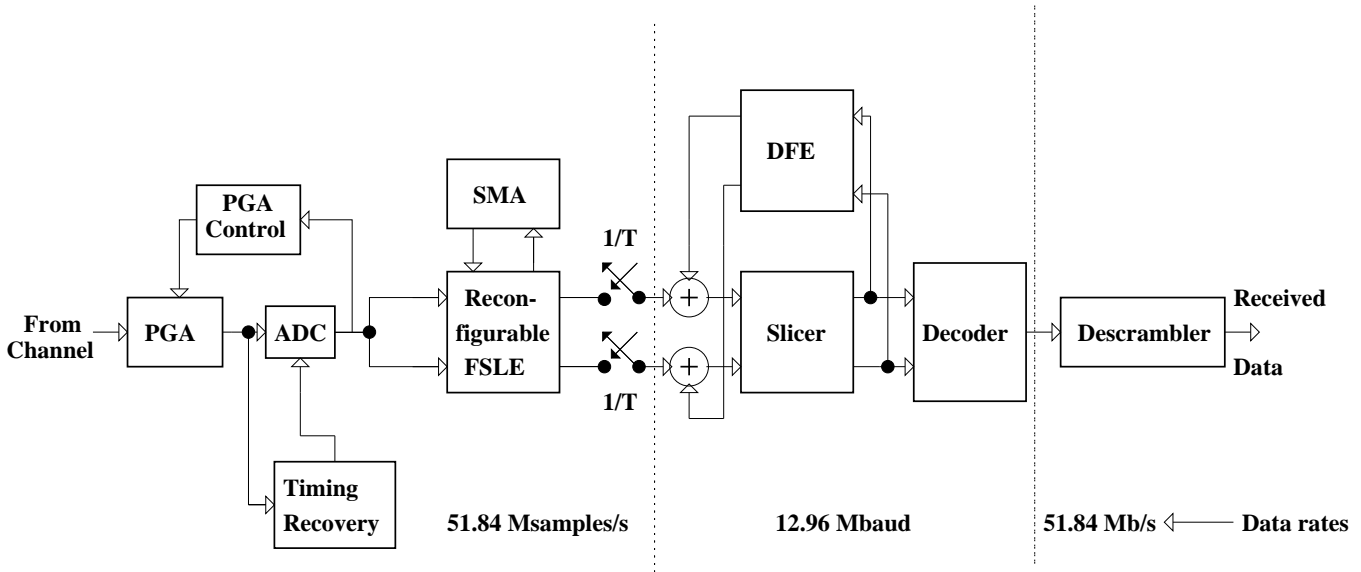


Figure 6: DAT-based receiver for 51.84 Mb/s VDSL.

symbols (see Fig. 5(b))  $a(n) = a_r(n) + ja_i(n)$  corresponding to 16-CAP (4 bits/symbol) line code. The symbols  $a_r(n)$  and  $a_i(n)$  are processed by digital shaping filters. The shaping filter impulse response is specified by a square-root raised cosine pulse with center frequency  $f_c = 12.96 MHz$  and excess bandwidth  $\alpha = 38\%$ . This requires that the shaping filters be operated at a sampling frequency  $f_s$ , which is at least twice the maximum frequency component of the transmit spectrum. We choose  $f_s = 51.84 MHz$  here. The outputs of the filters are subtracted and the result is passed through a digital-to-analog (D/A) converter operating at 51.84 MHz.

On the receiver (see Fig. 6), analog signal is first amplified by a programmable gain amplifier (PGA). The gain of PGA is controlled by a digital PGA control block. The output of PGA is passed to an A/D operating at 51.84 MHz, which converts the analog signal to the digital signal. The sampling instant of the A/D is controlled by a timing recovery block. The digital output of the A/D is processed by a decision feedback equalizer (DFE). The output of the DFE is passed through a 16-CAP slicer to obtain the output symbols. The DFE consists of the two filters - a feedforward filter and a feedback filter. The feedforward filter is a fractionally-spaced linear equalizer (FSLE), which is a pair of 48-tap adaptive filters. The feedback filter is a complex 10-tap adaptive filter operating at symbol rate. A low-power strength-reduced architecture proposed in [11] is employed for implementing the complex adaptive filter. As the precision requirements and the number of taps in the feedback filter

are much smaller than those in the feedforward filter, we apply DAT only to the feedforward filter.

The complexity of the DFE is reduced by simplifying the adaptation algorithm. The equalizer is blindly adapted by employing reduced constellation algorithm (RCA) [47]. In this algorithm, the adaptive filter first converges to a coarse solution based on a 4-CAP constellation (in place of a 16-point constellation). After the convergence with a 4-CAP constellation, the filter is adapted with a 16-CAP constellation. We also employ *powers-of-two approximations* [48] of the output error for the update of the coefficients. This simplification along with powers-of-two step-sizes allows the replacement of the multipliers in the weight-update block with shifters.

The equalizer output is passed through the slicer, decoder and descrambler to retrieve the  $51.84\text{Mb/s}$  data. The algorithmic performance measure in this case is the  $SNR$  at the slicer ( $SNR_o$ ), which is equal to the ratio of signal constellation power (which equals 10 for 16-CAP) to the  $MSE$  across the slicer. For 16-CAP, an  $SNR_o = 21.5\text{dB}$  is sufficient to obtain a probability of error less than  $10^{-7}$ . To reduce undesirable glitching, we employ a window of  $\delta = 2\text{dB}$  around  $SNR_o = 21.5\text{dB}$ . This implies that if  $SNR_o$  lies between  $21.5\text{dB}$  and  $23.5\text{dB}$ , then no reconfiguration takes place. From [33], we obtain the parameters of the worst-case design as a  $1\text{kft}$  cable length and 11 FEXT interferers. In order to achieve  $SNR_o = 21.5\text{ dB}$ , the requirements for the feedforward filter are: number of filter taps  $N = 48$ , coefficient precision  $B_w = 10$  bits, and data precision  $B_x = 8$  bits. Similarly, the requirements for the feedback filter are: number of filter taps  $N = 10$ , coefficient precision  $B_w = 8$  bits, and data precision  $B_x = 3$  bits.

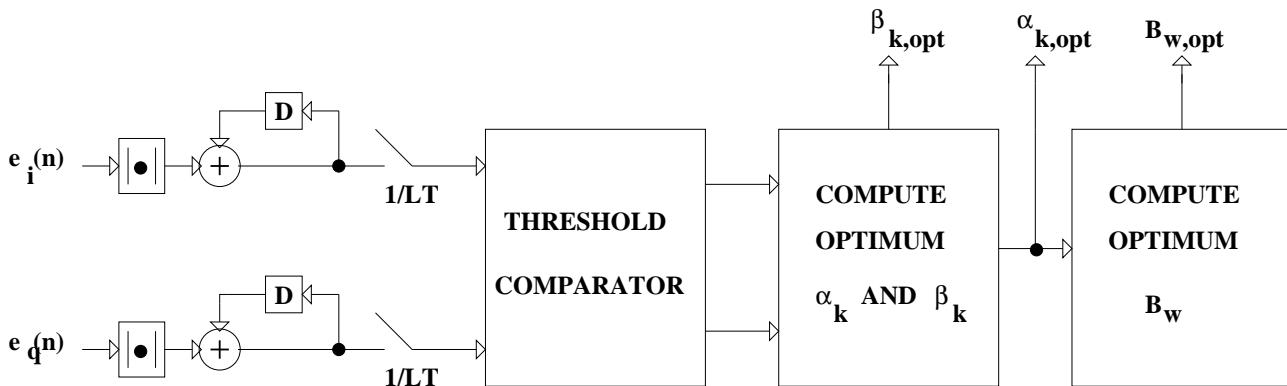


Figure 7: **SMA** block for the DAT-based  $51.84\text{Mb/s}$  VDSL receiver.

We assume that cable length can vary from  $1\text{kft}$  to  $0.1\text{kft}$  in steps of  $0.1\text{kft}$ . Similarly,

the number of FEXT interferers can be 11, 7 or 4. These variations define the state space  $\mathcal{S}$  with 30 states,

$$\mathcal{S} = \{(1kft, 11 - FEXT), (1kft, 7 - FEXT), (1kft, 4 - FEXT), \dots, (0.1kft, 4 - FEXT)\}. \quad (5.3)$$

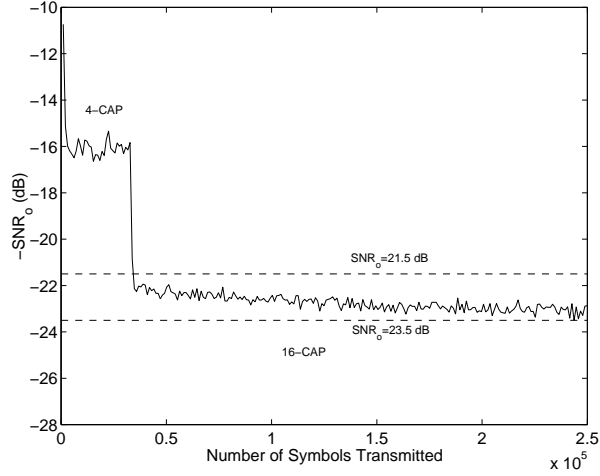
The exact probability distribution of the states requires a survey of the installation of the VDSL network. Since this information is not known at the present moment, we assume that states in (5.3) have a Gaussian distribution with  $0.6kft$  and 4-FEXT as the mean values (nominal case) and standard deviation of  $0.2kft$  and 3-FEXT.

The **SMA** block detects transitions in the input states by monitoring  $SNR_o$  as a function of the cable length or the number of FEXT interferers. The optimum **SPA** configuration is then computed via the reconfiguration strategy presented in section IV(B). The variation in  $SNR_o$  can be detected by observing  $E[|e_i(n)|]$  and  $E[|e_q(n)|]$ , where  $e_i(n)$  and  $e_q(n)$  are errors across in-phase slicer and quadrature-phase slicer, respectively. As shown in Fig. 7,  $E[|e_i(n)|]$  and  $E[|e_q(n)|]$  are computed by summing  $|e_i(n)|$  and  $|e_q(n)|$  over  $L$  symbols with  $L$  being chosen to be 4096 for this experiment. All the blocks before and including threshold comparator are always powered-up. The subblocks after the threshold comparator in Fig. 7 compute the energy-optimum value of  $\alpha_{ks}$ ,  $\beta_{ks}$  and  $B_w$  only when a state transition occurs.

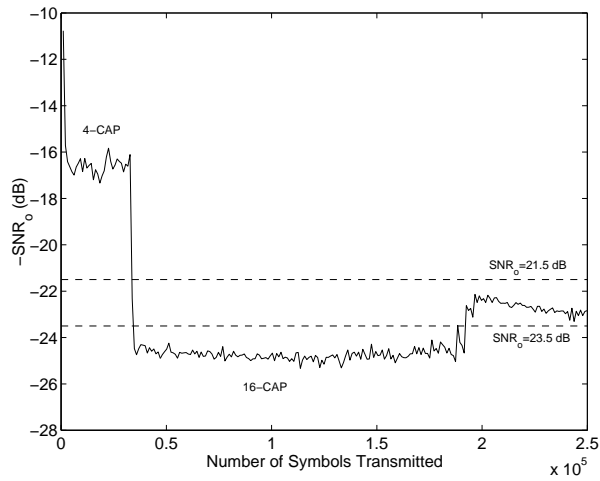
### C. Simulation Results

In this subsection, we present simulation results for  $51.84Mb/s$  VDSL in terms of converged configurations,  $SNR_o$  and energy consumption. The  $SNR_o$  is computed as a ratio of the signal constellation power (which is 10 for 16-CAP) and  $MSE$  across the 16-CAP slicer. Multiplier energy consumption  $\mathcal{E}_m(w_k)$  is obtained via a real-delay gate-level simulator MED [44] assuming a  $0.18\mu m$ ,  $2.5V$  CMOS technology. The energy consumption of the equalizer is then obtained by summing the energy values of the powered-up multipliers in the **F** block. We assume that all multipliers are powered-up in the worst-case design. The energy consumption of the **SMA** block is due to the blocks upto the threshold comparator in Fig. 7, and is assumed fixed at  $0.061mW/MHz$ . The energy savings for a DAT-based design are computed via (3.2).

In Fig. 8, we show  $SNR_o$  convergence curves for a DAT-based VDSL equalizer for two input states. Fig. 8(a) shows convergence curve for the worst-case input state corresponding to a  $1kft$  cable length and 11 FEXT interferers. Recall that the equalizer is adapted blindly



(a)



(b)

Figure 8:  $SNR_o$  convergence plots for the DAT-based 51.84 Mb/s VDSL receiver: (a) 1 kft cable length and 11-FEXT and (b) 1 kft cable length and 4-FEXT.

via reduced constellation algorithm (RCA). In the 4-CAP constellation mode, the equalizer converges to an  $SNR_o$  of 16 dB after 32768 symbols, after which it is switched to a 16-CAP constellation mode. Finally, the equalizer converges to an  $SNR_o$  of 22.5 dB, which is within the  $SNR_o$  constraint window of [21.5 dB, 23.5 dB]. Now, consider the situation where the cable length is 1 kft and the number of FEXT interferers is 4. The  $SNR_o$  convergence curve is plotted in Fig. 10(b). The adaptive filter converges to an  $SNR_o$  of 24.5 dB in this case, thereby falling outside the desired  $SNR$  window [21.5 dB, 23.5 dB]. The **SMA** block detects the surplus  $SNR_o$  and activates the reconfiguration strategy after 131072 symbols in Fig. 8(b). This is done by powering down taps one at a time starting with the one with the smallest value of  $[w_k^2/\mathcal{E}_m(w_k)]$  until the final  $SNR_o$  lies between 21.5–23.5 dB. The final configuration vector of

the in-phase adaptive filter is  $\underline{\alpha} = [00000000000000000000000011111100011100010101011101]$ , where a ‘1’ indicates a tap that is powered up and ‘0’ indicates that it is powered down. It can be seen that, only 16 (out of 48) taps are powered up. Unlike [25, 37], taps other than those at the end are also powered down. Similarly, the final configuration vector for the quadrature-phase filter  $\underline{\alpha} = [000000000000000000000000111111100101100101010101010]$ , which corresponds to 17 powered-up taps. In this configuration, an  $SNR_o$  of 22.2 dB is achieved. Overall, the filter dissipates 63% less energy as compared to the worst-case design.

Table 1 shows the converged configurations for in-phase and quadrature-phase filters. It can be seen that number of powered-up taps (i.e.  $\sum \alpha_{k,opt}$ ) decreases as the cable length decreases. Similarly, the number of powered-up taps reduces with reduction in the number of FEXT interferers for a fixed cable length. From Table 1, and consistent with (2.5), we observe that  $B_w$  reduces by 1 bit for shorter cable lengths. For example, the number of powered-up taps in the in-phase filter ranges from 9 to 48, and  $B_w$  ranges from 9 to 10 for cable lengths ranging from 100 ft to 1000ft, and number of FEXT interferers fixed at 11. The number of powered-up taps in the in-phase filter range from 16 to 48 when the number of FEXT interferers varies from 4 to 11, and the cable length is fixed at 1kft. Similar observations can be made for the number of powered-up taps and precision of the quadrature-phase filter.

Table 2 shows the converged  $SNR_o$  for equalizers based on both the worst-case design and DAT-based design. It can be seen that the  $SNR_o$  for the worst-case design increases from 21.3dB to 28.8dB as the cable length decreases from 1kft to 0.1kft and the number of FEXT interferers is fixed at 11. The  $SNR_o$  ranges from 21.3dB to 23.1dB as the number of FEXT interferers reduces from 4 to 11 with the cable length fixed at 1kft. Overall, the DAT-based receiver maintains the  $SNR_o$  within the range of 21.5dB to 23.5dB for all the states.

Table 3 compares the energy consumption of the worst-case and DAT-based designs. The energy consumption of the worst-case design varies from 2.8mW/MHz to 3.5mW/MHz even though the configuration is fixed. This variation reflects the variation of the energy consumption on input data. The average energy consumption of the worst-case design was found to be 3.2mW/MHz. The energy consumption of the DAT-based design varies from 0.6mW/MHz to 3.6mW/MHz. Of this, the energy of the **SPA** block varies from 0.5mW/MHz to 3.5mW/MHz, while that of the **SMA** block is 0.06mW/MHz. Employing (3.2), we find that the energy savings range from -2% to 81% with an average of 53% assuming a Gaussian distribution for the input states. Thus, it can be seen that the DAT-based

approach is quite attractive from the viewpoint of energy savings for the VDSL application.

## VI. Conclusions and Future Work

In this paper, we have presented dynamic algorithm transforms (DAT) as a systematic method for designing low-power reconfigurable DSP systems. In particular, we employed DAT in the equalization scenario for a 51.84Mb/s VDSL transceiver. Substantial energy savings are observed due to variations in the cable length and the number of FEXT interferers. DAT techniques jointly optimize system performance and energy dissipation thus representing a growing trend to synergize across the design hierarchy. In addition, DAT provides a convenient framework within which on-going research in the areas of data-adaptive DSP algorithms and reconfigurable circuits and architectures can be synergistically combined to enable the design of energy-efficient reconfigurable DSP systems.

Dynamic algorithm transforms have a broad range of applicability which include: developing low-energy software for programmable DSPs, determining energy-optimal reconfiguration strategies for FPGAs, design of low-power wireless transceivers, lattice-based adaptive equalizers, forward error-correction (FEC) codecs and computer-aided design (CAD) tools that enable the design of complex reconfigurable DSP and communication systems.

## Appendix A Derivation of (4.9)

Consider the optimization problem in (4.8),

$$\max_{\lambda \geq 0} \min_{\substack{\alpha \in \{0,1\}^N \\ \mathcal{J}(\underline{\alpha}) \leq \mathcal{J}_o}} \mathcal{L}(\underline{\alpha}, \lambda) = \left[ \sum_{k=1}^N \alpha_k \left[ \mathcal{E}_m(w_k) - \lambda |w_k|^2 \sigma_x^2 \right] + \lambda \left( \sigma_d^2 - \mathcal{J}_o \right) \right]. \quad (\text{A.1})$$

We will solve (A.1) by first determining the solution  $\alpha_k^*$  of the inner optimization problem,

$$\min_{\substack{\alpha \in \{0,1\}^N \\ \mathcal{J}(\underline{\alpha}) \leq \mathcal{J}_o}} \sum_{k=1}^N \alpha_k \left[ \mathcal{E}_m(w_k) - \lambda |w_k|^2 \sigma_x^2 \right], \quad (\text{A.2})$$

as a function of  $\lambda$  and then proving that the outer optimization problem

$$\max_{\lambda \geq 0} \left[ q(\lambda) \triangleq \min_{\substack{\alpha \in \{0,1\}^N \\ \mathcal{J}(\underline{\alpha}) \leq \mathcal{J}_o}} \mathcal{L}(\underline{\alpha}, \lambda) \right], \quad (\text{A.3})$$

has a solution  $\lambda^*$ . The optimum value  $\alpha_{k,opt}$  can then be obtained by the substitution  $\lambda = \lambda^*$  in the solution to (A.2). The solution to (A.2) is derived via the following theorem.

**Theorem 1:** *For the following optimization problem,*

$$\min_{\underline{\alpha} \in \{0,1\}^N} \sum_{k=1}^N \alpha_k \left[ \mathcal{E}_m(w_k) - \lambda |w_k|^2 \sigma_x^2 \right], \quad (\text{A.4})$$

the optimum value of  $\alpha_k$  is given by

$$\alpha_k^*(\lambda) = \begin{cases} 1, & \frac{|w_k|^2}{\mathcal{E}_m(w_k)} > \tilde{\lambda} \\ 0, & \frac{|w_k|^2}{\mathcal{E}_m(w_k)} \leq \tilde{\lambda} \end{cases} \quad (\text{A.5})$$

where  $\tilde{\lambda} = 1/(\sigma_x^2 \lambda)$ .

**Proof :** Define function  $f_k(\lambda)$  as follows,

$$f_k(\lambda) = \mathcal{E}_m(w_k) - \lambda |w_k|^2 \sigma_x^2, \quad k = 1, 2, \dots, N. \quad (\text{A.6})$$

Then, the objective function in (A.4) can be written as,

$$\begin{aligned} \mathcal{F}(\underline{\alpha}, \lambda) &= \sum_{k=1}^N \alpha_k f_k(\lambda) \\ &= \alpha_1 f_1(\lambda) + \alpha_2 f_2(\lambda) + \dots + \alpha_N f_N(\lambda), \end{aligned} \quad (\text{A.7})$$

where  $f_k(\lambda)$  is as defined in (A.6). From (A.7), it is clear that the terms for different indices  $k$  are decoupled from each other. Therefore, for a particular index  $k$ , the optimum  $\alpha_k$  can be obtained by solving the following optimization problem,

$$\min_{\alpha_k \in \{0,1\}} \alpha_k f_k(\lambda), \quad (\text{A.8})$$

where we want to find the optimum value of  $\alpha_k$  and  $f_k(\lambda)$  is a term independent of  $\alpha_k$ . The optimum value  $\alpha_k^*(\lambda)$  is obtained as follows.

**Case 1:** ( $f_k(\lambda) \leq 0$ )

$$\begin{aligned} f_k(\lambda) &\leq 0 \\ \Rightarrow \alpha_k f_k(\lambda)|_{\alpha_k=1} &\leq \alpha_k f_k(\lambda)|_{\alpha_k=0} \\ \Rightarrow \alpha_k^*(\lambda) &= 1. \end{aligned}$$

**Case 2:** ( $f_k(\lambda) \geq 0$ )

$$\begin{aligned} f_k(\lambda) &\geq 0 \\ \Rightarrow \alpha_k f_k(\lambda)|_{\alpha_k=1} &\geq \alpha_k f_k(\lambda)|_{\alpha_k=0} \\ \Rightarrow \alpha_k^*(\lambda) &= 0. \end{aligned}$$

Combining **Case 1** and **Case 2** above, we get the following solution,

$$\alpha_k^*(\lambda) = \begin{cases} 1, & f_k(\lambda) < 0 \\ 0, & f_k(\lambda) \geq 0. \end{cases} \quad (\text{A.9})$$

Substituting  $f_k(\lambda)$  from (A.6) into (A.9) and rearranging we obtain,

$$\alpha_k^*(\lambda) = \begin{cases} 1, & \frac{|w_k|^2}{\mathcal{E}_m(w_k)} > \frac{1}{\sigma_x^2 \lambda} \\ 0, & \frac{|w_k|^2}{\mathcal{E}_m(w_k)} \leq \frac{1}{\sigma_x^2 \lambda}, \end{cases} \quad (\text{A.10})$$

which is identical to (A.5). ♣

Next, we prove that the optimization problem (A.3) has a well-defined maximum point  $\lambda^*$ .

**Theorem 2:**  $q(\lambda)$  defined in (A.3) is a concave function.

**Proof:** For any  $\underline{\alpha}$ ,  $\lambda_1$ ,  $\lambda_2$ , and  $\epsilon \in [0, 1]$ , we have from (A.1),

$$\mathcal{L}(\underline{\alpha}, \epsilon\lambda_1 + (1 - \epsilon)\lambda_2) = \epsilon\mathcal{L}(\underline{\alpha}, \lambda_1) + (1 - \epsilon)\mathcal{L}(\underline{\alpha}, \lambda_2). \quad (\text{A.11})$$

Taking the minimum over  $\alpha_k \in \{0, 1\}$  of both sides in (A.11) we obtain,

$$\min_{\alpha_k \in \{0, 1\}} \mathcal{L}(\underline{\alpha}, \epsilon\lambda_1 + (1 - \epsilon)\lambda_2) \geq \epsilon \min_{\alpha_k \in \{0, 1\}} \mathcal{L}(\underline{\alpha}, \lambda_1) + (1 - \epsilon) \min_{\alpha_k \in \{0, 1\}} \mathcal{L}(\underline{\alpha}, \lambda_2), \quad (\text{A.12})$$

which leads to the following,

$$q(\epsilon\lambda_1 + (1 - \epsilon)\lambda_2) \geq \epsilon q(\lambda_1) + (1 - \epsilon)q(\lambda_2). \quad (\text{A.13})$$

Hence,  $q(\lambda)$  is a concave function of  $\lambda$ . ♣

Thus, we have shown that  $q(\lambda)$  is concave and hence, it has a well-defined maxima  $\lambda^*$ . Therefore,  $\alpha_{k,opt}$  can be obtained by substituting  $\lambda = \lambda^*$  in (A.5) as follows:

$$\alpha_{k,opt} = \begin{cases} 1, & \text{if } \frac{|w_k|^2}{\mathcal{E}_m(w_k)} > \tilde{\lambda} \\ 0, & \text{if } \frac{|w_k|^2}{\mathcal{E}_m(w_k)} \leq \tilde{\lambda}, \end{cases} \quad (\text{A.14})$$

which is identical to (4.9).

## References

- [1] A. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proceedings of the IEEE*, vol. 83, no. 4, pp. 498–523, Apr. 1995.
- [2] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in *1994 IEEE Symposium on Low Power Electronics*, pp. 8–11, Oct. 1994.

- [3] A. Shen, A. Ghosh, S. Devdas, and K. Keutzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks," in *IEEE Int. Conf. on Computer-Aided Design*, pp. 402–407, 1992.
- [4] S. Iman and M. Pedram, "An approach for multi-level logic optimization targeting low-power," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 8, pp. 889–901, Aug. 1996.
- [5] M. Alidina, J. Monterio, S. Devdas, A. Ghosh, and M. Papaefthymiou, "Precomputation-based sequence logic optimization for low-power," *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp. 398–407, Dec. 1994.
- [6] Y. Nakagome *et al.*, "Sub-1V swing internal bus architecture for future low-power ULSI's," *IEEE J. of Solid State Circuits*, vol. 28, no. 4, pp. 414–419, Apr. 1993.
- [7] W. C. Athas, L. J. Svensson, J. G. Koller, N. Tzartzanis, and E. Y.-C. Chou, "Low-power digital systems based on adiabatic switching principles," *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp. 398–407, Dec. 1994.
- [8] B. Davari, R. H. Dennard, and G. G. Shahidi, "CMOS scaling for high-performance and low-power - the next ten years," *Proceedings of the IEEE*, vol. 83, pp. 595–606, Apr. 1995.
- [9] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Brodersen, "Minimizing power using transformations," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 1, pp. 12–31, Jan. 1995.
- [10] D. A. Parker and K. K. Parhi, "Low-area/power parallel FIR digital filter implementations," *Journal of VLSI Signal Processing*, vol. 17, pp. 75–92, Sept. 1997.
- [11] 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.
- [12] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "Decorrelating transformations for low-power adaptive filters," in *International Symposium on Low Power Electronics and Design*, (Monterey, CA), pp. 250–255, Aug. 1998.

- [13] H. H. Loomis and B. Sinha, "High speed recursive digital filter realization," *Circuit, System and Signal Processing*, vol. 3, no. 3, pp. 267–294, 1984.
- [14] K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters - part I & II," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 37, pp. 1099–1134, July 1989.
- [15] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *Proceedings of the IEEE*, vol. 77, no. 12, pp. 1879–1895, Dec. 1989.
- [16] N. R. Shanbhag and K. K. Parhi, *Pipelined Adaptive Digital Filters*. Boston, MA: Kluwer Academic Publishers, 1994.
- [17] M. Potkonjak and J. Rabaey, "Fast implementation of recursive programs using transformations," in *Proc. ICASSP*, (San Francisco), pp. 569–572, Mar. 1992.
- [18] C. Leiserson and J. Saxe, "Optimizing synchronous systems," *J. of VLSI and Comp. Systems*, vol. 1, pp. 41–67, 1983.
- [19] A. Abnous and J. M. Rabaey, "Ultra-low-power domain-specific multimedia processors," in *Proceedings of IEEE VLSI Signal Processing Workshop*, Oct. 1996.
- [20] J. Villasenor and B. Hutchings, "The flexibility of configurable computing," *IEEE Signal Processing Magazine*, pp. 67–84, Sept. 1998.
- [21] S. Trimberger, D. Carberry, A. Johnson, and J. Wong, "A time-multiplexed FPGA," in *Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines* (J. Arnold and K. L. Pocek, eds.), (Napa, CA), pp. 22–28, Apr. 1997.
- [22] N. Shirazi, P. M. Athanas, and A. L. Abbott, "Implementation of a 2-D fast fourier transform on an FPGA-based custom computing machine," in *Field-Programmable Logic and Applications. 5th International Workshop on Field-Programmable Logic and Applications* (W. Moore and W. Luk, eds.), (Oxford, UK), pp. 282–292, Springer-Verlag, Sept. 1995.
- [23] J. R. Hauser and J. Wawrzynek, "GARP: A MIPS processor with a reconfigurable coprocessor," in *Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines* (J. Arnold and K. L. Pocek, eds.), (Napa, CA), pp. 12–21, Apr. 1997.

- [24] C. Rupp, M. Landguth, T. Garverick, E. Gomersall, H. Holt, J. Arnold, and M. Gokhale, "The NAPA adaptive processing architecture," in *IEEE Symposium on Field-Programmable Custom Computing Machines*, (Napa, CA), Apr. 1998.
- [25] 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, Mar. 1996.
- [26] V. Gutnik and A. P. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Trans. on VLSI Systems*, vol. 12, pp. 425–435, Dec. 1997.
- [27] K. Wesolowski, "Adaptive blind equalizers with automatically controlled parameters," *IEEE Trans. Comm.*, vol. 43, no. 2/3/4, pp. 170–172, Feb. 1995.
- [28] B. Vucetic, "An adaptive coding scheme for time-varying channels," *IEEE Trans. Comm.*, vol. 39, no. 5, pp. 653–663, May 1991.
- [29] S. Appadwedula, D. L. Jones, K. Ramchandran, and L. Qian, "Joint source channel matching for wireless image transmission," in *Proc. of International Conference on Image Processing*, (Chicago, IL), Oct. 1998.
- [30] M. Goel and N. R. Shanbhag, "Dynamic algorithm transformations (DAT) for low-power adaptive signal processing," in *International Symposium on Low Power Electronics and Design (ISLPED)*, (Monterey, CA), pp. 161–166, Aug. 1997.
- [31] M. Goel and N. R. Shanbhag, "Low-power reconfigurable signal processing via dynamic algorithm transformations (DAT)," in *Proc. ICASSP*, vol. 4, (Seattle, WA), pp. 3081–3084, May 1998.
- [32] P. S. Chow, J. C. Tu, and J. M. Cioffi, "Performance evaluation of a multichannel transceiver system for ADSL and VHDSL services," *IEEE J. on Selected Areas in Comm.*, vol. 9, pp. 909–919, Aug. 1991.
- [33] N. R. Shanbhag and G.-H. Im, "VLSI system design of 51.84 Mb/s transceiver for ATM-LAN and broadband access," *IEEE Trans. Signal Processing*, vol. 46, pp. 1403–1416, May 1998.
- [34] Davic 1.0 Specification Part 8, *Lower Layer Protocols and Physical Interfaces*, Jan. 1996.

- [35] M. Goel and N. R. Shanbhag, “Low-power equalizers for 51.84 Mb/s very high-speed digital subscriber loop (VDSL) modems,” in *IEEE Workshop on Signal Processing Systems (SiPS): Design and Implementation*, (Boston, MA), pp. 317–326, Oct. 1998.
- [36] S. A. Raghavan, J. K. Wolf, L. B. Milstein, and L. C. Barbosa, “Nonuniformly spaced tapped-delay-line equalizers,” *IEEE Trans. Comm.*, vol. 41, no. 9, pp. 1290–1295, Sept. 1993.
- [37] C. J. Nicol, P. Larsson, K. Azadet, and J. H. O’Neill, “A low power 128-tap digital adaptive equalizer for broadband modems,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1777–1789, Nov. 1997.
- [38] D. P. Bertsekas, *Nonlinear programming*. Boston, Mass.: Athena Scientific, 1995.
- [39] B. Widrow *et al.*, “Stationary and non-stationary learning characteristics of the LMS adaptive filter,” *Proceedings of the IEEE*, vol. 64, pp. 1151–1162, Aug. 1976.
- [40] S. Haykin, *Adaptive Filter Theory*. Englewood Cliff, NJ: Prentice Hall, 1991.
- [41] M. Goel and N. R. Shanbhag, “Finite precision analysis of the pipelined strength-reduced adaptive filter,” *IEEE Trans. Signal Processing*, pp. 1763–1769, June 1998.
- [42] F. Najm, “A survey of power estimation techniques in VLSI circuits,” *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp. 446–455, Dec. 1994.
- [43] M. Pedram, “Power minimization in IC design: principles and applications,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 1, no. 1, pp. 3–56, Jan. 1996.
- [44] 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.
- [45] TIA/EIA-568-A Standard, *Commercial Building Telecommunications Cabling Standard*, 1994.
- [46] G. H. Im and J. J. Werner, “51.84 Mb/s 16-CAP ATM-LAN standard,” *IEEE J. on Selected Areas in Comm.*, vol. 13, no. 4, pp. 620–632, May 1995.

- [47] R. D. Gitlin, J. F. Hayes, and S. B. Weinstein, *Data Communications Principles*. NY: Plenum Press, 1992.
- [48] P. Xue and B. Liu, "Adaptive equalizer using finite-bit power-of-two quantizer," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 34, pp. 1603–1611, Dec. 1986.

Table 1: Converged configurations

Cable Length ( <i>kft</i> )	Number of FEXT interferers	In-phase filter		Quadrature-phase filter	
		$\sum \alpha_{k,opt}$	$B_{w,opt}$	$\sum \alpha_{k,opt}$	$B_{w,opt}$
1.0	11	48	10	48	10
	7	48	10	48	10
	4	16	10	17	10
0.9	11	48	10	48	10
	7	48	10	48	10
	4	10	9	10	9
0.8	11	48	10	48	10
	7	41	10	42	10
	4	12	9	11	9
0.7	11	48	10	48	10
	7	15	10	14	10
	4	14	10	7	9
0.6	11	45	10	44	10
	7	15	10	13	10
	4	8	9	8	9
0.5	11	27	10	23	10
	7	12	9	11	9
	4	10	9	14	10
0.4	11	13	10	8	9
	7	16	10	11	9
	4	9	9	9	9
0.3	11	8	9	9	9
	7	10	9	9	9
	4	9	9	10	9
0.2	11	9	9	9	9
	7	9	9	6	9
	4	10	9	8	9
0.1	11	11	9	8	9
	7	14	10	8	9
	4	15	10	9	9

Table 2: Converged  $SNR_o$ 

Cable Length ( <i>kft</i> )	Number of FEXT interferers	Worst-case design $SNR_o$ (dB)	DAT-based design $SNR_o$ (dB)
1.0	11	21.3 dB	21.3 dB
	7	21.8 dB	21.8 dB
	4	23.1 dB	22.2 dB
0.9	11	21.6 dB	21.6 dB
	7	22.6 dB	22.6 dB
	4	23.8 dB	22.5 dB
0.8	11	21.9 dB	21.9 dB
	7	22.9 dB	22.5 dB
	4	24.2 dB	22.2 dB
0.7	11	22.4 dB	22.4 dB
	7	23.3 dB	22.3 dB
	4	24.2 dB	22.4 dB
0.6	11	22.8 dB	22.0 dB
	7	23.9 dB	22.5 dB
	4	25.0 dB	22.8 dB
0.5	11	23.5 dB	22.2 dB
	7	24.3 dB	22.6 dB
	4	25.5 dB	22.9 dB
0.4	11	24.1 dB	22.3 dB
	7	25.2 dB	22.7 dB
	4	26.1 dB	23.0 dB
0.3	11	25.0 dB	22.4 dB
	7	25.9 dB	23.0 dB
	4	27.1 dB	23.3 dB
0.2	11	26.3 dB	23.8 dB
	7	27.2 dB	24.0 dB
	4	28.5 dB	23.2 dB
0.1	11	28.8 dB	23.6 dB
	7	29.7 dB	23.2 dB
	4	30.5 dB	23.1 dB

Table 3: Energy consumption

Cable length ( <i>kft</i> )	Number of FEXT interferers	$p(\mathbf{s}_i)$	Worst-case design $\mathcal{E}_{WC}$ ( <i>mW/MHz</i> )	DAT-based design			
				$\mathcal{E}_{SPA}$ ( <i>mW/MHz</i> )	$\mathcal{E}_{SMA}$ ( <i>mW/MHz</i> )	$\mathcal{E}_{DAT}$ ( <i>mW/MHz</i> )	% $\mathcal{E}_{sav}$
1.0	11	0.007	3.498	3.498	0.061	3.559	-1.8
	7	0.014	3.439	3.439	0.061	3.500	-1.8
	4	0.007	3.408	1.199	0.061	1.260	63.0
0.9	11	0.016	3.412	3.412	0.061	3.474	-1.8
	7	0.033	3.417	3.417	0.061	3.479	-1.8
	4	0.016	3.426	0.684	0.061	0.745	78.3
0.8	11	0.031	3.395	3.395	0.061	3.456	-1.8
	7	0.061	3.262	2.981	0.061	3.042	6.7
	4	0.031	3.334	0.852	0.061	0.914	72.6
0.7	11	0.045	3.452	3.452	0.061	3.514	-1.8
	7	0.089	3.169	1.012	0.061	1.073	66.1
	4	0.045	3.405	0.855	0.061	0.916	73.1
0.6	11	0.051	3.223	2.971	0.061	3.033	5.9
	7	0.101	3.418	1.144	0.061	1.205	64.7
	4	0.051	3.196	0.553	0.061	0.614	80.8
0.5	11	0.045	3.124	1.445	0.061	1.506	51.8
	7	0.089	3.230	0.744	0.061	0.805	75.1
	4	0.045	2.847	0.809	0.061	0.870	69.4
0.4	11	0.031	2.844	0.623	0.061	0.684	75.9
	7	0.061	3.169	0.915	0.061	0.977	69.2
	4	0.031	2.978	0.611	0.061	0.672	77.4
0.3	11	0.016	3.090	0.571	0.061	0.632	79.5
	7	0.033	3.257	0.581	0.061	0.642	80.3
	4	0.016	3.135	0.665	0.061	0.727	76.8
0.2	11	0.007	3.189	0.550	0.061	0.611	80.8
	7	0.014	3.099	0.537	0.061	0.599	80.7
	4	0.007	3.347	0.601	0.061	0.663	80.2
0.1	11	0.002	3.326	0.659	0.061	0.721	78.3
	7	0.004	3.484	0.723	0.061	0.785	77.5
	4	0.002	3.294	0.815	0.061	0.876	73.4
<b>Average</b>			<b>3.240</b>	<b>1.471</b>	<b>0.061</b>	<b>1.532</b>	<b>52.7</b>