Decorrelating (DECOR) Transformations for Low-Power Digital Filters

Sumant Ramprasad, Naresh R. Shanbhag, *Member, IEEE,* and Ibrahim N. Hajj, *Fellow, IEEE*

***Abstract*— Decorrelating transformations (referred to as DECOR transformations) to reduce the power dissipation in digital filters are presented in this paper. The transfer function and/or the input is decorrelated such that fewer bits are required to represent the coefficients and inputs. Thus, the size of the arithmetic units in the filter is reduced, thereby reducing the power dissipation. The DECOR transform is suited for narrow-band filters because there is significant correlation between adjacent coefficients. Simulations with fixed coefficient filters indicate reduction in transition activity, ranging from 6% to 52% for filter bandwidths ranging from 0.30π to 0.05π, respectively, (where π corresponds to half the sample rate). Simulations with adaptive filters indicate reduction in transition activity in the F-block, ranging from 12% to 38% for filter bandwidths ranging from 0.30π to 0.05π, respectively. The DECOR transforms result in greater energy savings and over a larger bandwidth than existing methods.**

***Index Terms*—CMOS VLSI, digital filters, low-power.**

I. Introduction

POWER reduction techniques form an integral part of low-power VLSI systems design and is presently an active area of research [3], [5], [12], [22]. These techniques have been proposed at each level of the design hierarchy, beginning with algorithms and architectures and ending with circuits and technological innovations. Existing techniques include those at the algorithmic level (such as reduced complexity algorithms [5]), architectural level (such as pipelining [13], [19], and parallel processing), logic (logic minimization [27] and precomputation [2]), circuit (reduced voltage swing [18], adiabatic logic [4]), and technological level [8]. It is now well recognized that an astute algorithmic and architectural design can have a large impact on the final power dissipation characteristics of the fabricated VLSI solution.

The recent proliferation of portable wireless communication systems has specifically made low-power high-performance digital signal processing (DSP) an important research area. A common DSP operation is filtering where the output $y(n)$ at time n is given by

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k) + \sum_{k=1}^{M} a_k y(n-k) \tag{1.1}$$

where b_k and a_k are the coefficients of the filter and $x(n)$ is the input. Equivalently, in the z-transform domain

$$Y(z) = H(z)X(z) \tag{1.2}$$

where $Y(z)$, $H(z)$, and $X(z)$ are the z-transforms of the output, filter, and input, respectively. In addition, to the generic power-reduction techniques mentioned above, techniques exist specifically for filters [1], [10], [14]–[16], [21], [24]–[26]. These techniques employ the fact that power dissipation in CMOS VLSI circuits occurs mainly during signal transitions and is given by

$$P = tC_L V_{\mathrm{dd}}^2 f \tag{1.3}$$

where t is the transition activity, C_L is the capacitance, V_{dd} is the supply voltage, and f is the frequency of operation. The *transition activity* of a bit-level signal b_n is defined as

$$t = \Pr(b_n = 0 \text{ and } b_{n-1} = 1) + \Pr(b_n = 1 \text{ and } b_{n-1} = 0)$$

where $\Pr(A)$ is the probability of the occurrence of event A.

Power dissipation is reduced by reducing one or more of t, C_L, V_{dd}, and f in (1.3). The techniques are (with the quantities being reduced in parenthesis) multirate architectures (C_L and V_{dd}), prefiltering (C_L) [1], block finite-impulse response (FIR) filters (C_L and V_{dd}) [21], coefficient optimization (C_L), differential coefficients (C_L) [24], and coefficient reordering (t). In other work on low-power adaptive filters [14], the total switched capacitance is reduced by dynamically varying the filter order based on signal statistics. In [10], power reduction is achieved by a combination of powering down filter taps and modifying the coefficients. In [26], the strength reduction transformation is applied at the algorithmic level to reduce power dissipation in complex adaptive filters.

In this paper, an algorithm transformation technique, referred to as **DECOR** transforms, is presented to reduce the number of bits required to represent the filter coefficients or the input samples by decorrelating the coefficients or the input.

Manuscript received June 1, 1998; revised January 4, 1999. This work was supported by the Defence Advanced Research Projects Agency under Contract DABT63-97-C-0025, and by the National Science Foundation under CAREER Award MIP-9623737 and Award MIP-9710235.This paper was recommended by Associate Editor M. Simaan.

The authors are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

Publisher Item Identifier S 1057-7130(99)04874-0.

The **DECOR** transform is applicable to filters in which the magnitude of the difference between the absolute values of adjacent coefficients of $H(z)$ is less than the magnitude of the coefficients themselves. Hence, fewer bits are required to represent the differences compared to the actual coefficients. Thus, the size of the arithmetic units in the filter is reduced, thereby reducing the power dissipation. In addition to a reduction in power dissipation, there is also a reduction in delay and area due to smaller bit widths at the inputs to the multipliers. **DECOR** transforms are well-suited to custom, hardwired, fixed-point implementations since power reduction is achieved by reducing the size of the arithmetic units. Since **DECOR** transforms maintain the regularity of the original direct form (DF) filter, they can also be considered for implementation on a programmable fixed-point processor.

The signal flow graph transformations (SFGT) in [16], which were developed independently, are a special case of the **DECOR** transform. The differences between our work and [16] are as follows. The **DECOR** transforms are applied to infinite-impulse response (IIR) filters, adaptive filters [23], filters with rounding after the output of multipliers, and the inputs to a filter in addition to fixed coefficient FIR filters which retain full numerical precision. We study the types of filters that are suitable for **DECOR** transforms and provide gate-level simulations to illustrate the effect of such filter parameters as cutoff frequency and filter order on the energy savings.

Another approach close to **DECOR** in literature is the differential coefficients method (**DCM**) [24], in which differential coefficients are employed for FIR filters. In FIR filters, the output is given by

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k). \tag{1.4}$$

The first-order differential coefficients δ_k^1 are given by

$$\delta_k^1 = b_k - b_{k-1}. \tag{1.5}$$

Each product term $b_k x(n-k)$, (except $b_0 x(n)$) in (1.4), is written as

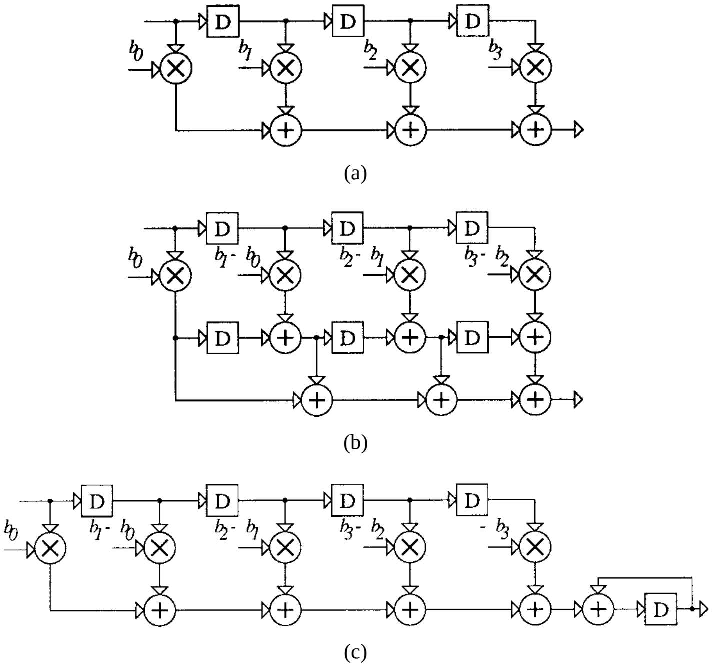
$$b_k x(n-k) = \delta_k^1 x(n-k) + b_{k-1} x(n-k). \tag{1.6}$$

The result of applying **DCM** to the DF filter in Fig. 1(a) is shown in Fig. 1(b).

It is possible to employ second-order differences δ_k^2 by repeating the above procedure on the first-order differential coefficients δ_k^1. The advantage of **DCM** is that the width of the coefficients are reduced but $N-1$ additional adders and latches are required for an N tap filter. In addition to **DCM**, there are other approaches that exploit coefficient correlation. In [7], the frequency response of the filter is used to select an appropriate architecture from among the fast FIR algorithms proposed in [17].

In this paper, motivated by **DCM**, an alternative approach is proposed to realize a filter with differential coefficients. Our formulation results in the following advantages over **DCM**:

1) lower overhead for a given filter order;
2) overhead is independent of the filter order;
3) energy savings over a wider range of filter bandwidths;
4) easily and efficiently implementable in software;
5) applicable to adaptive filters.



(a)

(b)

(c)

Fig. 1. (a) DF. (b) **DCM**. (c) **DECOR** filters.

Note that, power reduction techniques such as parallel processing and pipelining and those in [10], [14], [26] can be applied in addition to **DECOR** for further power savings.

The rest of this paper is organized as follows. In Section II, the **DECOR** transform is presented and applied to fixed coefficient FIR and IIR filters. In Section III, the **DECOR** transform is applied to adaptive filters and in Section IV, analytical and simulation results for the reduction in power dissipation are presented.

II. DECOR Transform Applied to Fixed-Coefficient Filters

In this section, the **DECOR** transform is presented and applied to fixed coefficient FIR and IIR filters. The relaxed **DECOR** transform, applicable to filters which do not retain full numerical precision, is then presented. Finally, the **DECOR** transform is applied to the filter inputs.

A. *The* DECOR *Transform*

In **DECOR**, the transfer function $H(z)$ is multiplied and divided by the polynomial

$$f(z) = (1 + \alpha z^{-\beta})^m \tag{2.1}$$

where α, β, and m are integers that are chosen, depending on the frequency profile of $H(z)$. Thus, the z-transform of the output is given by

$$Y(z) = H(z)\frac{(1+\alpha z^{-\beta})^m}{(1+\alpha z^{-\beta})^m} X(z). \tag{2.2}$$

The frequency response is not altered by multiplying and dividing by $(1+\alpha z^{-\beta})^m$ as long as finite precision effects are

TABLE I
OVERHEAD IN **DECOR** AND **DCM** FOR AN N TAP FIR FILTER

Arithmetic Unit	**DECOR**	**DCM**
Delay	$2\beta m$	mN
Adder	$2\beta m$	mN
Multiplier	βm	0

```
        ...                         ...
DF_loop:                    DECOR_loop:
        ...                         ...
rpt    #N                   rpt    #N+1
mac    (*r1)+,(*r2)+,b      mac    (*r1)+,(*r2)+,b
...                         ...
clr    b                    ; Note: b is not set to 0
jmp    DF_loop              jmp    DECOR_loop
...                         ...
```

Fig. 2. DF and **DECOR** Code for an N tap FIR filter.

accounted for. The numerator polynomial $H(z)(1+\alpha z^{-\beta})^m$ results in a filter with differential coefficients. The denominator $(1+\alpha z^{-\beta})^m$ introduces a recursive section. The parameter α (determined in Appendix A) is either 1 or -1 and determines if coefficients spaced β sample delays apart are either added or subtracted. The parameter m is analogous to the order of difference in [24]. The coefficient bit width typically decreases with m. There is, however, a limit to m, since the bit widths of the coefficients cannot be less than zero and the overhead of **DECOR** increases with m. Fig. 1(c) shows the filter obtained after applying **DECOR** with $\alpha = -1$, $\beta = 1$, and $m = 1$ (i.e., $f(z) = 1 - z^{-1}$) to the DF filter in Fig. 1(a). In Fig. 1(c), all coefficients except for the left-most and right-most, are differences of adjacent coefficients in the original filter. Note that the left-most coefficients in **DECOR** [see Fig. 1(c)] and DF [see Fig. 1(a)] are identical, while the right-most coefficients have opposite signs. Thus, for there to be a reduction in bit-width, the left-most and right-most coefficients in the original filter must be small in magnitude, which is true in most practical filters.

In Fig. 1(c), the result of applying **DECOR** to an FIR filter is an IIR filter with additional hardware, compared to the original FIR filter. The overhead in **DECOR** and **DCM** is summarized in Table I. Unlike **DCM**, the overhead in **DECOR** is independent of the filter order and, for this reason, is less than that of **DCM**. The overhead in **DECOR** depends on $H(z)$. For instance, if m is one and all of the coefficients of the original filter are nonzero, then for symmetric and anti-symmetric filters, no additional multiplier is required. The overhead in **DECOR** also depends on whether the filter is implemented in software or hardware. As Fig. 2 shows, it is nearly as efficient to implement a **DECOR** filter as a DF filter in software. The **DECOR** code executes one additional multiply-accumulate instruction, but saves on setting register b to zero due to the presence of the recursive section in Fig 1(c).

The values of α and β are determined in Appendix A and summarized in Table II. The values of α and β in Table II were verified to be effective in reducing the bit widths when the filter was designed employing such other methods as the window-based method, Parks–McClellan, or least squares. In Table II $|\alpha|$, the absolute value of α, is always equal to one.

TABLE II
α AND β FOR DIFFERENT TYPES OF FIR FILTERS

Filter Type	α	β	$f(z)$
Low-pass	-1	1	$(1-z^{-1})^m$
High-pass	1	1	$(1+z^{-1})^m$
Band-pass (ω_c=center freq.)	1	$\frac{\pi}{\omega_c}$	$(1+z^{-\frac{\pi}{\omega_c}})^m$
Band-stop	-1	2	$(1-z^{-2})^m$

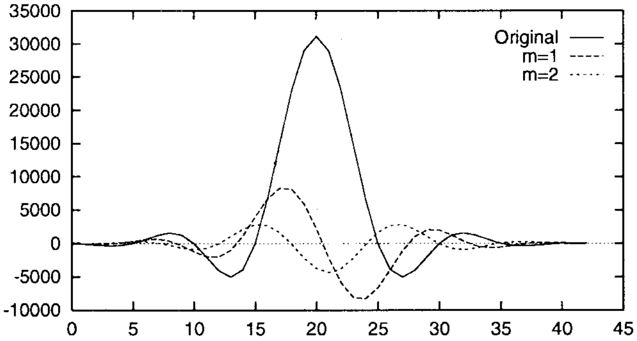


Fig. 3. Coefficients of a low-pass filter before and after **DECOR** ($\alpha = -1$, $\beta = 1$).

Since β is less than the order of the original DF filter, it is also possible perform an exhaustive search to determine optimum α and β. There are three disadvantages if $|\alpha|$ is not equal to one.

1) The magnitude of the difference of the absolute values is not minimized since, typically, the magnitude of the filter coefficients increases till the center coefficient and then decreases.
2) A multiplier is needed in the recursive section of the IIR filter resulting in extra overhead and a quantizer.
3) If $|\alpha| \neq 1$ and $H(z)$ is either symmetric or anti-symmetric, then $H(z)(1+\alpha z^{-\beta})^m$ is not necessarily either symmetric or anti-symmetric. Thus, if $|\alpha| \neq 1$, complexity reductions due to symmetry cannot be exploited.

For the above reasons, α is assumed to be either 1 or -1 throughout this paper.

Fig. 3 shows the coefficients of a 41 tap low-pass FIR filter with cutoffs $\pm\pi/5$ (where π corresponds to half the sample rate) before and after applying **DECOR**. There is a decrease in the range of coefficients after applying **DECOR** once ($\alpha = -1$, $\beta = 1$, $m = 1$) and a further decrease after applying **DECOR** twice ($\alpha = -1$, $\beta = 1$, $m = 2$). Fig. 4 shows the coefficients of a 41 tap high-pass FIR filter with cutoffs $\pm 4\pi/5$ before and after applying **DECOR**. There is a decrease in the range of coefficients after applying **DECOR** once ($\alpha = 1$, $\beta = 1$, $m = 1$).

If the filter passband is narrow, then from (A.1)–(A.4) for the ideal low-pass, high-pass, bandpass, and bandstop filters, respectively, we see that the sinc function is sampled at smaller intervals, leading to a smaller difference between the magnitude of adjacent coefficients. Thus, as the width of the passband is reduced, the reduction in coefficient bit width is increased, and the effectiveness of **DECOR** is increased. If

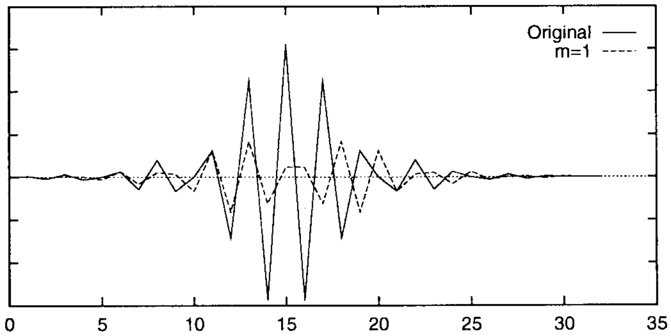


Fig. 4. Coefficients of a high-pass filter before and after **DECOR** ($\alpha = 1$, $\beta = 1$, $m = 1$).

the original coefficients are quantized such that the magnitude of the maximum coefficient is close to the maximum value possible for the given bit width and $m = 1$, then it can be shown that the cutoff frequency ω must be less than 0.385π (where π corresponds to half the sample rate) for there to be a reduction in the number of bits for a low-pass filter generated using the sinc function. Similarly, for a high-pass filter, the cutoff frequency must be greater than 0.615π. Simulation results are presented in Section IV to illustrate the impact of cutoff frequency on the effectiveness of **DECOR**.

Multiplying and dividing the transfer function by $(1 + \alpha z^{-\beta})^m$ introduces m poles and m zeros at each of the β roots of $-\alpha$. Since $|\alpha|$ is equal to one, all the new poles and zeros are on the unit circle. For stability, the new poles must exactly cancel with the new zeros. To ensure exact cancelation of poles and zeros, the coefficients in $H(z)$ must be *first* quantized and then multiplied with $(1 + \alpha z^{-\beta})^m$. In addition, the multiplications and additions in the actual implementation of $H(z)$ have to be exact, i.e., without rounding or truncation. The next subsection describes the issues that arise when this assumption is relaxed.

B. *Relaxed* DECOR

In most filters, quantization (i.e., rounding or truncation) is employed at the output of the multipliers in order to reduce the size of the adders. The noise due to quantization will depend on the amount of quantization; the noise is increased as more bits are dropped at the output of the multipliers. So far, no quantization has been employed at the output of the multipliers in **DECOR** filters in order to guarantee stability. Relaxed **DECOR** is identical to **DECOR**, except that quantization is employed at the output of multipliers. There are two issues in relaxed **DECOR** filters: stability and quantization noise.

In relaxed **DECOR** filters, the pole on the unit circle introduced by the recursive section is not cancelled by the preceding feed-forward section. Hence, to guarantee stability, the IIR section can be modified to include a saturating adder. Alternatively, a multiplier with coefficient close to, but less than unity can also be employed in the recursive section so as to move the pole inside the unit circle. In our simulations, a saturating adder is employed in the recursive section because, in practice, it results in lower overhead and lower deviation from the ideal (i.e., unquantized) output.

The quantization at the output of the multipliers in relaxed **DECOR** filters has to be done such that the quantization error does not build up in the recursive part. A build-up of the quantization error can occur if the quantization error has a dc bias. For instance, the output, $y_{\text{relaxed}}(n)$, of a relaxed **DECOR** filter with $\alpha = -1$, $\beta = 1$, and $m = 1$ is given by

$$\begin{aligned} y_{\text{relaxed}}(n) = Q[b_0 x(n)] + \sum_{k=1}^{N-1} Q[(b_k - b_{k-1})x(n-k)] \\ -Q[b_{N-1}x(n-N)] + y_{\text{relaxed}}(n-1) \end{aligned}$$

where $Q[]$ represents the quantization operation. If $Q[x] \leq x$, then it can be shown by induction on n that the difference between the ideal (i.e., unquantized) output and relaxed **DECOR**, $y(n) - y_{\text{relaxed}}(n)$ is a nondecreasing function of n. The condition $Q[x] \leq x$ is in fact satisfied when two's complement with truncation is employed. Hence, if two's-complement arithmetic is employed, then rounding, not truncation, must be employed at the output of the multipliers in relaxed **DECOR**. If, however, sign-magnitude representation is employed, then it may be possible to employ either truncation or rounding in relaxed **DECOR**. In our simulations, rounding with two's-complement arithmetic was employed.

C. *Low-Power IIR Filters*

In an IIR filter, the transfer function $H(z)$ can be written as

$$H(z) = \frac{N(z)}{D(z)} \tag{2.3}$$

where $N(z)$ and $D(z)$ are the numerator and denominator polynomials, respectively. To apply **DECOR** to IIR filters, the transfer function will have to be multiplied and divided by two polynomials, i.e.,

$$Y(z) = \frac{N(z)}{D(z)} \frac{(1 + \alpha_N z^{-\beta_N})^{m_N}}{(1 + \alpha_N z^{-\beta_N})^{m_N}} \frac{(1 + \alpha_D z^{-\beta_D})^{m_D}}{(1 + \alpha_D z^{-\beta_D})^{m_D}} X(z)$$

where (α_N, β_N, m_N) and (α_D, β_D, m_D) depend on $N(z)$ and $D(z)$, respectively. If $N(z)$ and $D(z)$ are such that $\alpha_N = \alpha_D = \alpha$ and $\beta_N = \beta_D = \beta$, then $H(z)$ needs to be multiplied and divided by one polynomial, i.e.,

$$Y(z) = \frac{N(z)(1 + \alpha z^{-\beta})^m}{D(z)(1 + \alpha z^{-\beta})^m} X(z).$$

Practical values of (α_N, β_N) and (α_D, β_D) for low-pass, high-pass, bandpass, and bandstop filters were obtained experimentally and are shown in Table III. For example, for low-pass Chebyshev, Butterworth, and Elliptic filters, the values of (α_N, β_N) and (α_D, β_D) were experimentally determined as $(-1, 1)$ and $(1, 1)$, respectively.

In Appendix B, we show that, unlike FIR filters, the output of the filter obtained after applying **DECOR** to an IIR filter will not be identical to the output of the original DF filter. The difference in outputs is unavoidable and is due to the quantizer in IIR filters.

The **DECOR** transform can be combined with other algorithm transformations such as look-ahead pipelining [20]. For instance, when look-ahead pipelining is applied to the IIR

TABLE III
α_N, α_D, β_N, and β_D FOR DIFFERENT TYPES OF IIR FILTERS

Filter Type	α_N	β_N	α_D	β_D	$f(z)$
Low-pass	-1	1	1	1	$\frac{(1-z^{-1})^{m_N}(1+z^{-1})^{m_D}}{(1-z^{-1})^{m_N}(1+z^{-1})^{m_D}}$
High-pass	1	1	-1	1	$\frac{(1+z^{-1})^{m_N}(1-z^{-1})^{m_D}}{(1+z^{-1})^{m_N}(1-z^{-1})^{m_D}}$
Band-pass (ω_c = center freq.)	1	1	-1	1	$\frac{(1+z^{-\frac{\pi}{\omega_c}})^{m_N}(1-z^{-\frac{\pi}{\omega_c}})^{m_D}}{(1+z^{-\frac{\pi}{\omega_c}})^{m_N}(1-z^{-\frac{\pi}{\omega_c}})^{m_D}}$
Band-stop	-1	2	1	2	$\frac{(1-z^{-2})^{m_N}(1+z^{-2})^{m_D}}{(1-z^{-2})^{m_N}(1+z^{-2})^{m_D}}$

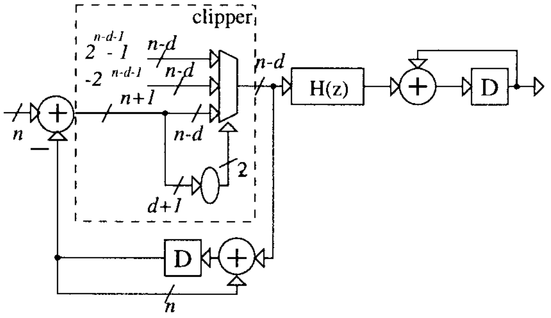


Fig. 5. **DECOR** for filter input ($\alpha = -1$, $\beta = 1$, $m = 1$).

filter $H(z) = 1/(1 - az^{-1})$, it can be shown that there is a decrease in the bit-width of the filter coefficients of the feed-forward path after applying **DECOR** if $1 > |a| > 2/3$ and the look-ahead pipelining level is greater than $2 - 1/(\log_2 |a|)$.

D. **DECOR** *Applied to Filter Inputs*

The **DECOR** transform can be applied to the input to a filter by multiplying and dividing $X(z)$ by $(1 + \alpha z^{-\beta})^m$. The parameters α and β should be chosen depending on the region in the frequency spectrum that contains most of the energy of the input. For instance, if the input is correlated (i.e., most of the energy is concentrated in a small frequency band), then α is equal to -1 and β is equal to 1 (see derivation of α and β for low-pass FIR filters in Appendix A). One difference between the coefficients and the inputs is that typically the input sequence is not known *a priori*. Hence, if a lossless implementation is desired then there will actually be an increase in the input bit width after multiplication with $(1 + \alpha z^{-\beta})^m$. The input bit-width can, however, be reduced if a certain amount of clipping of differences is acceptable. A block diagram for reducing the input bit-width by d bits is presented in Fig. 5. If the difference between the current and previous input samples is greater than $2^{n-d-1} - 1$ (less than -2^{n-d-1}), then the difference is set to $2^{n-d-1}-1$ (-2^{n-d-1}). The clipper output is employed to predict the current input sample (backward prediction) in order to reduce noise.

III. **DECOR** TRANSFORM APPLIED TO ADAPTIVE FILTERS

In this section, the **DECOR** transform is applied to adaptive filters. In order to do so, we derive the following from (1.4):

$$y(n) = -\alpha y(n-\beta) + \sum_{i=0}^{N+\beta-1} \delta_i(n)x(n-i). \tag{3.1}$$

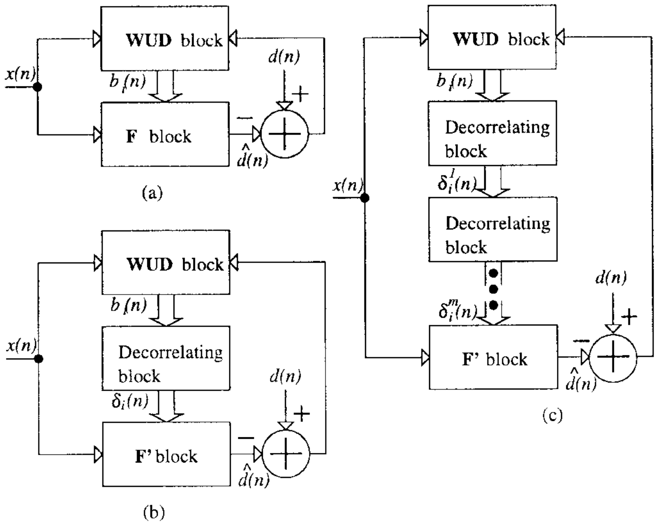


Fig. 6. Adaptive Filter. (a) Traditional. (b) **DECOR** adaptive filter. (c) **DECOR** adaptive filter with multiple decorrelators.

The derivation of (3.1) is presented in Appendix C. The $\delta_i(n)$'s in (3.1) are defined as follows:

$$\delta_i(n) = \begin{cases} b_i(n), & 0 \le i < \beta \\ b_i(n) + \alpha b_{i-\beta}(n-\beta), & \beta \le i < N \\ \alpha b_{i-\beta}(n-\beta), & N \le i < N+\beta \end{cases} \tag{3.2}$$

In (3.2), the first β differential coefficients are identical to the first original β coefficients and the last β differential coefficients are the last original β coefficients multiplied by α. The center $N - \beta$ coefficients are sums or differences (depending on whether α is 1 or -1) of the original coefficients. The size of the multipliers will be reduced if $\max(|\delta_i(n)|)$ is less than $(\max(|b_i(n)|))/2$. Reducing the size of the multipliers reduces the power dissipation and in certain situations, also reduces the delay and the area of the filter.

The traditional adaptive filter, shown in Fig. 6(a), has two blocks.

1) Weight update (**WUD**) block. This block uses the inputs and the error to compute the new coefficients. The weight-update equation for a least-mean squares (LMS) filter is

$$b_i(n+1) = b_i(n) + \mu e(n)x(n-i) \tag{3.3}$$

where μ is the step size and $e(n)$ is the adaptation error given by

$$e(n) = d(n) - y(n). \tag{3.4}$$

In (3.4), $d(n)$ is the desired response of the filter.

2) Filter (**F**) block. This block filters the input employing the coefficients computed by the **WUD** block according to (1.4).

We employ (3.1), (3.2), and (3.3) to construct the **DECOR** adaptive filter. There are three blocks in the **DECOR** adaptive filter (one block each to compute (3.1), (3.2), and (3.3) [see Fig. 6(b)].

1) Weight update (**WUD**) block. This block is identical to the **WUD** block in the original adaptive filter.

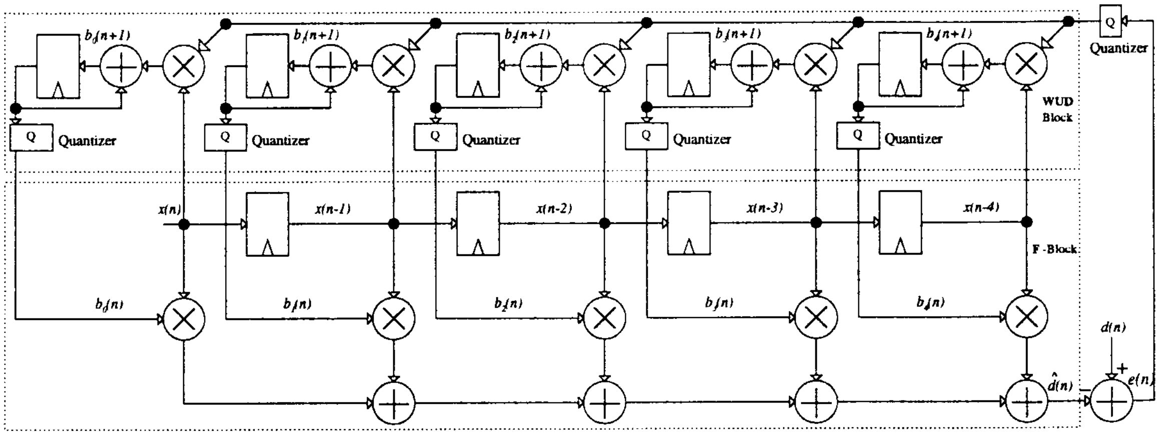


Fig. 7. Traditional LMS filter.

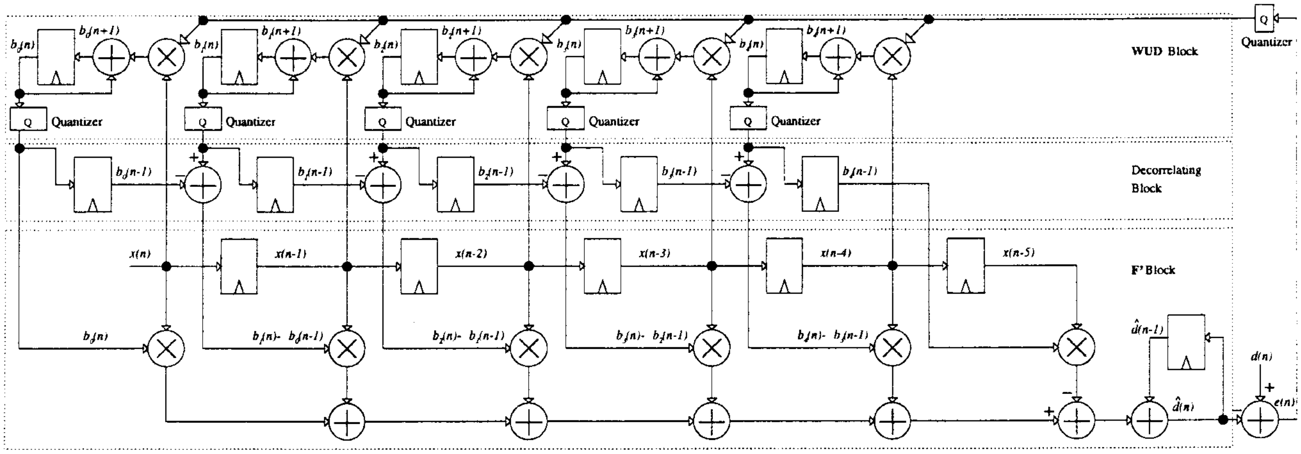


Fig. 8. **DECOR** LMS filter.

2) Decorrelating block. The inputs to this block are the coefficients $b_i(n)$ computed by the **WUD** block. The output consists of decorrelated coefficients $\delta_i(n)$ computed employing (3.2).
3) Filter (**F**$'$) block. This block filters the input employing the coefficients computed by the decorrelating block according to (3.1). As can be seen from Fig. 1(c), this block will be different from the **F**-block in the original adaptive filter.

The inputs to the decorrelating block are the N coefficients $\{b_i(n)\}_{i=0}^{N-1}$. The outputs of the decorrelating block are the $N+\beta$ coefficients $\{\delta_i(n)\}_{i=0}^{N+\beta-1}$. The parameters α and β are chosen as in Table II, depending on the type of filter to be implemented. The output of the **DECOR** adaptive filter is identical to the original adaptive filter as long as the computations in the **F**-block are exact (i.e., no rounding or truncation). Hence, the finite precision analysis for the original adaptive filter holds for the **DECOR** adaptive filter as well.

It is possible to employ higher order differences corresponding to $m > 1$ in (2.2) by cascading more than one decorrelating block, as shown in Fig. 6(c). In Section IV, we will see that, typically, a single decorrelating block provides the most reduction in power dissipation, and hence, in most situations, multiple decorrelating blocks are not needed.

After the filter has converged, the power dissipation in the **WUD** and decorrelating blocks can be reduced substantially by powering them down. Hence, after convergence, only the **F**$'$ block will consume power. The block diagrams for an adaptive LMS filter and a **DECOR** adaptive LMS filter are shown in Figs. 7 and 8, respectively.

Note that dynamic algorithm transformations [10] can be applied in addition to the **DECOR** transform, in which case, the **WUD** block would send a zero coefficient for the taps to be turned off. The decorrelating block would, as always, generate a new set of coefficients employing the coefficients from the **WUD** block.

It may be possible to combine the **WUD** block with the decorrelating block by deriving update equations for $\delta_i(n)$. We can employ (3.2) and (3.3) to calculate $\delta_i(n)$ directly from its previous values as indicated by

$$\delta_i(n+1) = \begin{cases} \delta_i(n)+\mu e(n)x(n-i), & 0\le i<\beta \\ \delta_i(n-\beta+1)+\mu\sum_{k=0}^{\beta-1}(e(n-k) & \\ \quad +\alpha e(n-k-\beta))x(n-i-k), & \beta\le i<N \\ \delta_i(n-\beta+1) & \\ \quad +\alpha\mu\sum_{k=0}^{\beta-1} e(n-k-\beta)x(n-k-i), & N\le i<N+\beta. \end{cases} \tag{3.5}$$

Hence, it is possible to combine the decorrelating block and the **WUD** block by modifying the **WUD** block to calculate $\delta_i(n)$ directly, instead of the **WUD** block computing $b_i(n)$ and the decorrelating block computing $\delta_i(n)$. One potential problem with combining the **WUD** block and the decorrelating block employing (3.5) is that quantizers are often employed in the **WUD** block to reduce the precision of the coefficients (see Figs. 7 and 8). The output of the **DECOR** LMS filter will not be identical to the original filter, in general, because of the quantizers. The effect of the quantizer will, however, be small if enough bits are employed for $b_i(n)$, in which case the **WUD** and decorrelating blocks can be combined.

In this section, the application of the **DECOR** transform to the coefficients has been described. For an LMS filter, **DECOR** transforms can also be applied to the inputs. For instance, the following can be derived from (1.4):

$$\begin{aligned} y(n) = &-\alpha y(n-\beta) \\ &+ \sum_{i=0}^{N-1} b_i(n-\beta)(x(n-i)+\alpha x(n-i-\beta)) \\ &+ \mu \sum_{i=0}^{N-1}\sum_{j=0}^{\beta-1} e(n-j)x(n-i)x(n-i-j). \end{aligned} \tag{3.6}$$

The derivation of (3.6) is similar to that of (3.1). In (3.6), one of the inputs to the multiplier is the difference between successive input samples instead of the input samples themselves. The size of the multipliers can be reduced if the input is correlated, and the maximum of the difference is less than half the maximum of the original samples. This implies that the input must have most of its energy concentrated in a narrow band of the frequency spectrum. There is no restriction on the transfer function of the filter when **DECOR** is applied to the input. There is an overhead in the form of the final double summation in (3.6). After convergence, this term will be small because the error $e(n)$ will be small. Thus, the final term does not have to be computed if the **DECOR** transform is applied to the input only after convergence.

IV. Results

In this section, analytical and simulation results for estimating the reduction in power dissipation due to **DECOR** and **DCM** are presented. The analytical model in [24] will be employed to estimate the reduction in transition activity for FIR filters. We will then present gate-level simulations for the reduction in transition activity when **DECOR** is applied to FIR, IIR, and adaptive filters.

A. Analytical Results

The analytical model in [24] was employed to estimate the reduction in power dissipation and the speedup due to **DECOR** and **DCM** for a 40-tap low-pass FIR filter with cutoffs $\pm\pi/6$. In this model, a multiplication is performed by repeatedly adding the multiplicand to the left end of a partial sum, with the MSB of the multiplicand aligned with the MSB of the partial sum. The multiplicand is added only if the multiplier bit is one. The partial sum is right shifted by one bit. The results of the analysis, shown in Figs. 9 and 10, indicate that **DECOR** provides a greater reduction in power dissipation and a greater speedup compared to **DCM**. Furthermore, the advantages of **DECOR** over **DCM** increases with m. The greater reduction in power dissipation is due to the lower overhead in **DECOR**. In the analysis, the original input and coefficients were 16-bits wide and a square-root model was employed for the memory with $\mathcal{K}_{e1} = 1$ and $\mathcal{K}_{t1} = 0.1$.

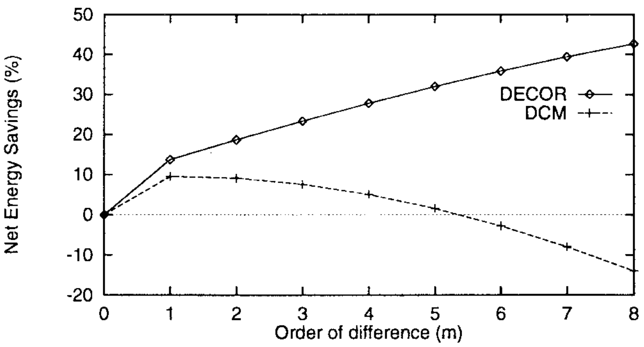


Fig. 9. Analytical estimate of reduction in transition activity due to **DECOR** and **DCM**.

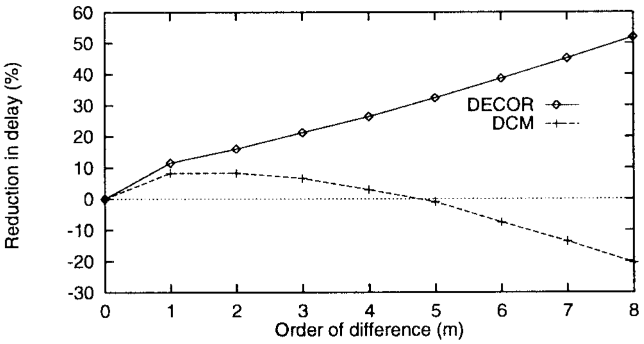


Fig. 10. Analytical estimate of speedup due to **DECOR** and **DCM**.

B. Simulation Model

In addition to the above analysis, a tool to perform gate-level simulations of DF, **DCM**, and **DECOR** filters was developed in C. This tool contains gate-level models of a ripple-carry adder and a two's-complement array multiplier [11]. The array multiplier was employed in the simulations because of its simplicity and regularity. The simulation results are different from the analytical results presented in Section IV-A because the array multiplier model is different from the shift- and add-multiplier employed in the analysis in [24]. Also, unlike the analysis which assumed only one adder and one multiplier were available, the simulation assumed all the multiplications and additions were performed on separate units. All multipliers and adders were assumed to have operands of the same bit width.

In our simulations, the following three quantities were measured.

1) The total number of transitions at the inputs to the gates. This is a measure of the power dissipation in CMOS cir-

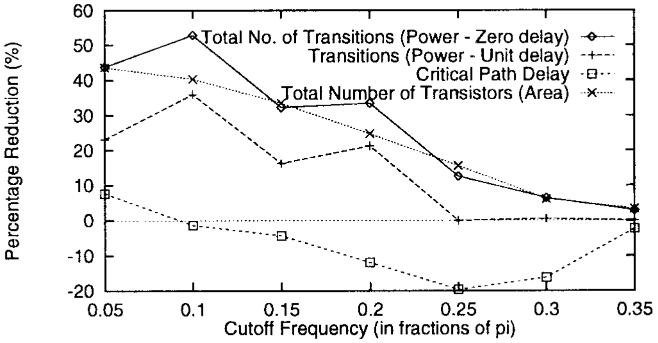


Fig. 11. Effect of cutoff frequency (filter order = 40, coefficient precision = 16 bits, data precision = 16 bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

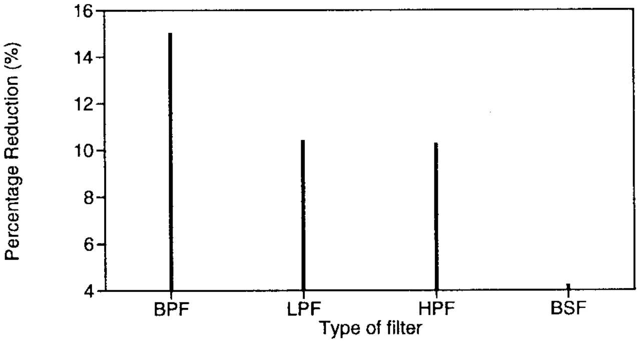


Fig. 12. Reduction in transition activity due to **DECOR** (coefficient precision = 16 bits, data precision = 16 bits, $\beta = 1$, $m = 1$).

cuits because power is dissipated predominantly during signal transitions.

2) The maximum number of gates between two latches. This is a measure of the critical path or delay.
3) The total number of transistors, which provides a measure of the area.

Only inverters and 2 and 3 input NAND gates were employed to construct the adders and multipliers, since the aim is to estimate power dissipation in CMOS circuits. The adder and multiplier models were employed to construct filters. The transitions in a latch were assumed to be equal to the transitions at its inputs. The input for gate-level simulations consisted of 4096 samples of 16-bit PCM audio data. The gates were assumed to have zero delay, since zero-delay simulations are a good measure of transition activity if the multiplier and adder architectures are balanced so that glitches are minimized. A second advantage of zero-delay simulations are that they are fast. For instance, a zero-delay gate-level simulation of a 40-tap low-pass FIR filter with 4096 samples of 16-bit input data required 100 s on a Sparc Ultra-2, whereas a unit-delay simulation of the same filter required 9840 s. One set of simulations was run with unit delay to examine the effect of changing the delay model on the reduction in power dissipation. The correctness of simulations of FIR filters was verified by comparing the outputs of the gate-level simulations and RTL simulations.

C. Simulation Results

Fig. 11 shows the maximum reduction in transition activity after applying **DECOR** to 40-tap low-pass FIR filters. The reduction in transition activity decreases with cutoff frequency from a maximum of 52% for a cutoff of $\pi/10$ to 3% for a cutoff of 0.35π. The order of difference m was chosen to maximize reduction in transition activity. Fig. 11 indicates that **DECOR** is suited for narrow-band filters. The range of bandwidths over which **DECOR** results in power savings (0–0.35π) is, however, much greater than that for **DCM**, due to the lower overhead in **DECOR**. Fig. 11 also indicates that the savings in power dissipation are lower with the unit-delay model compared to the zero-delay model. This is because, in an array multiplier, the number of transitions is higher under the unit-delay model due to glitching caused by reconvergent fanout. Hence, the overhead due to **DECOR** is higher under the unit-delay model, since **DECOR** introduces additional multipliers. The overhead can be reduced by employing a different multiplier (e.g., Booth multiplier) with less glitching, or by modifying the array multiplier to reduce glitching (e.g., introducing latches).

To examine the effectiveness of relaxed **DECOR**, the baseline DF filter was modified so that the least significant 16 bits of the 32-bit output of the multiplier are dropped. This baseline filter was compared with a relaxed **DECOR** filter. The cutoff frequency was $\pi/10$, filter order = 40, coefficient precision = 16 bits, $\alpha = -1$, $\beta = 1$, and $m = 1$. The reduction in total number of transitions was 25.8%, the reduction in total number of transistors was 9.4%, and the increase in the length of the critical path was 5.2%.

Fig. 12 shows that there is a reduction in transition activity for different types of FIR filters after applying **DECOR**. The cutoffs of the bandpass, low-pass, high-pass, and bandstop filters were $[0.41\pi, 0.59\pi]$, $\pi/6$, $5\pi/6$, and $[0.1\pi, 0.92\pi]$, respectively. The cutoff frequencies were chosen such that application of **DECOR** results in a decrease of the coefficient bit-width. A reduction in transition activity was obtained with **DCM** only for the band-pass filter. **DCM**, as presented in [24] has been applied only to low-pass filters. However, the applicability of **DCM** can be easily extended to other types of filters by adding and subtracting $\alpha b_{k-\beta}$ at the output of the multipliers in (1.4), i.e.,

$$b_k x(n-k) = (b_k + \alpha b_{k-\beta})x(n-k) - \alpha b_{k-\beta} x(n-k) \tag{4.1}$$

where α and β are as determined in Appendix A. Another approach to extending **DCM** to high-pass, bandpass, and bandstop filters is by sorting the coefficients and then applying **DCM** [25]. The approach in (4.1) will give greater savings because, for instance, in a high-pass filter, the sum of adjacent coefficients is less than the difference between every second coefficient. For this reason, (4.1) was employed when comparing results for **DCM** for high-pass, bandpass, and bandstop filters.

Fig. 13 shows the effect of m on reduction in transition activity for a low-pass FIR filter with cutoff $\pi/6$. The reduction in transition activity increases with m up to $m = 3$. For

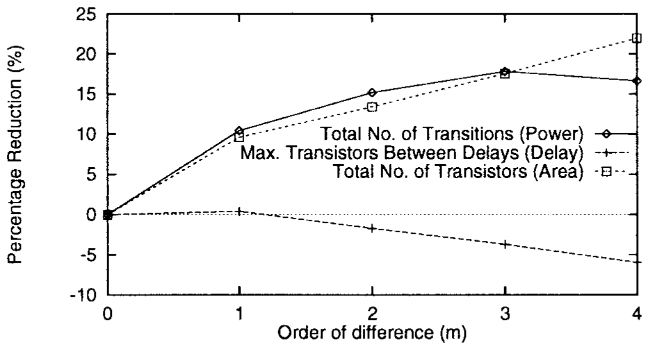


Fig. 13. Reduction in transition activity with m for an FIR filter (cutoff $= \pi/10$, filter order $= 40$, coefficient precision $= 16$ bits, data precision $= 16$ bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

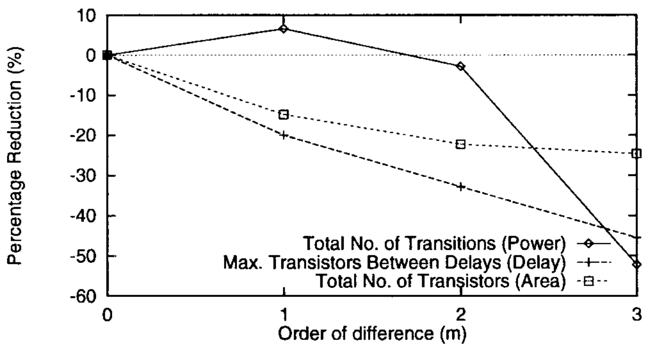


Fig. 14. Effect of m for an IIR filter (cutoff $= \pi/5$, filter order $= 15$, coefficient precision $= 16$ bits, data precision $= 16$ bits, $m = 1$).

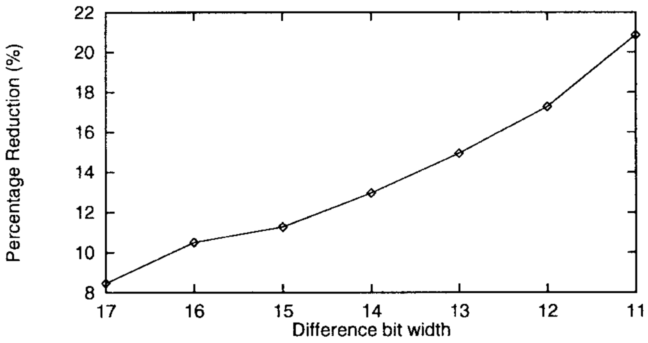


Fig. 15. Reduction in transition activity with difference bit width (cutoff $= \pi/6$, filter order $= 40$, coefficient precision $= 16$ bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

$m > 3$, the reduction in bit width is not significant, leading to an increase in transition activity due to the overhead.

Fig. 14 shows that there is a reduction in transition activity only for $m = 1$ after applying **DECOR** on the recursive part of a 15th-order Butterworth IIR filter with cutoff $\pi/5$. The difference between the output of the filter with and without **DECOR** was 4.2 dB.

Fig. 15 shows the effect of applying **DECOR** to the input to a 40-tap low-pass FIR filter with cutoff $\pi/6$. Reducing the difference bit-width increases the reduction in transition activity for an FIR filter. Transitions in the clipper in Fig. 5 were ignored because most of the differences are not clipped.

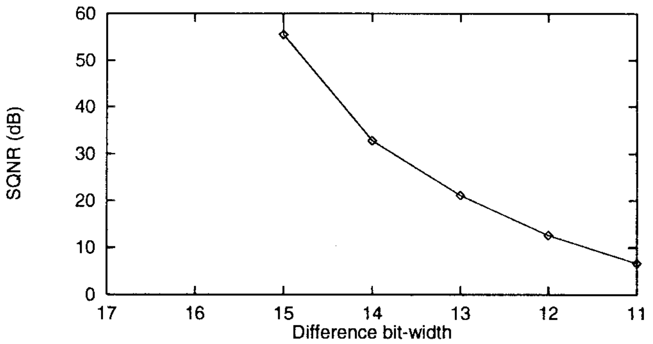


Fig. 16. SQNR (cutoff $= \pi/6$, filter order $= 40$, coefficient precision $= 16$ bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

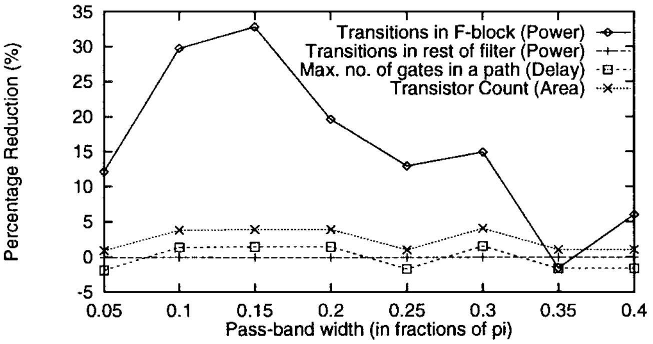


Fig. 17. Effect of passband width (filter order $= 40$, coefficient precision $= 8$ bits, data precision $= 17$ bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

There is a reduction in transition activity of 8.5%, even when the difference bit width is 1 bit more than the original input bit width of 16. This reduction in transition activity is because the dynamic range of the input is reduced by computing the difference. Fig. 16 shows that reducing the difference bit-width decreases the signal to quantization noise ratio (SQNR) at the output of an FIR filter. A longer input sequence of 1 440 000 samples of 16-bit PCM audio data was employed to measure the SQNR, since it can be computed via RTL simulations, which are faster. There is no noise due to clipping of differences for difference bit widths of 16 and 17, since all differences fit in 16 bits.

The results of zero-delay gate-level simulations of serial LMS filters and **DECOR** LMS filters are now presented. The adaptive filter was employed for system identification of low-pass FIR filters. The input was 4096 samples of uniform white noise. The desired response was obtained by applying the input to a filter generated by employing MATLAB's *fir1* command. The step-size μ was set at 0.007 812 5. In order to easily compare the output of the serial LMS filter and the **DECOR** LMS filter, all computations in the $\mathbf{F}'$ block were made exact (i.e., without any rounding or truncation). The baseline filter was a low-pass FIR filter with cutoff of 0.1π, data precision of 17 bits, coefficient precision of 8 bits, and filter order of 40. The power dissipation in the $\mathbf{F}'$ block was measured separately from the rest of the filter because, after the coefficients have converged, all blocks other than the $\mathbf{F}'$ block can be turned off, reducing the power dissipation in those blocks to zero. The

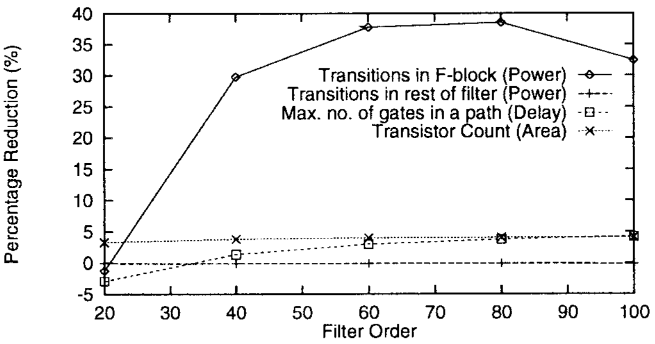


Fig. 18. Effect of filter order (cutoff = 0.1π, coefficient precision = 8 bits, data precision = 17 bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

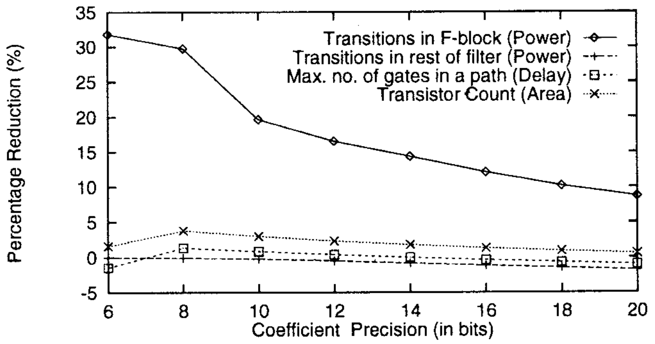


Fig. 19. Effect of coefficient precision (cutoff = 0.1π, filter order = 40, data precision = 17 bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

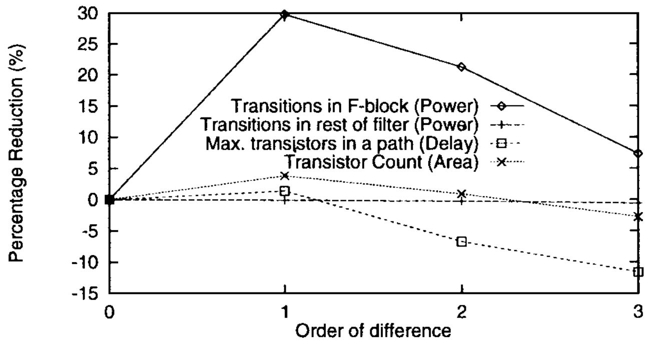


Fig. 20. Effect of order of difference m (cutoff = 0.1π, filter order = 40, coefficient precision = 8 bits, data precision = 17 bits, $\alpha = -1$, $\beta = 1$).

coefficient width was determined experimentally by running RTL simulations and noting the maximum magnitude of the coefficients.

In Fig. 17, in general, the percentage reduction in power dissipation in the **F**′-block is increased as the width of the passband is reduced. This is because the differences between adjacent coefficients is smaller for narrow-band filters leading to a greater reduction in bit width of coefficients. There is little change in delay and area. In Fig. 18, the percentage reduction in power dissipation in the **F**′ block is increased as the filter order is increased. This is because the relative overhead in the **DECOR** transform is decreased as the filter order is increased. There is also a small reduction in delay and area. In Fig. 19, the percentage reduction in power dissipation in the **F**′ block is increased as the precision of the coefficients is decreased. This is because the reduction in the number of bits is generally independent of the precision, due to which the fractional savings is higher for lower precision. There is a small percentage reduction in delay and area which is higher for lower precision. In Fig. 20, the percentage reduction in power dissipation is maximized for $m = 1$. This is because of the overhead for $m > 1$. There is a small reduction in delay and area for $m = 1$. In Fig. 21, the percentage reduction in power dissipation in the **F**′ block is nearly independent of data precision. In all the experiments, the power dissipation in blocks other than the **F**′-block (i.e., **WUD** and decorrelating blocks) changed by less than ±2%.

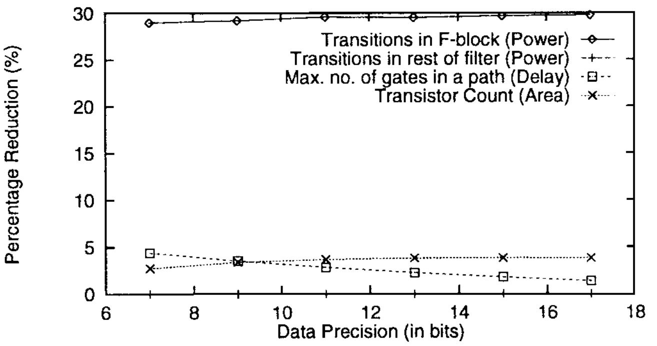


Fig. 21. Effect of data precision (cutoff = 0.1π, filter order = 40, coefficient precision = 8 bits, $\alpha = -1$, $\beta = 1$, $m = 1$).

V. Conclusion

In this paper, the **DECOR** transform is presented, which is a power-reduction technique suited for narrow-band filters. In this method, the coefficients (input) is decorrelated such that fewer bits are required to represent the coefficients (input), thereby reducing the size of the multipliers. Simulations with fixed-coefficient filters indicate a reduction in transition activity ranging from 6% to 52% for filter bandwidths ranging from 0.30π to 0.05π, respectively. Simulations with adaptive filters indicate reduction in transition activity in the **F**′ block ranging from 12% to 38% for filter bandwidths ranging from 0.30π to 0.05π, respectively.

In future, we plan to implement **DECOR** transforms in VLSI circuits and compare the savings in power dissipation with simulation results. The effect of quantization in IIR filters on **DECOR** transforms will also be studied.

Appendix A
Determining α and β for FIR Filters

In this appendix, practical values of α and β for low-pass, high-pass, bandpass, and bandstop filters are determined. This is done by analyzing the ideal sinc prototype filter impulse responses.

The coefficients of an ideal low-pass filter with cutoffs $\pm\omega$ are given by

$$b_k = \frac{\omega}{\pi}\,\mathrm{sinc}(\omega k), \quad -\infty < k < \infty. \tag{A.1}$$

The magnitude of the difference between adjacent coefficients is typically less than the magnitude of the original coefficients. Hence, by choosing $\alpha = -1$ and $\beta = 1$ (i.e., $f(z) = (1 - z^{-1})^m$) the difference between adjacent coefficients can be obtained.

The coefficients of an ideal high-pass filter with cutoffs $\pm(\pi - \omega)$ are given by

$$b_k = (-1)^k \frac{\omega}{\pi} \operatorname{sinc}(\omega k), \quad -\infty < k < \infty. \tag{A.2}$$

Due to the $(-1)^k$ term in (A.2), the magnitude of the sum of adjacent coefficients is typically less than the magnitude of the original coefficients. Hence, by choosing $\alpha = 1$ and $\beta = 1$ (i.e., $f(z) = (1 + z^{-1})^m$) the sum of adjacent coefficients can be obtained.

The coefficients of an ideal bandpass filter with cutoffs $\pm(\omega_c \pm \omega)$ are given by

$$b_k = \cos(\omega_c k) \frac{\omega}{\pi} \operatorname{sinc}(\omega k), \quad -\infty < k < \infty. \tag{A.3}$$

Due to the $\cos(\omega_c k)$ term in (A.3), the magnitude of the sum of coefficients π/ω_c apart is typically less than the magnitude of the original coefficients. Hence, by choosing $\alpha = 1$ and $\beta = \pi/\omega_c$ (actually, π/ω_c rounded to the nearest integer) or $f(z) = (1 + z^{-\pi/\omega_c})^m$, the sum of coefficients spaced π/ω_c apart can be obtained.

The coefficients of an ideal bandstop filter with cutoffs $\pm\omega_1$ and $\pm(\pi - \omega_2)$ are given by

$$b_k = \frac{\omega_1}{\pi} \operatorname{sinc}(\omega_1 k) + (-1)^k \frac{\omega_2}{\pi} \operatorname{sinc}(\omega_2 k), \quad -\infty < k < \infty. \tag{A.4}$$

Due to the $(-1)^k$ term in (A.4), adjacent coefficients will be the sum and difference of two sinc functions. Hence, the range of coefficients can be reduced by choosing $\alpha = -1$ and $\beta = 2$ (i.e., $f(z) = (1 - z^{-2})^m$).

Appendix B
Effect of Quantization in IIR Filters on **DECOR**

In this appendix we show that the output at time n, $y_{df}(n)$, of an N-tap direct form IIR filter will not be identical to the output, $y_{\text{decor}}(n)$ of the filter obtained after applying **DECOR**, i.e.,

$$y_{\text{decor}}(n) = y_{df}(n) + e_n,$$

where e_n is the difference between the two outputs due to the quantizer.

Proof: Without loss of generality, assume that both the DF and **DECOR** filters are in the same state at time n, i.e.,

$$y_{df}(n-i) = y_{\text{decor}}(n-i) = y(n-i), \quad \text{for } i = 1, \dots, N.$$

The output at time n of the DF filter is given by

$$y_{df}(n) = Q\left[-\sum_{i=1}^{N} a_i y(n-i) + x(n)\right]$$

where $Q[\cdot]$ represents the quantization operation. The output of the **DECOR** filter is given by

$$\begin{aligned}
&y_{\text{decor}}(n)\\
&= Q\left[-\sum_{i=1}^{N+1} (a_i + a_{i-1}) y(n-i) + x(n) + x(n-1)\right]\\
&\qquad\qquad \text{[Note: } a_{N+1} = 0, a_0 = 1]\\
&= Q\left[-\sum_{i=1}^{N} a_i y(n-i) - y(n-1)\right.\\
&\qquad \left. - \sum_{i=2}^{N+1} a_{i-1} y(n-i) + x(n) + x(n-1)\right]\\
&= Q\left[-\sum_{i=1}^{N} a_i y(n-i) - Q\left[-\sum_{i=1}^{N} a_i y(n-1-i) x(n-1)\right]\right.\\
&\qquad \left. + -\sum_{i=1}^{N} a_i y(n-1-i) + x(n) + x(n-1)\right]\\
&\qquad \text{[Applying definition of } y_{df}(n) \text{ to } y(n-1) \text{ since, by}\\
&\qquad \text{assumption, } y_{df}(n-1) = y_{\text{decor}}(n-1) = y(n-1)]\\
&= Q\left[-\sum_{i=1}^{N} a_i y(n-i) + \sum_{i=1}^{N} a_i y(n-1-i) - x(n-1)\right.\\
&\qquad \left. + \epsilon_n - \sum_{i=1}^{N} a_i y(n-1-i) + x(n) + x(n-1)\right]\\
&\qquad\qquad [\epsilon_n \text{ is the noise due to the quantizer]}\\
&= Q\left[-\sum_{i=1}^{N} a_i y(n-i) + \epsilon_n + x(n)\right]\\
&= Q\left[-\sum_{i=1}^{N} a_i y(n-i) + x(n)\right] + e_n\\
&\qquad\qquad [e_n \text{ is the noise due to the quantizer]}\\
&= y_{df}(n) + e_n \quad \text{[From definition of } y_{df}(n)],
\end{aligned}$$

which is the desired result. □

Appendix C
Derivation of (3.1)

The derivation of (3.1) is presented in this Appendix.

$$y(n) = \sum_{i=0}^{N-1} b_i(n) x(n-i) \tag{C.1}$$

$$\alpha y(n-\beta) = \alpha \sum_{i=0}^{N-1} b_i(n-\beta) x(n-\beta-i) \tag{C.2}$$

[Replace n with $n - \beta$ in (C.1) & multiply with α]

$$\begin{aligned}
&y(n) + \alpha y(n-\beta)\\
&= \sum_{i=0}^{N-1} b_i(n) x(n-i) + \alpha \sum_{i=0}^{N-1} b_i(n-\beta) x(n-\beta-i)\\
&\qquad\qquad \text{[Add (C.1) and (C.2)]}\\
&= \sum_{i=0}^{\beta-1} b_i(n) x(n-i) + \sum_{i=\beta}^{N-1} b_i(n) x(n-i)
\end{aligned}$$

$$
\begin{aligned}
&+\alpha \sum_{i=0}^{N-\beta-1} b_i(n-\beta)x(n-\beta-i)\\
&+\alpha \sum_{i=N-\beta}^{N-1} b_i(n-\beta)x(n-\beta-i)\\
&= \sum_{i=0}^{\beta-1} b_i(n)x(n-i) + \sum_{i=\beta}^{N-1} b_i(n)x(n-i)\\
&+\alpha \sum_{i=\beta}^{N-1} b_{i-\beta}(n-\beta)x(n-i)\\
&+\alpha \sum_{i=N}^{N+\beta-1} b_i(n-\beta)x(n-i)\\
&\text{[Replace } i \text{ with } i-\beta \text{ in the second and third sums]}\\
&= \sum_{i=0}^{\beta-1} b_i(n)x(n-i) + \sum_{i=\beta}^{N-1} (b_i(n)\\
&+\alpha b_{i-\beta}(n-\beta))x(n-i)\\
&+\alpha \sum_{i=N}^{N+\beta-1} b_{i-\beta}(n-\beta)x(n-i)
\end{aligned}
$$

$$
\begin{aligned}
y(n) &= -\alpha y(n-\beta) + \sum_{i=0}^{\beta-1} b_i(n)x(n-i)\\
&+ \sum_{i=\beta}^{N-1} (b_i(n) + \alpha b_{i-\beta}(n-\beta))x(n-i)\\
&+\alpha \sum_{i=N}^{N+\beta-1} b_{i-\beta}(n-\beta)x(n-i)\\
&= -\alpha y(n-\beta) + \sum_{i=0}^{N+\beta-1} \delta_i(n)x(n-i), \qquad \text{[Using (3.2)]}
\end{aligned}
$$

which is the desired equation. □

References

[1] J. W. Adams and A. N. Willson, "Some efficient digital prefilter structures," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 260–265, May 1984.

[2] M. Alidina, J. Monterio, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low-power," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 426–436, Dec. 1994.

[3] A. P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Broderson, "Optimizing power using transformations," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 12–31, Jan. 1995.

[4] 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 J. Solid-State Circuits*, vol. 2, pp. 398–407, Dec. 1994.

[5] A. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, pp. 498–523, Apr. 1995.

[6] A. Chatterjee and R. K. Roy, "Synthesis of low power linear DSP circuits using activity metrics," in *Proc. 7th Int. Conf. VLSI Design*, Calcutta, India, Jan. 1994, pp. 265–270.

[7] J.-G. Chung, Y.-B. Kim, H.-J. Jeong, and K. K. Parhi, "Efficient parallel FIR filter implementations using frequency spectrum characteristics," in *Proc. IEEE Int. Symp. Circuits Syst.*, Monterey, CA, May–June 1998, pp. V-354–V-358.

[8] B. Davari, R. H. Dennard, and G. G. Shahidi, "CMOS scaling for high-performance and low-power—The next ten years," *Proc. IEEE*, vol. 83, pp. 595–606, Apr. 1995.

[9] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 569–577, Sept. 1995.

[10] M. Goel and N. R. Shanbhag, "Dynamic algorithm transformations (DAT) for low-power adaptive signal processing," in *Int. Symp. Low Power Electronics and Design*, Monterey, CA, Aug. 1997, pp. 161–166.

[11] M. Hatamian and G. L. Cash, "Parallel pipelined multiplier," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 505–513, Aug. 1986.

[12] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in *Proc. Symp. Low Power Electronics*, San Diego, CA, Oct. 1994, pp. 8–11.

[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] J. T. Ludwig, S. H. Nawab, and A. P. Chandrakasan, "Low-power digital filtering using approximate processing," *J. Solid-State Circuits*, vol. 31, pp. 395–400, Mar. 1996.

[15] M. Mehendale, S. D. Sherlekar, and G. Venkatesh, "Algorithmic and architectural transformations for low-power realization of FIR filters," in *Proc. 11th Int. Conf. VLSI Design*, Chennai, India, Jan. 1998, pp. 12–17.

[16] M. Mehendale, S. B. Roy, S. D. Sherlekar, and G. Venkatesh, "Coefficient transformations for area-efficient implementation of multiplier-less FIR filters," in *Proc. 11th Int. Conf. VLSI Design*, Chennai, India, Jan. 1998, pp. 110–115.

[17] Z.-J. Mou and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering," *IEEE Trans. Signal Processing*, vol. 39, pp. 1322–1332, June 1991.

[18] Y. Nakagome, K. Itoh, M. Isoda, K. Takeuchi, and M. Aoki, "Sub-1-V swing internal bus architecture for future low-power ULSI's," *J. Solid-State Circuits*, vol. 28, pp. 414–419, Apr. 1993.

[19] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *Proc. IEEE*, vol. 77, pp. 1879–1895, Dec. 1989.

[20] K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters—Parts I, II," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1099–1134, July 1989.

[21] D. N. Pearson and K. K. Parhi, "Low-power FIR digital filter architectures," *IEEE Int. Symp. Circuits Syst.*, Seattle, WA, Apr.–May 1995, pp. 231–234.

[22] M. Pedram, "Power minimization in IC design: Principles and applications," *IEEE/ACM Trans. Design Automation of Electronic Systems*, vol. 1, pp. 3–56, Jan. 1996.

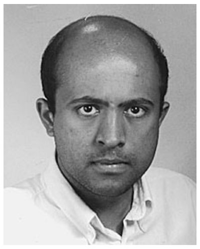
[23] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "Decorrelating (DECOR) transformations for low-power adaptive filters," in *Int. Symp. Low-Power Electronics and Design*, Monterey, CA, Aug. 1998, pp. 250–255.

[24] N. Sankarayya, K. Roy, and D. Bhattacharya, "Algorithms for low power and high speed FIR filter realization using differential coefficients," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 488–497, June 1997.

[25] N. Sankarayya, K. Roy, and D. Bhattacharya, "Optimizing computations in a transposed direct form realization of floating-point LTI-FIR systems," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 1997, pp. 120–125.

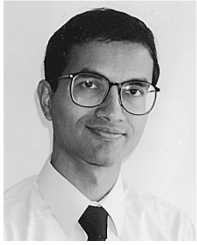
[26] N. R. Shanbhag and M. Goel, "Low-power adaptive filter architectures and their application to 51.84 Mb/s ATM-LAN," *IEEE Trans. Signal Processing*, vol. 45, pp. 1276–1290, May 1997.

[27] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks," in *IEEE/ACM Int. Conf. Computer-Aided Design*, Santa Clara, CA, Nov. 1992, pp. 402–407.



Sumant Ramprasad received the B.Tech. degree in computer science and engineering in 1988 from the Indian Institute of Technology, Bombay, India, and the M.S. degree in computer and information sciences from Ohio State University, Columbus, in 1990. He is presently working toward the Ph.D. degree in computer science at the University of Illinois at Urbana-Champaign.

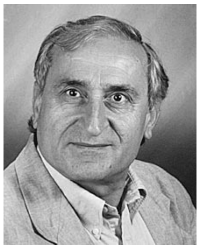
From 1991 to 1996, he was with the Chicago Corporate Research Laboratories of Motorola, Inc. Currently, he is working on high-level estimation, synthesis, and methodologies for low-power design.



Naresh R. Shanbhag (S'87–M'88) received the B.Tech. degree from the Indian Institute of Technology, New Delhi, India, in 1988, the M.S. degree from Wright State University, Dayton, OH, and the Ph.D. degree from the University of Minnesota, Minneapolis St. Paul, in 1993, all in electrical engineering.

From July 1993 to August 1995, he was with AT&T Bell Laboratories, Murray Hill, NJ, in the Wide-Area Networks Group, where he was responsible for development of VLSI algorithms, architectures, and implementation for high-speed data communications applications. In particular, he was the lead chip architect for AT&T's 51.84 Mb/s transceiver chips over twisted-pair wiring for asynchronous transfer mode (ATM)-LAN and broadband access applications. Since August 1995, he has been a Research Assistant Professor with the Coordinated Science Laboratory in the VLSI Circuits Group and an Assistant Professor with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. His research interests are in exploring the limits of computation in an integrated circuit media in terms of power dissipation, reliability, and throughput, and developing VLSI algorithms, architectures, and integrated circuits for signal processing and communication systems. He has published more than 20 journal articles and book chapters and more than 30 conference publications on these subjects.

Dr. Shanbhag received the the 1999 IEEE Leon K. Kirchmayer Best Paper Award, the National Science Foundation CAREER Award in 1996, and the 1994 Darlington Best Paper Award from the IEEE Circuits and Systems Society. He is a Distinguished Lecturer of the IEEE Circuits and Systems Society (1997–1999). Since July 1997, he has served as Associate Editor for IEEE TRANSACTION ON CIRCUITS AND SYSTEMS II.



Ibrahim N. Hajj (S'64–M'70–SM'82–F'90) is a Professor of Electrical and Computer Engineering and a Research Professor at the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, Illinois. He received the B.E. degree (with distinction) from the American University of Beirut, the M.S. degree from the University of New Mexico, Albuquerque, and the Ph.D. degree from the University of California at Berkeley, all in electrical engineering.

Before joining the University of Illinois, he was with the Department of Electrical Engineering at the University of Waterloo, Ontario, Canada. His current research interests include computer-aided design of VLSI circuits, design for reliability and low-power, synthesis, physical design, and testing. He has published over 160 journal and conference papers and book chapters on these subjects. He is a co-author of the book *Switch-Level Timing Simulation of MOS VLSI Circuits* (Norwell, MA: Kluwer, 1989).

Dr. Hajj currently serves on the Board of Governors of the IEEE Circuits and Systems Society. He has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and an Associate Editor of the *IEEE Circuits and Systems Magazine*. In 1992, he was a co-recipient of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN Best Paper Award. He is a member of the Computer-Aided Network Design (CANDE), a member of ACM, and a member of Sigma Xi.