ANALYTICAL ESTIMATION OF TRANSITION ACTIVITY FOR DSP ARCHITECTURES

Sumant Ramprasad[1], *Naresh R. Shanbhag*[2], *and Ibrahim N. Hajj*[2]
[1] *Dept. of Computer Science,* [2] *Dept. of Electrical and Computer Engineering,*
Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign,
Urbana IL USA 61801.
{*ramprasa, shanbhag, hajj*}*@uivlsi.csl.uiuc.edu*

ABSTRACT

We present an analytical method to estimate the average number of transitions, T, for a given DSP architecture. First, an *exact* relation between the transition activity, probability, and autocorrelation for a 1-bit signal is derived. Next, we estimate word-level signal transition activity using word-level signal statistics, the signal generation model, and the signal encoding. Input signal statistics are propagated through the DSP operators in a system and then employed to estimate T. Simulations with filters result in errors in T between 1% and 12%. Also, the transpose form is shown to have fewer signal transitions than the direct form for the same input.

1. INTRODUCTION

Power dissipation has become a critical VLSI design concern in recent years [2]. Therefore estimation of power dissipation is currently an active area of research. While extensive work has been done for power estimation at the circuit and logic levels, results at the architectural level [3] are preliminary. Architectural level power estimation tools enable the choice between competing architectures and permit major design changes when it is easiest to do so. Since power dissipation in CMOS VLSI circuits depends on the number of signal transitions occuring at the capacitive nodes, estimation of the average number of signal transitions is a key requirement in various power estimation techniques.

The closest approach to our work is [3] where a word is divided into three segments: uncorrelated data bits, correlated data bits, and sign bits. The uncorrelated data bits are defined to be from the least significant bit (LSB) up to a certain break-point BP_0. The transition activity of the sign bits, which are from the most significant bit (MSB) to another break-point BP_1, are measured by an RTL simulation. A linear model is used for the switching activity of correlated data bits, which lie between the sign bits and uncorrelated data bits. Empirical equations defining BP_0 and BP_1 in terms of word-level statistics were also presented in [3].

Our approach is similar in spirit to [3] in that we estimate the average number of transitions, T, in a signal from its word-level statistical description. However, first, in section 2, we derive a relation between the transition activity, probability, and autocorrelation for a 1-bit signal. Then, we employ the word-level signal statistics, the signal generation model, and the number representation to estimate T for a signal. In section 3, we propagate input statistics through commonly used digital signal processing (DSP) blocks. Finally, we estimate T for each signal in a system composed of these DSP blocks and add them up to determine T for the system.

1.1. Preliminaries

Let $x(n)$ be a B-bit signal with *mean* μ, *variance* σ^2, and *lag-i autocorrelation* $\rho(i)$. We will denote $\rho(1)$ by just ρ. Let $\mathcal{X}$ be the set of values that $x(n)$ can assume. Let p_i be the probability that $b_i(n)$, the i^{th} bit of $x(n)$, is 1. If $\mathcal{X}_i$ is the set of elements in $\mathcal{X}$ such that the i^{th} bit is 1, then $p_i = Pr(x(n) \in \mathcal{X}_i)$. Note that p_i can be calculated from the probability distribution of $x(n)$ which, without loss of generality, we assume is known. The *autocorrelation*, ρ_i, of $b_i(n)$ is $\frac{E[b_i(n)b_i(n-1)]-p_i^2}{p_i-p_i^2}$. The *transition activity* [5], t_i, of $b_i(n)$ is $Pr(b_i(n) = 0 \bigwedge b_i(n-1) = 1) + Pr(b_i(n) = 1 \bigwedge b_i(n-1) = 0)$. The word-level transition activity T is defined as:

$$T = \sum_{i=0}^{B-1} t_i. \qquad (1)$$

In this paper, we employ $ARMA$ models [1] to calculate transition activity. An $ARMA(N, M)$ model can be represented as,

$$x(n) = \sum_{i=0}^{N} b_i \gamma(n-i) + \sum_{i=1}^{M} a_i x(n-i) \qquad (2)$$

where $\gamma(n)$ is an uncorrelated noise source with zero mean, and $x(n)$ is the signal being generated. We can rewrite (2) in terms of only the inputs as follows:

$$x(n) = \sum_{i=0}^{\infty} h_i \gamma(n-i), \qquad (3)$$

where h_i can be computed using the following recursion, $h_k = b_k + \sum_{i=1}^{N} a_i h_{k-i}$, where $h_k = 0$ for $k < 0$, and $h_0 = b_0$. An M^{th} order auto-regressive ($AR(M)$) model is an $ARMA(0, M)$ model and an N^{th} order moving-average ($MA(N)$) model is an $ARMA(N, 0)$ model.

In section 2, we will employ the following lemma [4],

Lemma 1 $E[b_i(n)b_i(n-1)] = p_i - \frac{t_i}{2}$

2. WORD-LEVEL TRANSITION ACTIVITY

In this section, we estimate word-level transition activity T of a signal $x(n)$ from its word-level statistics. First, for a single-bit signal $b_i(n)$, we have the following result,

Theorem 1 $t_i = 2p_i(1-p_i)(1-\rho_i)$

Proof: *Substitute for* $E[b_i(n)b_i(n-1)]$ *from Lemma 1 into the definition of* ρ_i *and solve for* t_i. □

We can calculate p_i from the probability distribution of $x(n)$ and substitute in Theorem 1. That leaves the estimation of ρ_i which is described next.

0-7803-3583-X/97 $10.00 ©1997 IEEE

2.1. Estimation of ρ_i: Approximate Method

We will present a computationally efficient, but approximate, method to estimate ρ_i from word-level statistics. This method uses a model similar to that used in [3].

Figure 1 is a plot of ρ_i for the last 7 signals in Table 1. ρ_i is close to 0 for the LSBs and close to ρ for the MSBs. There is also a region in between the LSBs and MSBs where ρ_i increases approximately linearly. As in [3], we divide the word into 3 regions of contiguous bits referred to as the LSB, *linear*, and MSB regions. The break-points BP_0 and BP_1 separate the LSB from the linear region and the linear from the MSB region, respectively. Our approach differs from [3] in the way BP_0 and BP_1 are computed, and our use of Theorem 1 and (1) to compute T *analytically*. We do not employ simulations to estimate transition activity of the MSBs.

For now, we assume that 2's complement representation is being employed. By definition, $\rho_i = 0$ for $i < BP_0$ and $\rho_i = \rho_{BP_1}$ for $i \geq BP_1 - 1$. Hence, we

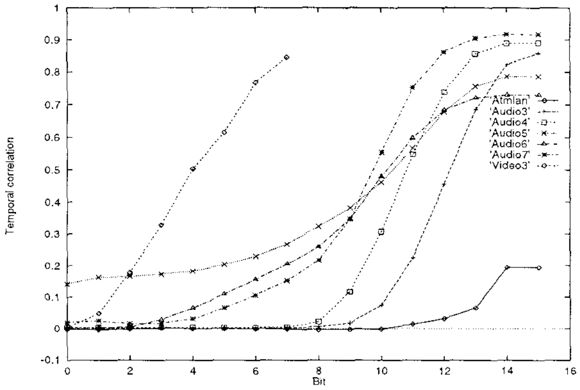

Figure 1. Temporal correlation versus bit

can make the following approximation for 2's complement:

$$\rho_i = \begin{cases} 0 & (i < BP_0) \\ \frac{(i-BP_0+1)\rho_{BP_1}}{BP_1-BP_0} & (BP_0 \leq i < BP_1 - 1) \\ \rho_{BP_1} & (i \geq BP_1 - 1) \end{cases} \quad (4)$$

We now estimate $\{\rho_{BP_1}, BP_0, BP_1\}$ from the word-level statistics, the signal model, and the signal encoding.

2.1.1. *Calculation of BP_0, BP_1, and ρ_{BP_1}*

For an uncorrelated signal, $\gamma(n)$, a good estimate of BP_0 is $\log_2 \sigma_\gamma$, where σ_γ^2 is the variance of $\gamma(n)$ [3]. In general, a stationary signal $x(n)$ can be modeled via an ARMA model (see (3)), which can then be used to calculate BP_0. Since the signals $h_i\gamma(n-i)$ in (3) are uncorrelated, BP_0 for each of the signals can be estimated as $\log_2 |h_i|\sigma_\gamma$. The break-point BP_0 for a signal $x(n) = \sum_i h_i\gamma(n-i)$ can now be estimated as the maximum of the BP_0's of the signals $h_i\gamma(n-i)$. Hence, if $h_{max} = \max(|h_i|)$ and $[k]$ is the integer nearest to k, then

$$BP_0 = [\log_2 h_{max}\sigma_\gamma], \quad (5)$$

We define BP_1 such that for $i \geq BP_1 - 1$, ρ_i is approximately constant. If the values of $x(n)$ lie between the values x_{min} and x_{max}, then $\log_2(x_{max} - x_{min})$ bits are required to cover this range. Hence,

$$\begin{aligned} BP_1 &= [\log_2(x_{max} - x_{min})], \\ &= [\log_2 6\sigma] \text{ (for a normal distr.)}, \quad (6) \end{aligned}$$

Table 1. Signal details

Signal	Description	σ_γ	μ	σ	ρ
SIG1	$x(n) = \gamma(n) - 0.5x(n-1)$	866.00	0.00	1000.00	-0.50
SIG2	$x(n) = \gamma(n) + 0.99x(n-1)$	141.00	0.00	1000.00	0.99
SIG3	$x(n) = \gamma(n) + 0.5\gamma(n-1)$	100.00	0.00	111.80	0.40
SIG5	$x(n) = \gamma(n) + 0.4\gamma(n-1) + 0.2\gamma(n-2) + .07\gamma(n-3) + .5x(n-1) + .3x(n-2) + 0.1x(n-3) + 0.05x(n-4) - .2x(n-5)$	1000.00	0.00	2309.00	0.89
Audio3	2.88MB of 16 bit audio	N/A	1.43	7349.20	0.96
Audio4	2.88MB of 16 bit audio	N/A	-17.63	4040.40	0.97
Audio5	0.37MB of 16 bit audio	N/A	59.46	2661.75	0.90
Audio6	0.61MB of 16 bit audio	N/A	23.61	2328.79	0.96
Audio7	2.88MB of 16 bit audio	N/A	-39.35	3086.30	0.99
ATM LAN	0.80MB of 16 bit comm. data	N/A	0.49	5581.60	0.30
Video3	9.70MB of 8 bit video	N/A	99.71	55.57	0.92

Table 2. T for different number representations

Signal	Unsigned, 2's comp.		1's comp.		Sign magnitude	
	Meas.	Est.	Meas.	Est.	Meas.	Est.
SIG1	8.79	8.82	8.79	8.82	6.07	6.16
SIG2	4.99	5.03	4.99	5.03	4.65	4.74
SIG3	6.97	6.94	6.97	6.94	4.20	4.15
SIG5	6.54	6.42	6.55	6.42	5.91	5.89
Audio3	6.42	6.32	6.43	6.32	6.17	6.24
Audio4	5.80	6.06	5.80	6.06	5.55	5.89
Audio5	4.78	4.40	4.79	4.40	4.22	4.23
Audio6	5.38	5.59	5.38	5.59	4.62	5.43
Audio7	5.05	5.52	5.05	5.52	4.78	5.44
ATM LAN	7.76	7.56	7.76	7.56	7.09	6.94
Video3	2.31	2.15	2.31	2.15	2.16	2.15

where σ^2 is the variance of $x(n)$. When $|\mu| > 3\sigma$ there are 3 regions in which ρ_i is a constant. The first region consists of the bit positions i such that $i < BP_0$. The second region has bit positions i lying between BP_1 and another break-point BP_2, which can be calculated by computing the common MSBs of x_{max} and x_{min}. The third region consists of bits with positions beyond BP_2 where the bits do not have any transitions.

It can be shown that ρ_{BP_1} for $x(n)$ can be calculated from the parameters of the AR(1) or MA(N) model for the signal. On the other hand, we can also assume that $\rho_{BP_1} = \rho$ as can be seen from Figure 1.

2.2. Calculation of T

We computed T for the signals in Table 1 using (1), Theorem 1, (5), and (6). The errors between measured and estimated T are between 1-18%. We assumed $\rho_{BP_1} = \rho$ for the last 7 signals in Table 1. In order to estimate BP_0 we employed AR(1) models for all signals except Audio5 and Video3 for which we employed MA(10) models because AR(1) models resulted in higher errors.

We have so far considered 2's complement representation. The unsigned representation has the same transition activity as 2's complement because the MSBs of the former behave identical to the sign bits of the latter. For positive numbers, 1's complement is identical to 2's complement. For negative numbers, we can generate the 2's complement representation from that of the 1's complement by adding 1 to the LSB, which will usually affect only the LSBs. Since we assume that LSBs are uncorrelated, the activity in 1's complement will be close to that in 2's complement. The transition activity for sign-magnitude is less than or equal to 2's complement because the number of sign bits in sign magnitude representation (1) is less than or equal to that in 2's complement. These conclusions are supported by the results in Table 2.

3. TRANS. ACTIVITY FOR DSP ARCH.

In this section the total transition activity for DSP architectures is estimated by first propagating word-level statistics through the architecture and employing the results of the previous section to estimate T for each signal in the architecture.

3.1. Propagation of Word-Level Statistics

In this subsection, we propagate the input statistics to the output for the following commonly occuring DSP operators: adder, multiplier, multiplexer, and delay.

In Figure 2, the statistics at the output of the adder are given by the following equations.

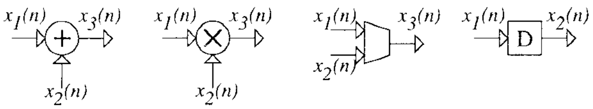

Figure 2. Common DSP operators

$$\begin{aligned}\mu_3 &= E[x_3(n)] = E[x_1(n) + x_2(n)] = \mu_1 + \mu_2\\ \sigma_3^2 &= E[x_3^2(n)] - \mu_3^2 = E[(x_1(n) + x_2(n))^2] - \mu_3^2\\ &= \sigma_1^2 + \sigma_2^2 + 2E[x_1(n)x_2(n)] - 2\mu_1\mu_2\\ \rho_3 &= \tfrac{E[x_3(n)x_3(n-1)]-\mu_3^2}{\sigma_3^2} = \tfrac{E[(x_1(n)+x_2(n))(x_1(n-1)+x_2(n-1))]-\mu_3^2}{\sigma_3^2}\\ &= \tfrac{\rho_1\sigma_1^2+\rho_2\sigma_2^2+E[x_2(n)x_1(n-1)]+E[x_1(n)x_2(n-1)]-2\mu_1\mu_2}{\sigma_3^2}\end{aligned}$$

If $x_1(n) = \sum_{i=0}^{k-1} c_i x(n-i)$ and $x_2(n) = c_k x(n-k)$ as in the case of an FIR filter, then

$$\mu_3 = \mu(\textstyle\sum_{i=0}^{k} c_i) \quad (7)$$

$$\sigma_3^2 = \sigma^2(\textstyle\sum_{i=0}^{k} c_i^2 + 2\sum_{i=0}^{k-1}\sum_{j=i+1}^{k} \rho_0(j-i)c_i c_j) \quad (8)$$

$$\rho_3\sigma_3^2 = \sigma^2(\textstyle\sum_{i=0}^{k-1} c_i c_{i+1} + \sum_{i=0}^{k}\sum_{j=i}^{k} c_i c_j \rho_0(j-i+1) + \sum_{i=0}^{k-2}\sum_{j=i+2}^{k} c_i c_j \rho_0(j-i-1)) \quad (9)$$

In Figure 2, the statistics at the output of the multiplier are given by:

$$\begin{aligned}\mu_3 &= \mathrm{E}[x_3(n)] = \mathrm{E}[x_1(n)x_2(n)]\\ \sigma_3^2 &= \mathrm{E}[x_3^2(n)] - \mu_3^2 = \mathrm{E}[x_1^2(n)x_2^2(n)] - \mu_3^2\\ \rho_3 &= \tfrac{E[x_3(n)x_3(n-1)]-\mu_3^2}{\sigma_3^2} = \tfrac{E[x_1(n)x_2(n)x_1(n-1)x_2(n-1)]-\mu_3^2}{\sigma_3^2}\end{aligned}$$

If $x_2(n)$ is a constant c_1, then $\mu_3 = c_1\mu_1$, $\sigma_3 = c_1\sigma_1$, and $\rho_3 = \rho_1$.

When 2 signals with statistics $\{\mu_1, \rho_1, \sigma_1\}$ and $\{\mu_2, \rho_2, \sigma_2\}$ are alternately multiplexed (Figure 2), the statistics $\{\mu_3, \rho_3, \sigma_3\}$ of $x_3(n)$ at the output of the multiplexer are given by:

$$\mu_3 = E[x_3(n)] = \tfrac{\mu_1+\mu_2}{2} \quad (10)$$

$$\sigma_3^2 = E[x_3^2(n)] - \mu_3^2 = \tfrac{2\sigma_1^2+2\sigma_2^2+(\mu_1-\mu_2)^2}{4} \quad (11)$$

$$\begin{aligned}\rho_3\sigma_3^2 &= E[x_3(n)x_3(n-1)] - \mu_3^2\\ &= E[\tfrac{x_1(n)x_2(n)+x_1(n+1)x_2(n)}{2}] - \mu_3^2 \quad (12)\end{aligned}$$

Finally, the statistics at the output of a delay element are identical to that at the input.

3.2. Example 1: FIR filter

We illustrate propagating word-level statistics using the 5-tap Finite Impulse Response (FIR) filter in Figure 3, where coefficients $c_1 = c_5 = 0.09765625$, $c_2 = c_4 = 0.1953125$, and $c_3 = 0.39453125$. We need the lag-2 through lag-5 correlations of the input in order to propagate the statistics. If they are not available, the lag-i correlation can be approximated by $\rho_0^i(1)$. The statistics of signals in the filter can be calculated using (7), (8), and (9). As an example, the equations for the statistics of the output, $x_{13}(n)$ are:

Table 3. Statistics for direct form FIR filter

Signal	μ		ρ		σ	
	Meas.	Est.	Meas.	Est.	Meas.	Est.
x_0, x_1, x_2, x_3, x_4	99.71	99.71	0.9199	0.9199	55.57	55.57
x_5, x_9	9.74	9.74	0.9183	0.9199	5.46	5.43
x_6, x_8	19.48	19.47	0.9198	0.9199	10.86	10.85
x_7	39.35	39.34	0.9196	0.9199	21.94	21.92
x_{10}	29.23	29.21	0.9529	0.9534	16.03	15.99
x_{11}	68.57	68.55	0.9660	0.9661	37.11	37.06
x_{12}	88.06	88.03	0.9763	0.9764	47.17	47.11
x_{13}	97.80	97.76	0.9811	0.9812	51.99	51.90

$$\begin{aligned}\mu_{13} &= \mu(\textstyle\sum_{i=1}^{5} c_i)\\ \sigma_{13}^2 &= \sigma^2(\textstyle\sum_{i=1}^{5} c_i^2 + 2\sum_{i=1}^{4}\sum_{j=i+1}^{5} \rho(j-i)c_i c_j)\\ \rho_{13}\sigma_{13}^2 &= \sigma^2(\textstyle\sum_{i=1}^{4} c_i c_{i+1} + \sum_{i=1}^{5}\sum_{j=i}^{5} c_i c_j \rho(j-i+1) +\\ &\quad \textstyle\sum_{i=1}^{3}\sum_{j=i+2}^{5} c_i c_j \rho(j-i-1))\end{aligned}$$

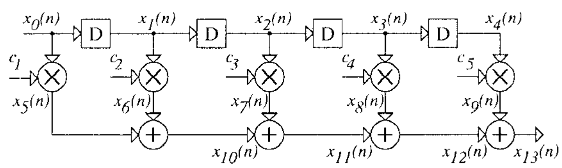

Figure 3. Direct form FIR filter

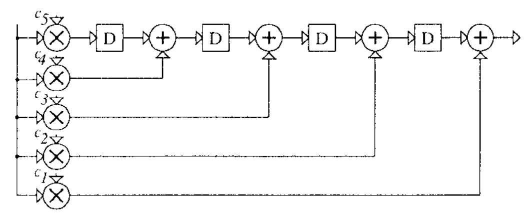

Figure 4. Transpose FIR filter

The error between the measured and estimated word-level statistics for video3 data, shown in Table 3, is less than 1%. Table 4 shows the measured and estimated T for the FIR filter in Figure 3 and its transpose in Figure 4. It can be seen that T for the transpose form is less than that for direct form. This is caused by the lower transition activity at the delays which in turn is because multiplying by a constant of magnitude less than 1 reduces the variance. Reducing the variance reduces BP_0 and BP_1 and hence the transition activity.

3.3. Example 2: Folded FIR filter

The 5 tap FIR filter in Figure 3 containing 5 multiplies and 4 adds can be folded [6] onto 3 multipliers and 2 adders using additional delays and multiplexers as shown in Figure 5. The statistics of the signals of the unfolded filter can be calculated using (7), (8), and (9). These are used along with (10), (11), and (12) to calculate the statistics of signals of the folded filter. As an example, the statistics of the signal, $x_{11,7}(n)$, obtained by multiplexing $x_{11}(n)$ and $x_7(n)$ are given by:

$$\begin{aligned}\mu_{11,7} &= \tfrac{(c_1+c_2+2c_3)\mu}{2}\\ \sigma_{11,7}^2 &= 2(\mathrm{c}_1^2 + \mathrm{c}_2^2 + 2\mathrm{c}_3^2 + 2\mathrm{c}_1\mathrm{c}_2\rho + 2\mathrm{c}_2\mathrm{c}_3\rho + 2\mathrm{c}_1\mathrm{c}_3\rho(2))\sigma^2\\ &\quad +(\mathrm{c}_1 + \mathrm{c}_2)^2\mu^2\\ \rho_{11,7}\sigma_{11,7}^2 &= 2\sigma^2\mathrm{c}_3(\mathrm{c}_1(\rho(2) + \rho) + (\mathrm{c}_2 + \mathrm{c}_3)(\rho + 1)) - (\mathrm{c}_1 + \mathrm{c}_2)^2\mu^2\end{aligned}$$

Table 4. T for FIR filters

Signal	Direct form		Transpose	
	Meas.	Est.	Meas.	Est.
SIG1	148.31	148.92	145.45	145.97
SIG2	76.64	76.40	72.84	72.26
SIG3	113.15	113.64	109.00	109.44
SIG5	104.63	102.10	101.41	98.62
Audio3	102.16	100.76	99.14	98.01
Audio4	91.40	94.37	88.42	90.62
Audio5	75.80	68.42	73.23	66.55
Audio6	84.94	86.63	82.03	83.09
Audio7	78.82	85.65	76.07	82.41
ATM LAN	129.35	124.76	127.94	122.23
Video3	31.58	33.02	28.29	31.64

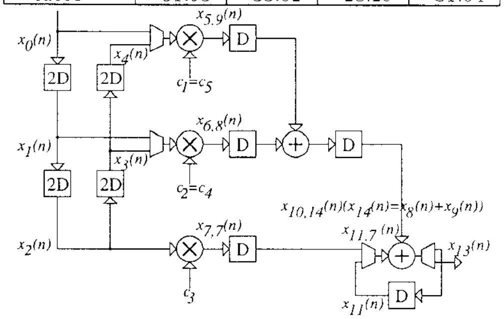

Figure 5. Folded direct form filter

The error between the measured and estimated word-level statistics is less than 1%. The error between the measured and estimated T, shown in Table 5, for the folded FIR filter is less than 8%. T for the folded architecture (Table 5) is higher than that for the original FIR filter (Table 4) because folding increases the number of transitions. This was also observed in [2].

3.4. Example 3: IIR filter

In this example we propagate word-level statistics through the simple Infinite Impulse Response (IIR) filter in Figure 6, where $c_1 = 0.1$. The equations for the statistics of the signals in the direct form IIR filter are:

$$x_3(n) = \sum_{i=1}^{n} c_1^i x_0(n-i)$$

$$\mathrm{E}[x_0(n)x_3(n)] = \mathrm{E}[\sum_{i=1}^{n} c_1^i x_0(n-i)x_0(n)]$$

$$= \frac{c_1\mu_0^2}{1-c_1} + \frac{\sigma_0^2 c_1 \rho_0}{1-c_1\rho_0} \text{ (assuming } \rho_0(i) = \rho_0^i(1))$$

$$\mathrm{E}[x_0(n-1)x_3(n)] = \frac{c_1\mu_0^2}{1-c_1} + \frac{\sigma_0^2 c_1}{1-c_1\rho_0} \text{ (assuming } \rho_0(i) = \rho_0^i(1))$$

$$\mathrm{E}[x_0(n)x_3(n-1)] = \frac{c_1\mu_0^2}{1-c_1} + \frac{\sigma_0^2 c_1 \rho_0^2}{1-c_1\rho_0} \text{ (assuming } \rho_0(i) = \rho_0^i(1))$$

$$\mu_1 = \frac{\mu_0}{1-c_1},\ \sigma_1^2 = \frac{\sigma_0^2 + 2E[x_0(n)x_3(n)] - 2\mu_0\mu_3}{1-c_1^2}$$

$$\rho_1 = \frac{\rho_0\sigma_0^2 + E[x_3(n)x_0(n-1)] + E[x_0(n)x_3(n-1)] - 2\mu_0 c_1 \mu_1}{\sigma_1^2(1-c_1^2)}$$

The error between the measured and estimated statistics is less than 1%. Table 6 shows the measured and estimated T for the direct form IIR filter and its transpose in Figure 6. Again, T is less for the transpose form due to the lower transition activity at the delays.

Table 5. T for folded direct form FIR filter

Signal	Meas.	Est.	% Error
SIG1	202.14	208.39	3.09
SIG2	119.48	123.30	3.20
SIG3	187.04	193.08	3.23
SIG5	166.56	169.78	1.93
Audio3	159.14	158.29	0.53
Audio4	145.40	146.89	1.02
Audio5	123.40	132.94	7.73
Audio6	136.12	138.88	2.03
Audio7	130.60	135.04	3.40
ATM LAN	202.46	207.93	2.70
Video3	51.32	52.95	3.18

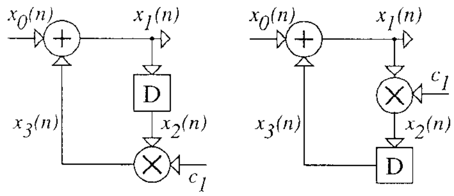

Figure 6. IIR direct form filter and transpose

Table 6. T for IIR filters

Signal	Direct form		Transpose	
	Meas.	Est.	Meas.	Est.
SIG1	35.22	35.52	35.68	35.97
SIG2	18.36	18.21	16.82	16.38
SIG3	26.86	26.92	26.33	27.25
SIG5	24.88	24.38	23.66	23.26
Audio3	24.36	24.26	22.92	22.56
Audio4	21.82	22.49	20.34	20.78
Audio5	18.06	17.15	16.99	15.55
Audio6	20.29	20.85	19.02	19.39
Audio7	18.87	20.33	17.38	18.59
ATM LAN	30.59	29.25	30.17	28.68
Video3	7.69	7.74	6.22	6.93

3.5. Run Time

Table 7 compares the run time for simulation and that for the approximate method on a 85 MHz SparcStation 5. In most cases the run time for the approximate method is an order of magnitude less than that for simulation. The simulation time depends on the length of the input sequence whereas for the approximate method it depends on the signal width (8-bit for video3 and 16-bit for the rest). The run time for the approximate method can be further reduced by optimizations such as setting the transition activity at the output of a delay to be equal to that at its input.

REFERENCES

[1] M. G. Bellanger, "Adaptive Digital Filters and Signal Analysis," Marcel Dekker, 1987.

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

[3] P. E. Landman, J. M. Rabaey, "Architectural Power Analysis: The Dual Bit Type Method," *IEEE Trans. VLSI Syst.*, pp. 173–187, June 1995.

[4] Tan-Li Chou, K. Roy, S. Prasad, "Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching," *Proc. 1994 Intl. Conf. on Computer-Aided Design*, pp. 300–303.

[5] F. Najm, "Transition density, a new measure of activity in digital circuits," *IEEE Trans. on Computer-Aided Design*, pp. 310–323, Feb. 1993.

[6] K. K. Parhi, Ching-Yi Wang, A. P. Brown, "Synthesis of Control Circuits in Folded Pipelined DSP Architectures," *IEEE Journal of Solid-State Circuits*, pp. 181–195, January 1992.

Table 7. Comparison of run times

Signal	Simulation (sec)	Approximate method (sec)
Audio3	42.30	6.85
Audio4	40.28	6.85
Audio5	5.00	6.93
Audio6	8.60	6.98
Audio7	39.16	6.96
ATM LAN	13.05	6.83
Video3	138.91	0.10