

# Power Estimation for High Level Synthesis

Paul E. Landman

Jan M. Rabaey

Electrical Engineering Department  
University of California, Berkeley  
Berkeley, California 94720, USA

## Abstract

*This paper describes techniques for rapidly and accurately estimating power consumption based on high level descriptions of system architectures. This novel approach, based on stochastic modeling of bus statistics, achieves the accuracy traditionally associated with gate and circuit level estimation tools while exploiting the reduced computational complexity offered by the architectural level of abstraction. Indeed, the results presented here indicate an estimation accuracy within 9.4% of gate level simulations, while existing high level techniques can be off 80% or more.*

## 1 Introduction

Recently, the concept of providing personal communication and computing services in a portable environment has stirred a great deal of interest in both commercial and research arenas. An integral component of the proposed personal communications system (PCS) would be a portable multi-media terminal [1]. Unfortunately, as with all portable, battery-operated applications, power dissipation becomes a critical issue.

Most contemporary design techniques and tools, however, give only passing consideration to power minimization, concentrating instead on delay and area optimization [4][8]. The increasing importance of portable applications such as PCS will, therefore, force designers to reevaluate performance criteria for this growing class of systems. Moreover, these new design objectives will propel designers to seek new methodologies specifically targeted at low-power VLSI systems.

Clearly, a low-power CAD framework is required to support this endeavor. Ideally, these tools should provide not only for the exploration of issues related to low-power design, but also for the automated synthesis of low-power systems. Not surprisingly, the success of this environment will depend critically on the availability of fast and accurate estimation of the key design space parameters: area,

delay, and power. While high level area and delay estimation have received substantial consideration, the issue of power estimation is almost completely unexplored above the gate level.

This paper presents a battery of novel techniques that extend the domain of accurate power estimation to the higher architectural design level. These techniques will be incorporated into the Hyper high level synthesis system [8], realizing the goal of computer-assisted design space exploration in the area, delay, and power dimensions. As the Hyper system is intended for high performance, datapath-intensive architectures arising in applications such as telecommunications, speech, and image processing, much of this paper will discuss techniques for power estimation within datapath modules.

## 2 Previous work

Traditionally, power estimation has been performed at the circuit level, relegated to such low level tools as SPICE (transistor level) and IRSIM (switch level). More recently, gate level tools based on probabilistic estimation have evolved [3][6]. The majority of these tools propagate signal probabilities through gate level networks to arrive at transition density estimates for the various nodes in the circuit. Combining this information with capacitive loading estimates, the tools arrive at a figure for the expected gate level power consumption. Of course, the problem becomes more difficult when reconvergent fan-out introduces correlations between input signals. Under these circumstances, the tools typically apply heuristics that trade off accuracy for speed in order to obtain the necessary transition probability estimates. An example of such a technique is the Correlation Coefficient Method [2] in which pairwise signal correlations are stored and used in probability calculations, while higher order correlations are ignored.

As intimated above, however, the main goal of this research is to facilitate high level synthesis of low-power systems. Unfortunately, the application of gate level power

estimation tools to the high level synthesis problem would require that each implementation be mapped down to the gate level - a time consuming task. Moreover, gate and circuit level estimation are themselves computationally intensive considering the explosion of gates resulting from expansion of even the smallest systems. Thus, it is not feasible to consider using gate and circuit level estimation tools to provide the fast, iterative estimation required for system level power minimization.

The need for a higher level of abstraction in power estimation is clearly indicated. Powell addressed this issue in [7] where he presented an architectural level estimation technique referred to as the Power Factor Approximation (PFA) Method. In this paper, Powell develops power estimates for various architectural modules (e.g. multipliers, RAM's, etc.) parameterized by bit width. Proportionality constants are then extracted through physical measurements or simulations using *independent white noise inputs*. Consequently, this techniques does not account for the strong dependency of power consumption on the statistics of the input data as the results of section 4 illustrate.

### 3 Proposed power estimation methodology

It is interesting to note that proven techniques for high level area and delay estimation exist [4][5], while accurate power estimation at this level is relatively unexplored. The explanation is two-fold: first, power has only recently emerged as a critical design concern and, second, power consumption is extremely data dependent and deterministic modeling techniques do not apply as readily as in the time and area domains. As a result, stochastic modeling techniques will be critical for high level power estimation.

Using concepts abstracted from the gate level, we've developed algorithmic and architectural estimation techniques based on high level statistics such as mean, variance, and autocorrelation. Thus, while gate level tools focus on the power consumed by boolean logic gates (NOT, AND, OR, etc.) as a function of their input bit probabilities, this research considers module (adder, multiplier, register, etc.) power consumption as determined by input word statistics.

#### 3.1 Stochastic word level data model

Not surprisingly, there exists a direct relationship between bit level probabilities and word level statistics. This correlation is illustrated in Figure 1 for three common DSP input signals: speech, music, and image. For each input, the right hand portion of the figure shows both signal and transition probabilities for each bit of the input data word (two's-complement assumed). Clearly, all three signals follow a similar pattern which can be exploited to

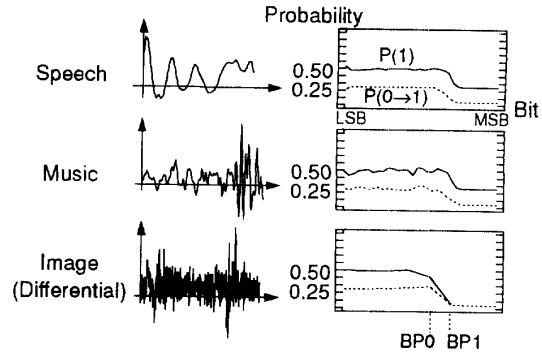


Figure 1. Stochastic word level data model

yield a simple piecewise linear model with breakpoints BP0 and BP1. The important features of this model - that is, the values of the breakpoints and the signal and transition probabilities - can all be extracted from three statistical parameters: the mean,  $\mu$ ; the variance,  $\sigma^2$ ; and the lag one correlation coefficient,  $\rho_1 = \text{cov}(X_t, X_{t+1})/\sigma^2$ .

Consider first the low-order region of the model from the LSB to BP0. As might be expected, these bits are uncorrelated in space and time and are essentially independent of the data distribution, having signal probabilities of 1/2 and transition probabilities of 1/4:

$$P_{\text{lsb}'s}(1) = 1/2, P_{\text{lsb}'s}(0 \rightarrow 1) = 1/4 \quad (1)$$

As an illustration of this fact, consider the LSB which has the interpretation of being zero for even data values and one for odd. A signal probability of 1/2 for this bit, then, indicates an equal likelihood of an even or odd data value at a given instant in time - clearly, a reasonable assumption for continuous, smoothly varying input distributions. The value of the breakpoint denoting the end of this low-order region is related to the spread, or variance, of the signal distribution and is given empirically by:

$$\text{BP0} = \log_2(3\sigma/32) \quad (2)$$

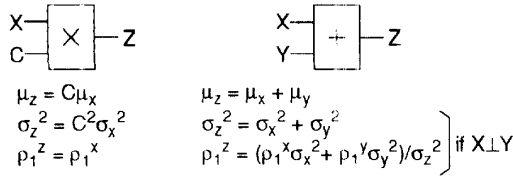
Strictly speaking, this formula is exact only for the *signal* probability curve. The breakpoint for the *transition* probability curve, although typically nearby, experiences an offset towards the LSB for highly correlated input signals:

$$\Delta\text{BP0} = \log_2(1-\rho_1^2)^{0.5} \quad (3)$$

In two's-complement representation, the purpose of the high-order bits is that of sign extension. As a result, the bits in this region exhibit complete dependence, having signal and transition probabilities that are functions of the mean, variance, and first-order correlation coefficient of the data word:

$$P_{\text{msb}'s}(1) = P(-) = F_1(\mu/\sigma) \quad (4)$$

$$P_{\text{msb}'s}(0 \rightarrow 1) = P(+ \rightarrow -) = F_{01}(\mu/\sigma, \rho_1) \quad (5)$$



Revised Adder Equations for Correlated Inputs:

$$\begin{aligned} \mu_z &= \mu_x + \mu_y \\ \sigma_z^2 &= \sigma_x^2 + 2\rho^{xy}\sigma_x\sigma_y + \sigma_y^2 \\ \rho_1^z &= (\rho_1^x\sigma_x^2 + [\rho_1^{xy} + \rho_1^{yx}]\sigma_x\sigma_y + \rho_1^y\sigma_y^2)/\sigma_z^2 \end{aligned}$$

**Figure 2. Statistical parameter propagation**

The exact probabilities depend, of course, on the distribution of the signal; however, noting that many typical DSP inputs are closely approximated by Gaussian processes, we substitute the univariate and bivariate normal distribution functions for  $F_1$  and  $F_{01}$ , respectively. Experimental justification for this approximation is provided in section 4 where power estimates for decidedly non-Gaussian input signals are shown to be within 9.4% of simulations. The breakpoint for the sign extension region is determined by the maximum extent of the signal distribution into either positive or negative values and is given specifically by:

$$BP1 = \log_2(|\mu| + 3\sigma) \quad (6)$$

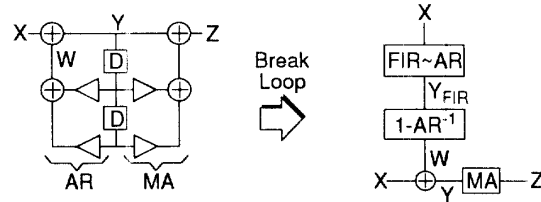
Consider now the range of bits between BP0 and BP1. The correlation of these mid-range bits falls between the extremes represented by the low and high-order regions and as a result a linear approximation for the probabilities in this transition region models the situation well.

### 3.2 Calculation of model parameters

Having developed a model relating word level parameters to bit level probabilities, it is now necessary to consider techniques for computing the values of the required parameters. Reminiscent of gate level methodologies, we first consider a propagation approach. In particular, given the statistics of the module inputs we calculate the statistics of its outputs. By propagating statistics in this fashion, we are able to derive the model parameters ( $\mu$ ,  $\sigma^2$ , and  $\rho_1$ ) for each bus in the architecture (ignoring, for the moment, reconvergent fan-out and feedback). The top half of Figure 2 presents the appropriate propagation equations for the case of an addition and a constant multiplication (the two key operations of linear, time-invariant systems). Once again, the issue of reconvergent fan-out tends to complicate matters by introducing correlations between module inputs which is not handled by these equations; however, we can overcome this difficulty by abstracting heuristic techniques similar to those applied at the gate level. For example, the revised equations at the bottom of

Figure 2 account for pairwise crosscorrelation terms by extending the gate level Correlation Coefficient Method.

The existence of feedback in the flowgraph introduces yet another level of complexity to parameter calculation. Referring to the left half of Figure 3, we see that in order to find the statistics of Y we need prior knowledge of the statistics of W and vice-versa. The solution lies in breaking the loop. One technique, illustrated in the right half of Figure 3, depends upon constructing a FIR approximation to the AR, or feedback, portion of the filter. With X as an input, this non-recursive filter produces an initial approximation to the signal Y. Having broken the loop, we now calculate the statistics of W by propagating the statistics of  $Y_{FIR}$  through the FIR network given by  $1-AR^{-1}$ . In this way, computing model parameters for buses within a feedback network becomes a problem of propagating parameters through a straightforward FIR network.



**Figure 3. Feedback and parameter propagation**

Propagation of model parameters is, however, but one method of deriving the required signal statistics. Another technique would be to use symbolic equation manipulation to evaluate the transfer function from the primary input to the internal nodes of the flowgraph. For given these transfer functions, all of the required bus statistics follow directly. System level functional simulation of the flowgraph is yet another straightforward approach to parameter calculation. Under this regime, internal bus statistics are simply accumulated during simulation for a typical stream of input data. While prohibitively expensive at the gate level, the reduced complexity offered by the architectural level of abstraction might make this a particularly attractive alternative. In addition, the applicability of this approach to non-linear, time-varying systems further increases its desirability.

### 3.3 Interconnect and module energy

Once internal bus statistics are known, the stochastic model of section 3.1 provides the transition probabilities for the bits in the low-order, high-order, and transition regions. These probabilities can then be applied directly to yield an estimate of the energy required to drive data with the given statistics onto a bus with load capacitance, C:

$$E_{bus} = \{0.25N_{lsb} + (0.25 + F_{01})N_{mid}/2 + F_{01}N_{msb}\} CV^2 \quad (7)$$

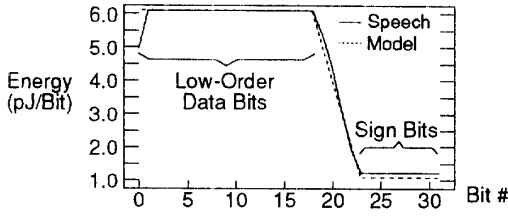


Figure 4. Interconnect energy

where  $N_A$  is the number of bits in region A of the model. Figure 4 illustrates the methodology and demonstrates an estimation accuracy within 2.4% for a speech signal being driven onto a 1 pF bus. Power consumed in the pad drivers can also be calculated in this manner by choosing C to include external loading capacitances.

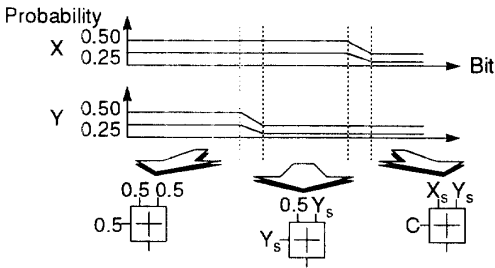


Figure 5. Adder energy calculation regions

Estimating module dissipation, however, requires deeper consideration. As an example, consider the average energy consumed by an adder module. Each of the two inputs follows the piecewise linear model derived above resulting in the four significant module breakpoints shown in Figure 5. These breakpoints divide the addition into five regions: three constant-probability regions and two intervening transition regions. Once again, energy consumption is calculated on a region-by-region basis. Since the least significant bits of each adder input are independent with probability 1/2 (see section 3.1), the energy calculation for the low-order region is greatly simplified. Indeed, the energy for a one-bit addition in this region can be pre-calculated and simply multiplied by the number of bits up to the first breakpoint:

$$E_{+,lsb} = E_{+,1-bit} N_{lsb} \quad (8)$$

In contrast, the energy required for additions in the high-order region cannot be pre-calculated but, instead, depends upon the input statistics. We can, however, exploit the dependence among the sign extension bits of the inputs, realizing that each of the one-bit additions in this region is completely identical. Therefore, in order to calculate the energy required to perform the high-order bit additions, we need only calculate the energy consumed by a one-bit full adder with inputs  $X_s$ ,  $Y_s$ , and C, where  $X_s$

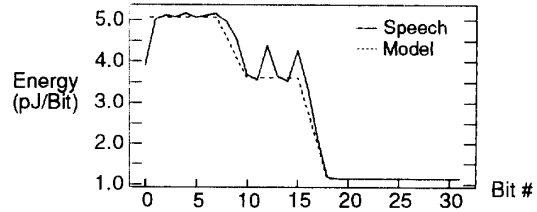


Figure 6. Adder module energy

and  $Y_s$  are the sign bits of inputs X and Y, respectively, and C is the carry input. If  $I_t = (X_s, Y_s, C)_t$  represents the inputs to the adder at time t, then one can show that there are only six possible values of  $I_t$ . From this it follows that there are 36 possible input transitions for each addition in this region. The energy required for each of the 36 cases may be pre-calculated through simulation or physical measurement. Then by weighting each energy by the probability that a particular input transition occurs, we come up with an energy estimate for the addition of the sign extension bits:

$$E_{+,msb} = N_{msb} \sum_{j=1}^{36} P(I_t^j \rightarrow I_{t+1}^j) E(I_t^j \rightarrow I_{t+1}^j) \quad (9)$$

where each input transition probability is calculated from the quadrivariate normal distribution.

The final constant probability region of Figure 5 occurs where the low-order region of X overlaps the sign extension region of Y. Each one-bit addition in this region is characterized by having one input equal to  $Y_s$  and the other an independent probability 1/2 boolean variable, Z. The carry inputs for the bits in this region are not strictly identical; however, it can be shown that they converge rapidly (geometrically) to  $Y_s$ . Thus,  $I_t = (Y_s, Z, Y_s)_t$  and the energy is calculated in a manner consistent with the high-order region summing over the 16 possible input transitions:

$$E_{+,mid} = N_{mid} \sum_{j=1}^{16} P(I_t^j \rightarrow I_{t+1}^j) E(I_t^j \rightarrow I_{t+1}^j) \quad (10)$$

Finally, energies for the two transition regions are approximated by linear interpolation based on the surrounding constant probability regions as shown in Figure 6, which demonstrates a typical estimation error of only 3.2%.

The adder is, of course, but one of the important datapath modules that must be considered; however, similar techniques can be applied to characterize the energy consumption of other key modules such as multipliers, comparators, registers, and shifters. The case of variable-constant multiplication,  $Y = CX$ , is a particularly straightforward example since typical array multipliers are realized by a series of additions for which we already know how to compute the energies:

$$Y = \sum_{j=0}^{N-1} 2^j C_j X \quad (11)$$

To this point, we have considered power estimation only for datapath components. Processing systems, of course, contain other components such as PLA's, memories, and controllers; however, as previously stated our target application range emphasizes mainly high throughput, datapath-intensive real-time systems with, typically, minimal control requirements [8]. As a result, power consumption in the datapath blocks and at the pads will account for a large fraction of the overall power budget. Non-datapath power consumption cannot, of course, be ignored and we are currently developing strategies for estimating power in these modules as well.

As an example, we consider the issues involved in estimating the energy consumption of a static single-ported RAM with differential bit-lines. The total power consumption is accounted for by five main elements: row decoders, word-line drivers, bit-line drivers, sense amps, and column decoders. Energy consumed in the charging and discharging of the bit-lines typically dominates and is proportional to the total number of bits in the memory. In contrast, energy for the sense amps and column decoders is proportional to the width of the memory as is the energy required to drive the word-lines. Finally, the row decoder energy is approximately proportional to the number of words in the memory. Combining all of these factors and realizing that the total memory power consumption is proportional to the access frequency, we arrive at an accurate estimate for power dissipation in a RAM memory module. It is also interesting to note that for a differential bit-line structure, this figure is fairly independent of the value being transferred since, regardless of the data, one bit-line is always charging while the other discharges.

#### 4 Results

The previous section presented novel techniques for architectural level power estimation that combined the speed of high level approaches, such as the PFA method, with the accuracy of gate and circuit level tools. This section, through two examples, will present results validating the approach and demonstrating its advantages over existing techniques. The first example consists of a low-power, three-tap FIR filter comprised of three multipliers, two adders, and two registers. Using a 2 $\mu$ m CMOS technology, the required sample rate of 3 MHz was achieved with of voltage supply of 1.95V (reduced to minimize power consumption). For a variety of input distributions and signals, Table 1 shows the actual power consumption (from gate level simulation) as well as the error in the estimated value obtained both from the PFA method and from the proposed approach.

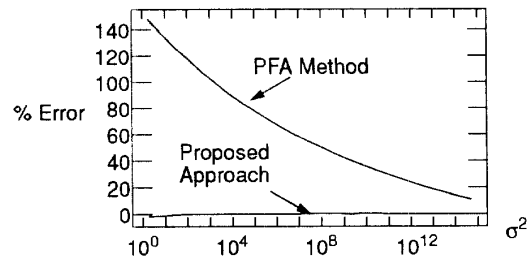
From this table, several key points become apparent. First, the previously mentioned Gaussian assumption is

clearly non-critical. For example, in the case of a uniformly distributed input, which is far from Gaussian, the

Input	Gate Level	PFA	Proposed
Gaussian	96.7 pJ	+61%	-1.7%
Uniform	95.0 pJ	+64%	+2.3%
Laplacian	92.0 pJ	+69%	+3.9%
Speech	92.0 pJ	+69%	+6.5%
Music	83.9 pJ	+86%	+9.4%
Image (Diff)	60.3 pJ	+4.9%	-5.7%

**Table 1: Results for three-tap FIR example**

proposed approach still achieves excellent results that match simulations to within 2.3%. Thus, although the proposed methodology employs certain Gaussian approximations, it still achieves very good results even for input signals that are far from Gaussian in distribution. Furthermore, the results demonstrate that the proposed approach generally outperforms the PFA method significantly in terms of accuracy. This is accounted for by the fact that the PFA method assumes a fixed power consumption for each module type ignoring the influence of input statistics and correlations on power. The table also demonstrates that the method performs equally well on real-world signals such as speech, music, and image that are typically found in DSP applications. It is interesting to note that the only case where the PFA method matches the accuracy of our approach is for a differentially-coded image signal input. This is quite natural, however, since this input has a near white spectrum and, thus, is not adversely affected by the failure of the PFA method to account for signal correlations. Figure 7 further emphasizes the applicability of the proposed approach over a wide range of input statistics as well as the limitations of existing approaches.



**Figure 7. Estimation error versus input power**

The second example is that of a biquadratic filter, again operated at 1.95V to facilitate comparisons. This larger example consisting of four multipliers, four adders, and two registers illustrates the ability of the method to handle feedback, and results are tabulated in Table 2. Interpretations similar to those of the first example again apply with all cases achieving estimation errors of 3.6% or less.

Input	Gate Level	PFA	Proposed
Gaussian	197.3 pJ	+41%	-0.87%
Uniform	197.6 pJ	+41%	-0.56%
Laplacian	197.9 pJ	+41%	-1.6%
Speech	208.3 pJ	+34%	-2.0%
Music	197.7 pJ	+41%	+3.6%
Image (Diff)	123.0 pJ	+3.3%	-2.9%

Table 2: Results for biquad example

## 5 Applications

The previous examples begin to suggest some of the power of the proposed high level power estimation techniques; however, further discussion of the possible applications of the approach is warranted. A high level synthesis environment equipped with power estimation capabilities could be used to observe the effects of transformations, module selection, partitioning, scheduling, and supply voltage scaling on delay, area, and power. As an illustration of this point consider the effect of varying the number of pipeline stages in the previous three-tap FIR example. Since each stage of pipelining reduces the critical path, the filter is able to operate at progressively lower voltage supplies, while maintaining identical throughput, as pipelining is increased. With only the input signal statistics and the flowgraph as inputs, the proposed power estimation tool could provide a plot of the power-area design space, graphically depicting this effect for the designer.

Moreover, the ability to rapidly observe the effects of such high level trade-offs on power would facilitate the development of additional low-power design methodologies similar to the pipelining technique discussed above. Indeed, one can envision automated synthesis of low-power systems, where by defining a cost function that heavily weights power consumption, one could initiate an automated search of the design space leading to an optimal, low-power solution.

Finally, it is important to note that the above power estimation techniques are not confined to static CMOS technologies. On the contrary, since power consumption in static logic depends on input transition probabilities, it is actually a more difficult problem than the dynamic case which can be evaluated purely on the basis of signal probabilities. Thus, the estimation techniques discussed above are also useful for comparing power consumption across a variety of different logic styles and circuit technologies.

## 6 Conclusions

Clearly, with the recent flood of interest in portable VLSI applications, such as personal communications systems, the importance of power consumption in system

design will dramatically increase. With this new emphasis comes the need for tools that support low-power design.

It is our contention that accurate high level power estimation is central to a useful CAD environment for exploring and automating low-power system design. While accurate power estimation has been previously confined to the circuit and gate levels, we have presented a battery of techniques to realize fast and accurate estimation at the higher architectural level of abstraction. Thus far, the results of this research are promising, however, several issues demand further consideration. Among these are estimation techniques for handling: non-linear, time-varying systems; non-datapath components (e.g. controllers); glitching in static circuits; and non-stationary signals.

The tools we develop in this area will support exploration of algorithmic, architectural, and gate level issues in low-power design, as well as, provide the foundation for the automated synthesis of low-power systems, thus, facilitating a complete three-dimensional design space exploration, not only of delay and area, but also of power.

## Acknowledgements

This project was sponsored by a fellowship from the National Science Foundation as well as DARPA grant J-FBI 90-073.

## References

- [1] R. W. Brodersen et al, "Technologies for Personal Communications," *Proceedings of the VLSI Symposium '91 Conference*, Japan, pp. 5-9, 1991.
- [2] S. Ercolani et al, "Estimate of Signal Probability in Combinational Logic Networks," *IEEE European Test Conference*, pp. 132-138, 1989.
- [3] A. Ghosh et al, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," *Proceedings of the 29th Design Automation Conference*, pp. 253-259, 1992.
- [4] R. Jain, "High Level Area-Delay Prediction with Application to Behavioral Synthesis," *Technical Report 89-23*, University of Southern California, 1989.
- [5] F. Kurdahi et al, "Techniques for Area Estimation of VLSI Layouts," *IEEE Transactions on CAD*, Vol. 8, No. 1, pp. 81-92, January 1989.
- [6] F. Najm, "Transition Density, a Stochastic Measure of Activity in Digital Circuits," *Proceedings of the 28th Design Automation Conference*, pp. 644-649, 1991.
- [7] S. R. Powell et al, "Estimating Power Dissipation of VLSI Signal Processing Chips: The PFA Technique," *VLSI Signal Processing IV*, pp. 250-259, 1990.
- [8] J. M. Rabaey et al, "Fast Prototyping of Datapath-Intensive Architectures," *IEEE Design & Test of Computers*, pp. 40-51, June 1991.