High-Level Current Macro-Model for Logic Blocks

Srinivas Bodapati, *Member, IEEE,* Farid N. Najm, *Fellow, IEEE*

Abstract— We present a frequency domain current macro-modeling technique for capturing the dependence of the block current waveform on its input vectors. The macro-model is based on estimating the *Discrete Cosine Transform* (DCT) of the current waveform and then taking the inverse transform to estimate the time domain current waveform. The DCT of a current waveform is very regular and closely resembles the DCT of a triangular or a trapezoidal wave. We use this fact and the relation between the DCT, DFT, DTFT and the Fourier Transform to *infer* template functions for the current macro-model. These template functions are characterized using various parameters like *amplitude, phase decay factor, time period* etc. These parameters are modeled as functions of the input vector pair using regression. Regression is done on a set of current waveforms generated for each circuit, using HSPICE. These template functions are used in an automatic characterization process to generate current macro-models for various CMOS combinational circuits.

Index Terms— Current estimation, Power grid

The minimum feature size of very large scale integrated (VLSI) circuits is continuously shrinking, and at the same time device densities are increasing several-fold, leading to increased power dissipation in these circuits. This large power dissipation has a significant impact on the performance (speed, battery life), reliability and economic cost of the integrated circuits. Therefore, from both performance and cost point of view power management has become a first order concern in the design of VLSI circuits. The most common approach to power management is to reduce the supply voltage. However, while this approach has prevented power levels from increasing as rapidly as they might have otherwise, it has resulted in a significant increase in supply currents. Today's high-end microprocessors, for example, can consume supply current of over 100 Amperes and it is getting worse. The International Technology Roadmap for Semiconductors (ITRS-2001) indicates that the total power supply current delivered to a high performance integrated circuit will grow to about 200 Amperes in 2006, and almost 500 Amperes in 2012. This is the result of increasing power consumption and decreasing supply voltage.

These large supply currents coupled with the lower power supply voltages and thinner wires employed in deep submicron designs adversely effect the robustness of the power grid. The power grid becomes more susceptible to large voltage drops at the devices due to high instantaneous currents, a phenomenon known as IR drop. The power supply lines also become more prone to the formation of voids and shorts to nearby wires due to *electromigration* caused by the large sustained currents. A large IR drop slows down the devices significantly, reduces the noise margin of circuits which might lead to functional failures or soft errors. Thus large currents in the power grid affect the reliability as well as performance of the circuit. As

a result, power grid design and analysis is now an important concern during chip design, not only for power dissipation and reliability reasons, but for performance reasons as well. Therefore, it is important to do early design planning of the power grid, in order to reduce the chance of having to redesign large parts of it [1]. To enable this, in an environment where design blocks are being reused (hard IP blocks) one needs block-level current models that can give the current waveform drawn by a logic block in response to a given input vector stream. With these current models, one can perform early and fast block-level analysis (simulation) of the currents and voltages in the power grid. These models can also be used to speed up post-layout power grid verification of a chip, where some hard IP blocks are being used.

In this regard we propose a bottom-up current waveform macro-model for logic blocks. We work at a level of abstraction that may be called structural RTL: the circuit is described as an interconnection of flip-flops and Boolean (combinational) logic blocks. In this context, we develop a cycle-based model for the current waveform of each combinational block that captures the dependence of the current waveform per-cycle on the vector pair applied at the block inputs. Models for larger sequential circuits may be built by composition of our models for the combinational blocks and simple cell level models for the memory elements. We assume that one is reusing a previously designed logic block (a hard IP block), so that all the internal structural details of the circuit are known. Previous work in bottom-up macro-modeling has targeted either the average power [2, 3, 4, 5, 6] or energy-per-cycle [7, 8]. Current waveform macro-modeling is difficult because of the large variations that are possible in current waveform shapes, and due to the very large number of possible vector pairs. To overcome this problem, we have developed an approach for current waveform modeling that is based on a transformation to the frequency domain. While time-domain waveform shapes are highly varied, it turns out that their frequency-domain transforms are *not*. Specifically, large variations in waveform shapes in the time domain translate to variations mostly in the values of *parameters* of the frequency-domain transforms, but not in their overall shape. Given a certain transform, we propose to construct a model that captures the dependence of its parameters on the input vector pairs. We have found that one can use low-order polynomial models to capture this dependence, and we use regression to compute the coefficients of these polynomials, based on a number of randomly generated vector pairs for which the circuit is simulated with a circuit simulator, in a process that is similar to cell library characterization. Given the vector pair at the circuit input, the model gives the parameter values and thus the frequency transform, which we then invert to obtain the time-domain current waveform.

More specifically, we use Discrete Cosine Transform (DCT), to construct the current macro-models. The reasons for using DCT will become clear in subsequent sections. In our approach a *template* equation is used to capture the DCT of the per-cycle current waveform. The form of this template equation is not exactly *derived*, but *inferred* by examining the forms of the frequency transforms of two types of current waveform shapes that one commonly sees in practice. In the next section, we will describe the frequency transforms that will be used as the basis for inferring a DCT model template and, in section II, we will illustrate what typical shapes these transforms take in practice. In section III, we will describe the DCT model template that we have used, and describe the model construction process and present an automatic macro-model characterization flow. Finally, in section IV we include experimental results that illustrate the validity of this approach.

I. DISCRETE COSINE TRANSFORM

The current waveform obtained from circuit simulation (HSPICE) is a discrete-time signal, which can be obtained from the periodic sampling of a continuous-time current waveform, i.e,

$$i[n] = i_c(nT), \quad 0 \le n \le N-1 \quad (1)$$

where N is the length of the current sequence, $i_c(\cdot)$ is the continuous-time current waveform and T is the *sampling period*, whose reciprocal is the *sampling frequency* [9]. In case of circuit simulation, T is equal to the (fixed) time step specified in the transient analysis (we have used a time step of 0.01 ns). The 1-dimensional Discrete Cosine Transform [10]. (DCT) of a sequence $\{i[n], 0 \le n \le N-1\}$ is defined as:

$$I[k] = \alpha(k) \sum_{n=0}^{N-1} i[n] \cos\left[\frac{\pi(2n+1)k}{2N}\right], \quad 0 \le k \le N-1 \quad (2)$$

where $\alpha(0) = \sqrt{1/N}$, $\alpha(k) = \sqrt{2/N}$, for $1 \le k \le N-1$, and the inverse transformation is given by:

$$i[n] = \sum_{k=0}^{N-1} \alpha(k) I[k] \cos\left[\frac{\pi(2n+1)k}{2N}\right], \quad 0 \le n \le N-1 \quad (3)$$

In order to gain some insight into the form that the DCT current macro-model may take, we look at the analytical forms of the Fourier transform (FT) corresponding to simplified representations of typical current waveforms. For example, we consider a piecewise-linear triangular current waveform and construct its FT as an analytical expression. The *form* of this FT will suggest what forms we should use in our DCT model template. We use FT to get an insight into DCT because there is a relationship between the two transforms, as follows: the DCT is closely related to the DFT (Discrete Fourier Transform), which is a sampled form of the DTFT (Discrete Time Fourier Transform), which is itself related to the FT. We will now explain these relationships.

The N-sample DCT is related to a $2N$-sample DFT [10], as follows. The DCT of an input sequence of N-samples can be obtained by extending the input to a $2N$-sequence sample

with even symmetry, taking a $2N$-point DFT, and saving N terms of it. The even extension of $i[n]$ is defined as:

$$i'[n] = \begin{cases} i[n] & n = 0, 1, \ldots, N-1 \\ i[2N-1-n] & n = N, N+1, \ldots, 2N-1 \end{cases} \quad (4)$$

and the $2N$-point DFT of $i'[n]$, is given by:

$$\begin{aligned} I'[k] &= \frac{1}{\sqrt{2N}} \sum_{n=0}^{2N-1} i'[n] e^{-j(2k\pi n/2N)} \\ &= \frac{1}{\sqrt{2N}} e^{(jk\pi/2N)} \sum_{n=0}^{N-1} i[n] \cos\left[\frac{(2n+1)k\pi}{2N}\right] \quad (5) \end{aligned}$$

which shows that the $2N$-point DFT and the DCT are closely related except for the constants and the extra phase term. The relationship between the DFT and the DTFT [9], is as follows. The DFT of a sequence, $\{i'_n\}_{n=0}^{2N-1}$, is a set of evenly spaced samples of the DTFT over the frequency range 0 to 2π, multiplied by a constant factor to make the DFT an orthonormal transform. Thus:

$$I'[k] = \frac{1}{\sqrt{2N}} I_d(\omega)\big|_{\omega=\frac{2\pi k}{2N}} \quad k = 0, 1, \ldots, 2N-1 \quad (6)$$

where $I_d(\omega)$ is the DTFT of the sequence $\{i'_n\}$, defined by:

$$I_d(\omega) = \sum_{n=0}^{2N-1} i'[n] e^{-j\omega n} \quad (7)$$

where ω is the frequency in radians. Finally, if we consider that the sequence $\{i'_n\}$ is obtained by sampling from an even-extended continuous time current waveform, then the relationship between $I_d(\omega)$ and the Fourier transform (FT) of the continuous waveform, denoted by $I_c(\Omega)$, is given by [9]:

$$I_d(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} I_c\left(\frac{\omega + 2\pi n}{T}\right) \quad (8)$$

where Ω has been replaced by $\left(\frac{\omega+2\pi n}{T}\right)$. When $n = 0$, which corresponds to a DTFT between $-\pi \le \omega \le \pi$, we get $\Omega = \frac{\omega}{T}$ [9]. Thus, in summary, the DCT of a discrete time sequence is related to the Fourier transform of an even-extended version of the continuous-time function from which the given discrete-time samples were taken. In section II and III, we discuss different possible approximations to the actual current waveform, present their corresponding continuous transforms, analyze them and then develop the DCT current macro-model templates based on them. But before we discuss the model templates, we present some properties of the DCT of the current waveform that make it an attractive choice for macro-model construction.

A. Properties of Current Waveform DCT

Some interesting properties of the DCT of the current waveform are as follows:

Energy-per-cycle property: The DC sample, $I[0]$, of the current waveform DCT corresponds to *energy-per-cycle* [4,7] for the given input vector pair. In fact:

$$I[0] = \frac{\text{EPC}}{V_{dd} \Delta t \sqrt{N}} \quad (9)$$

where EPC is the energy-per-cycle for the given input vector pair, V_{dd} is the supply voltage, Δt is the sampling time period N is the length of the current sample. Thus, while estimating the DCT of the current waveform, we also estimate the energy-per-cycle of the current waveform as a function of input vector pair.

Energy compaction property: The DCT of the current waveform offers *energy compaction* so that one needs to estimate only a few dominant frequency components in order to capture the overall features of the time domain current waveform. It can also tolerate some error in estimation because of this property as any error in the estimation of these dominant samples, gets spread out in time domain when we do the inverse transform. Even though the error affects all the time domain samples, the impact of this error gets distributed over several samples. For example, given a 10,000 point DCT, one can get a reasonable estimate of the time domain current waveform by estimating the first few dominant terms as shown in Fig 1. In this plot we show the actual current waveform, and the approximate current waveform obtained by just keeping the 200 dominant terms of a 10,000 point DCT and taking its inverse. The remaining terms of the DCT are reduced to zero. Thus we remove the high-frequency terms from the DCT of the current waveform, which do not have a significant impact on current waveform shape. By removing these high-frequency terms we make the current waveform smooth which may lead to some inaccuracies in studying the impact of di/dt, but as we will show in section IV for early design cycle estimation, the results are fairly accurate.
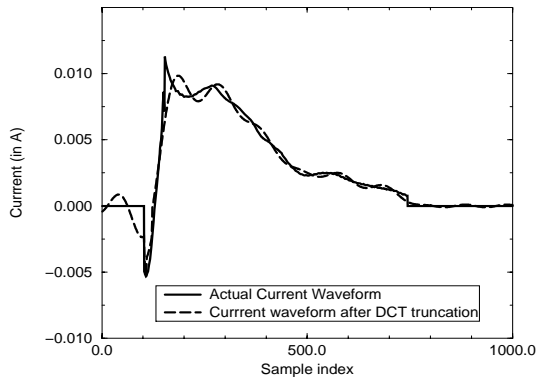
Figure 1 Actual vs. approximate current waveform with truncated DCT

II. ANALYSIS OF CURRENT WAVEFORMS

A CMOS combinational logic circuit draws current over one cycle in response to a vector pair at its inputs, and these waveform shapes typically fall into three categories:

Approximately triangular: In this case, the time domain current waveform has an approximately triangular geometry, and the DCT of the current waveform is similar to the DCT of an ideal triangular waveform.

Approximately trapezoidal: In this case, the time domain current waveform has an approximately trapezoidal geometry,

and the DCT of the current waveform is similar to the DCT of a regular trapezoidal waveform.

Multiple peaks In this case the time domain current waveform has multiple peaks which are separated in time. We use partitioning in the time-domain to convert such a waveform to a sequence of single-peak waveforms, which may be either triangular or trapezoidal, which are then modeled accordingly.

In the following, we will explore the FT of the triangular and the trapezoidal current waveforms, which will be used to infer reasonable forms of our DCT model templates.

A. Triangular Current Waveform

A typical triangular current waveform and its piece-wise linear triangular approximation are shown in Fig. 2. Before taking the DCT, we increase the waveform time duration and assume zero values for the waveform over the time extension. This is referred to as "zero padding," and its main advantage is that it increases the resolution of the DCT (the corresponding DTFT is sampled more closely), which gives a smoother DCT curve. With this, we use a simple generic template for the current macro-model as discussed in section III. Zero padding in the time domain is equivalent to using a higher sampling frequency in the frequency domain to sample DTFT and get the corresponding DFT/DCT [9]. Thus, even though the original current waveform ends at 2000 samples (which corresponds to 20 ns for our sampling period of 0.01 ns), we have shown 10,000 sample points in Fig 2.
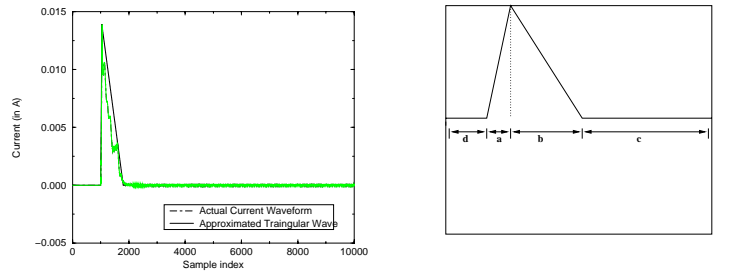
Figure 2. Typical triangular current waveform and its approx.

Figure 3. Diagram explaining $f_{\text{tri}}(t)$
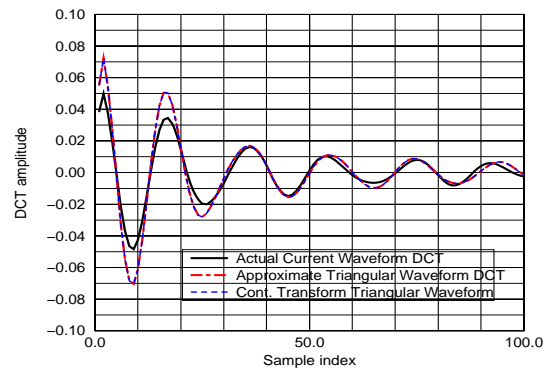
Figure 4 Typical DCT of a triangular current waveform

Since the DCT of a discrete time signal is related to the

Fourier transform of the even extension of the corresponding continuous time signal, we construct an even extension of the piece-wise linear triangular waveform and derive the corresponding continuous Fourier transform. The triangular waveform shown in Fig. 2 can be analytically expressed as:

$$f_{\text{tri}}(t) = \begin{cases} 0, & \text{for } 0 \leq t \leq d; \\ A(t-d)/a, & \text{for } d \leq t \leq a+d; \\ A(a+b+d-t)/b, & \text{for } a+d \leq t \leq a+b+d; \\ 0, & \text{for } a+b+d \leq t \leq Q. \end{cases}$$
(9)

where a, b, c, d are the dimensions shown in Fig. 3, and $Q = a+b+c+d$. The Fourier transform of the even extension of $f_{\text{tri}}(t)$, denoted by $F_{\text{tri}}(\Omega)$ is given by:

$$\begin{aligned} F_{\text{tri}}(\Omega) &= Ae^{-j\Omega(Q)}\left[\frac{2\cos\Omega(b+c) - 2\cos\Omega(a+b+c)}{a\Omega^2}\right] \\ &+ Ae^{-j\Omega(Q)}\left[\frac{2\cos\Omega(b+c) - 2\cos\Omega(c)}{b\Omega^2}\right] \end{aligned}$$
(10)

where Ω is the frequency in the analog (continuous) domain and is related to ω, the digital frequency by $\omega = \Omega T$. In our case, we have $T = .01$ ns and $N = 10,000$ before the even extension. After extension, we have $2N$ samples therefore $\omega = 2\pi k/2N = \pi k/10,000$ and $\Omega = \pi k/100$ rad/ns. It is important to note that for $\Omega = \pi k/100$, $F_{\text{tri}}(\Omega)$ is real for all integral values of k because $a+b+c+d = NT$.

In Fig. 4, we show the DCT of a typical triangular waveform, the DCT of its piecewise-linear approximation, and the DCT samples obtained from the continuous transform $F_{\text{tri}}(\Omega)$. The DCT plot of the current waveform has a very regular shape, making it a good candidate for model construction, which is one reason why we chose the DCT over other transforms. Compared to the DFT, the DCT has only real terms, which also makes the model simpler. Fig. 4 shows only the first 100 points of the 10,000 point DCT, for clarity, and is typical of all triangular current waveform shapes that we have seen. The DCT plot of the current waveform appears to be a decaying sinusoid. Therefore, instead of constructing a model for each point on the DCT (every frequency component), we use a generic function (a template) to model the entire DCT. In order to infer the form of this template, we compare the plot of continuous transform with the DCT plot. Except for a scale factor, it is obvious that $F_{\text{tri}}(\Omega)$ has the right shape and thus may help us define the form of the current macro-model.

The most important aspect to observe is that the amplitude of $F_{\text{tri}}(\Omega)$ decays as the square of the frequency, a fact which we will make use of in developing our DCT model template. Since the DCT looks like a decaying sinusoid, we can use a simple sinusoid which decays as a square of the frequency. But unlike a decaying sinusoid which has a constant time period, the plots in Fig. 4 show that in case of a current waveform and its approximations (both discrete and continuous transform plots) the time period is varying (we can use the difference between consecutive maxima/minima to measure the time period). We use this fact in our model as well. It actually helps simplify our current macro-model, in the sense that we do not use multiple cosine terms as in $F_{\text{tri}}(\Omega)$, instead we use a simplified expression with a variable time period, as discussed in section III.A.

B. Trapezoidal Current Waveform

In some cases, the current waveform has *trapezoidal* shape, as shown in Fig. 5, which also shows a piecewise linear trapezoidal approximation to the current waveform. The equation of the piece-wise linear trapezoidal wave is given by:

$$f_{\text{tra}}(t) = \begin{cases} 0, & \text{for } 0 \leq t \leq a \\ A(t-a)/b, & \text{for } a \leq t \leq a+b \\ A, & \text{for } a+b \leq t \leq a+b+c \\ A(U-t)/d, & \text{for } a+b+c \leq t \leq U \\ 0 & \text{for } U \leq t \leq W \end{cases}$$
(11)

where the dimensions are illustrated in Fig. 6, $U = a+b+c+d$ and $W = a+b+c+d+e$. The Fourier transform of the even extension of $f_{\text{tra}}(t)$, denoted by $F_{\text{tra}}(\Omega)$ is given by:

$$\begin{aligned} F_{\text{tra}}(\Omega) &= Ae^{-j\Omega(W)}\left[\frac{2\cos\Omega(V) - 2\cos\Omega(W-a)}{b\Omega^2}\right] \\ &+ Ae^{-j\Omega(W)}\left[\frac{2\cos\Omega(e+d) - 2\cos\Omega(e)}{d\Omega^2}\right] \end{aligned}$$
(12)

where $V = c+d+e$. As before, it turns out that F_{tra} is real for the values of Ω under consideration. In Fig 7, we show the DCT of the current waveform, the DCT of the trapezoidal waveform approximation, and the plot of the continuous transform. Again, we can observe that, except for a scale factor, the continuous transform plot compares well with the actual current waveform DCT. Thus, we can use the continuous transform to infer the shape of the DCT model template. The most obvious inference we can draw from the continuous transform, is, as before, the quadratic decay in the amplitude of the DCT, in terms of frequency.

If we compare the plots in Fig. 4 and Fig. 7, we can see that the DCT plot of a trapezoid shows some deviation from the perfectly decaying sinusoid which we got for triangular waves. The same observation holds true for the respective continuous transform as well. This makes the estimation of a trapezoidal current waveform difficult, because we cannot use a decaying sinusoid with a varying time period to simplify our model, as we do in the case of triangular waveforms (see section III). Therefore, we have proposed a slightly different analytical expression for the trapezoidal current waveform in our macro-model. The macro-model equation is based on $F_{\text{tra}}(\Omega)$ to some extent, and also based on the observation of the plots. Figure 7, shows that we cannot use a single cosine function to capture the current waveform DCT, therefore, for trapezoidal waveforms, we use two cosine terms (the continuous transform has 4 cosine terms), and again use a varying time period. In case of trapezoidal waveforms too, we use the difference between consecutive maxima (or minima) as the time period. The DCT plots show that this difference varies for successive maxima (or minima). We use this information in our macro-model, as discussed in section III.B.
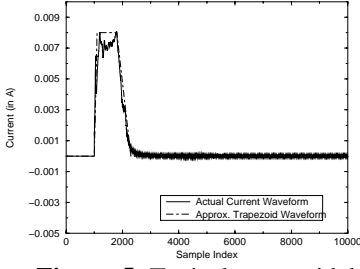
**Figure 5.** Typical trapezoidal current waveform and its approx.
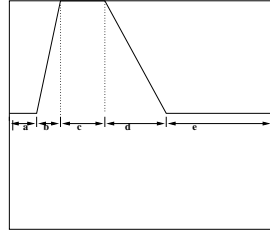
**Figure 6.** Diagram explaining $f_{\text{tra}}(t)$
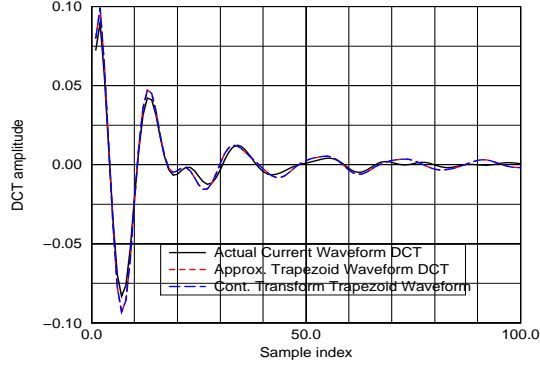
Figure 7 Typical DCT of a trapezoidal current waveform
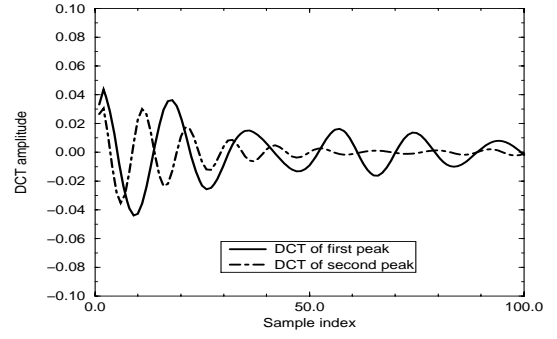
C. Waveform Partitioning

When we have current waveforms with multiple peaks which are far apart, the DCT of the current waveform gets distorted. Multiple peak current waveforms can actually be considered as a summation of time shifted single peak current waveforms. The DCT of such a summation of current waveforms is equal to sum of the DCT of each of those time shifted current waveforms. This is true because DCT is a linear transformation. In Fig. 8, we show a typical multiple peak current waveform with two distinct peaks which are far apart. The DCT of the multiple peak current waveform obtained after zero padding is shown in Fig 9. Since the DCT of current waveform shown in Fig. 9 is distorted it is difficult to model it as the DCT of a triangular or trapezoidal waveform.
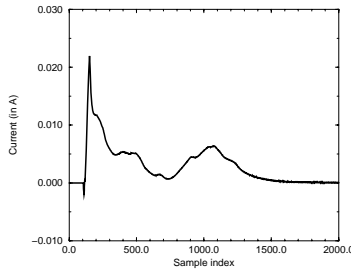
Figure 8. Typical multiple peak current waveform.
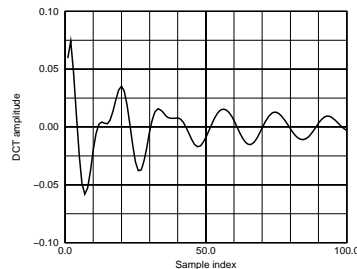
Figure 9. DCT of a mutiple peak current waveform

Figure 10 DCT of the current waveform after partitioning waveform

Therefore, for multiple peak current waveforms we use a simple heuristic technique to partition the time domain current waveform. Partitioning converts the multiple peak time domain current waveform into a sequence of single-peak waveforms. The flow chart for the partitioning technique is shown in Fig. 11 and the steps are described below.

Step 1: Given a sample current waveform of length N (here N corresponds to the current sample length before zero padding), detect the global peak current (current sample with the maximum magnitude) i_{pg} and the corresponding index i_{tg} at which this peak occurs. This can be done by traversing through the current waveform. Set $i_p = i_{pg}$, $i_t = i_{tg}$ and $index = 0$.

Step 2: From the peak index, i_t traverse to a point where the current value is less than 10% (a user defined threshold) of the global peak, i.e. $i[n] \leq 0.01 i_{pg}$ and note the index n. Set $p_{index} = n$. This is a partition.

Step 3: Consider the remaining current samples from $i[p_{index} + 1]$ to $i[N-1]$ (where N is the length of the current sample) and detect the new peak i_p and the corresponding index i_t among these samples.

Step 4: Compare the new peak i_p with a user defined threshold to see if it is significant. In our case we compare it with i_{pg} found in step 1. If the current $i_p \geq .15 i_{pg}$, go to step 5, else *stop*, partitioning is not required.

Step 5: Set $index = index + 1$. Go to Step 2.

With the above partitioning technique, one can partition multiple peak current waveforms, and even detect if there are any multiple peaks in a set of current waveforms. The current waveform shown in Fig. 8 was partitioned using the above algorithm and the DCT of the resulting partitions is shown in Fig. 10. The DCT of the partitions do not show any significant distortion and appear to be decaying sinusoids. In this case, therefore one can can use a triangular template function (discussed in section III) for each of the partitions.
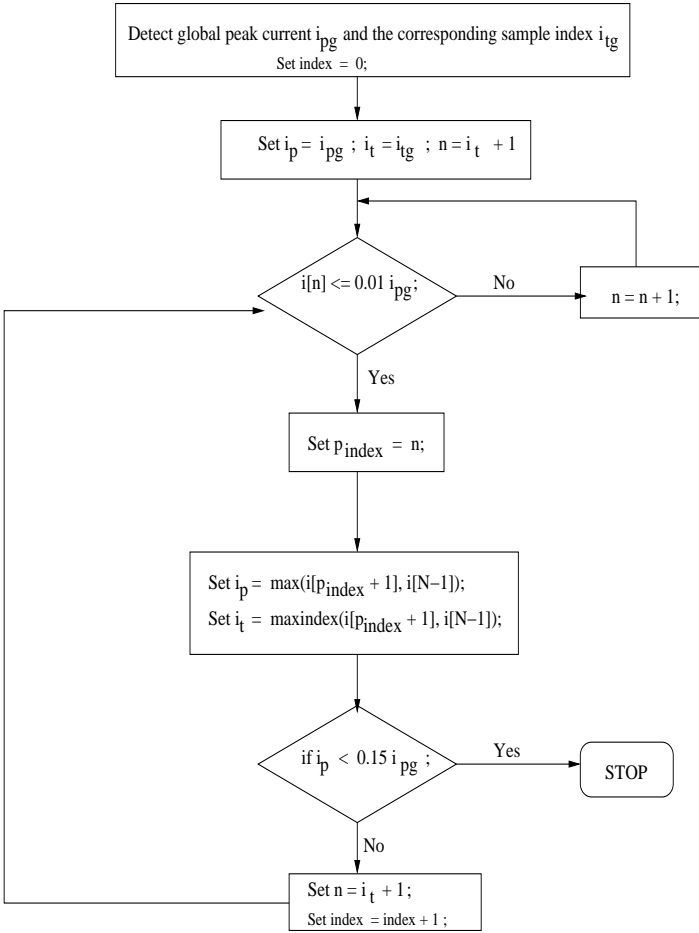
Figure 11. Partitioning algorithm

III. MACRO-MODEL CONSTRUCTION

In order to formulate a macro-model, one needs to develop a mapping between the input Boolean vector pair and certain variables that can be the input variables to the macro-model. These variables can then be used as *predictor variables* in regression to estimate the parameters of the DCT template function. Basically, the current macro-model is a template function whose parameters are modeled as functions of these predictor variables. The coefficients of these functions are obtained through regression analysis. Now, for a given input vector, a single node at the primary input can undergo one of the following four transitions $\{0 \rightarrow 1, 1 \rightarrow 0, 1 \rightarrow 1, 0 \rightarrow 0\}$. We treat the set of four possible transitions at each primary input as a categorical variable [11]. In general, a categorical variable with k levels (in our case 4) is transformed into $k-1$ variables (in our case 3) each with 2 levels. This process of creating variables from categorical variables is called *dummy coding* [11]. In our case, we use three variables x_1, x_2 and x_3 to code the categorical variables. The dummy coding for the 4 levels of our categorical variable is shown in the second, third and fourth columns of Table I (and also shown in [7]). These three columns are labeled as x_1, x_2, x_3 in Table I. The mapping shown in Table I, does not impose any implicit order among the variables and it prevents any bias in the effects these variables may have on the current waveform. Basically, each input transition is mapped to a vector of $[x_1 \; x_2 \; x_3]$, but

this has a clear disadvantage in that it increases the number of variables as well as the number of regression coefficients. But we have found that, depending on the Hamming distance of the vector pair, we can introduce a less expensive solution, as follows. For large Hamming distances (more than 60% of the primary inputs are switching), very few inputs undergo $0 \rightarrow 0$ and $1 \rightarrow 1$ transition, therefore we can assume that the categorical variable has just two levels, so that a single variable is required to represent a primary input. The mapping for large Hamming distance is shown in the fifth column of Table I, and it is labeled as y. Thus, we can map each transition at the primary input to the corresponding dummy variable and thus generate a set of input variables for a given input vector pair.

The above case analysis based on the fraction of primary inputs switching, was found to be useful for other reasons also. It was observed that in most cases if the input vector pair has a large Hamming distance (more than 60% of primary inputs switching), a triangular template was suitable but for small Hamming distances (less than or equal to 60% of primary inputs switching) one had to choose from either the triangular or trapezoidal template depending on the circuit. Therefore, it was really difficult to construct a single analytical expression to capture the DCT for all input vector pairs (in which the Hamming distance is implicit, not an explicit variable) when one can see both the triangular and trapezoidal current waveforms for the same logic block. Moreover the variance in the model parameters across Hamming distance was so large (especially between small and large Hamming distances) that it was difficult to capture them with simple polynomial functions with reasonable accuracy in a single analytical expression. Since complex higher order polynomials require more co-efficients, and are usually not very accurate when it comes to modeling through regression, we preferred classifying the input vector pair based on the fraction of primary inputs switching. Classifying the input vectors pairs based on the fraction of primary inputs switching is very similar to the earlier classification based on individual Hamming distance proposed in [12, 16]. But the biggest advantage of classification based on fraction of inputs switching is that, the number of analytical expressions required to build a complete model does not grow linearly with the number of primary inputs, p, for a given logic block. When partitioning is done based on Hamming distance of the input vector pairs, the actual current macro-model is a set of analytical expressions, corresponding to each Hamming distance. But if partitioning is done based on fraction of primary inputs switching, the number of analytical expressions required to specify the current model has an upper bound (in our case, we can have at most 5 analytical expressions). This in turn reduces the number of coefficients needed to build the model and it also reduces the number of simulations required to construct the model using regression. Therefore in our macro-model construction flow we partition the input vector pairs into following groups, and construct one single analytical expression for each group:

1. Up to 10% of primary input(s), p, switching. If $0.1p$ is not a whole number, consider the corresponding $\lceil 0.1p \rceil$ and the same holds for rest of the partitions.

TABLE I

DUMMY CODING FOR TRANSITIONS AT A SINGLE NODE.

y	x_1	x_2	x_3	y
$0 \to 1$	0	0	1	1
$1 \to 0$	0	1	0	2
$1 \to 1$	1	0	0	3
$0 \to 0$	0	0	0	4

2. For 10% to 20% of primary inputs switching.

3. For 20% to 30% of primary inputs switching.

4. For 30% to 60% of primary inputs switching.

5. For 60% to 100% of primary inputs switching, we construct a single model. This group falls in the category of large Hamming distance. We will show later it is much easier to construct accurate models for large Hamming distances.

In this classification, if we come across cases where we have already built the model for a given Hamming distance (in the previous partition), we do not build a new model for that Hamming distance in the next partition. This is because, in some cases, such as say a design with 4 inputs, the 10% switching criterion (=0.4) with ceiling function would correspond to 1, and the 20% switching (=0.8) with ceiling function would again correspond to 1. Even though we partition the input vector pairs based on the fraction of primary inputs switching, the actual analytical expression for each partition is a function of the input vector pair itself. In the next two sections, we present the template functions used to model the DCT of the current waveforms and present techniques to estimate the parameters of these template functions.

A. Triangular Template

The triangular template is a simplified as well as a modified version of $F_{\text{tri}}(\Omega)$. The simplification is achieved because we explicitly incorporate the effect of the changing time period. We estimate the first few time periods (in our case, five) directly and use them in a single cosine term, whose other parameters do not change. We can do this only because of the energy compaction achieved with transforms like DCT. The template equation for the DCT of the triangular current waveform is given by:

$$I_{\text{tri}}(k) = D(k) A \cos\left(\frac{2\pi(k-1)}{T_i} \right), \quad k = 1, 2, \ldots \quad (13)$$

where k is the sample index, $D(k)$ is a decay factor, A is a scale factor, which we refer to as amplitude, and T_i, $i \in \{1, 2, 3, 4, 5\}$ is the variable time period. The DCT value corresponding to $I_{\text{tri}}(k)|_{k=0}$ is called the DC Value. These terms are the *parameters* of the template, which we relate to the variables associated with the input vector pair. The triangular template is very similar to what was presented in [12], except for polynomial functions used to model the parameters of the template function, which have been modified to account for possible interaction between the input variables, through cross product terms. In order to construct the current macro-model we model the parameters of template function as a function of input vector pair as described in the next few sections.
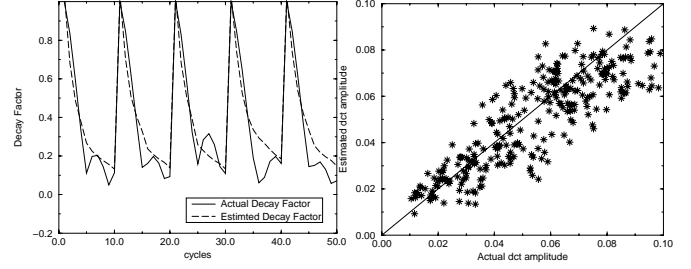
Figure 12. Decay factor est. for sample waveforms.

Figure 13. Act. vs. est. DCT in amplitude for alu2

1) Decay Factor : In order to model the decay factor as a function of the input vector pair, we use the following functional form, which is motivated by the inverse square dependence on frequency seen in F_{tri} :

$$D(k) = \frac{f(x)k}{g(x)k^2 + h(x)} \quad (14)$$

where x is either y (a vector of length one, refer to Table I) or the vector $[x_1 \ x_2 \ x_3]$ (refer to Table I), depending on whether the model is for a small or large Hamming distance input vector pair, and $f(x)$, $g(x)$, $h(x)$ are polynomial functions of the variables associated with the input vector pair. All the three polynomial functions $f(x)$, $g(x)$, $h(x)$ have the same form and it is given by:

$$\Pi_{i=1}^{p}(\alpha_i y_i + \beta_i), \text{ for large Hamm. dist.} \quad (15)$$

$$\Pi_{i=1}^{p}(\alpha_{i1} x_{1,i} + \alpha_{i2} x_{2,i} + \alpha_{i3} x_{3,i} + \beta_i),$$

$$\text{for small Hamm. dist.} \quad (16)$$

where p is the number of primary inputs. With the above polynomial functions, one can account for possible interaction between the input variables, through the cross-product terms, without significantly increasing the number of co-efficients compared to a linear model. The coefficients of these polynomials are obtained by substituting these polynomial functions into (14) and using non-linear regression. The sample current waveforms for regression are obtained from HSPICE simulations using randomly generated vector pairs for a given Hamming distance, in a process of characterization (see section III.C). For regression, we consider the points on the DCT corresponding to the maxima and minima of the decaying sinusoid, normalize them by the amplitude A (so that $D(1) = 1$; the normalization is required because the model for $I(k)$ given above includes an explicit term for the amplitude). Furthermore, in order to reduce the computational cost of building the model, we only consider the maxima and minima in the first five cycles during regression. Some typical decay factor curves and their corresponding estimates obtained using our model are shown in Fig 12. These were obtained from the alu2 MCNC benchmark circuit [13].

2) Amplitude and DC component: One of the primary reasons for partitioning the current macro-model based on the fraction of primary inputs switching, is the large variance in the values of A, the amplitude and $I(0)$, the DC value, across the input vector space. With partitioning, we can model both A

and $I(0)$ along the lines of the polynomial functions discussed in section III.A.1, given by:

$$\Pi_{i=1}^{p}(\alpha_i y_i + \beta_i), \text{ for large Hamm. dist.} \quad (17)$$

$$\Pi_{i=1}^{p}(\alpha_{i1} x_{1,i} + \alpha_{i2} x_{2,i} + \alpha_{i3} x_{3,i} + \beta_i),$$
$$\text{for small Hamm. dist.} \quad (18)$$

where p is the number of primary inputs and $x_{1,i}, x_{2,i}, x_{3,i}$ and y_i are the variables associated with the input vector pair and obtained using the mapping shown in Table I. The coefficients of the above polynomial functions are obtained using linear regression on the same set of waveforms used for $D(k)$. The first sample of the DCT of the current waveform corresponds to the DC Value $I[0]$ and the next sample corresponds to the Amplitude i.e, $A = I[1]$. These data points are obtained from the sample current waveforms and used in linear-regression to obtain the coefficients of the polynomial functions. It must be emphasized that A and $I[0]$ are modeled with different polynomial functions, as far as the coefficients go but they have the same functional form as represented in (17) and (18). Estimating the DC value separately simplifies the DCT template function for the current waveform I_{tri}. Otherwise, for $k = 0$ the template function would have required a phase angle, ϕ, as a function of the input vector pair. This would have required a non-linear regression to estimate ϕ which is computationally more expensive than simple linear regression. A typical plot of actual DCT amplitude vs estimated DCT amplitude is shown in Fig 13 and a similar plot for the DC value is shown in Fig 14. As discussed in section I.A, $I[0]$ corresponds to the *energy-per-cycle* for a given input vector pair and it is a by-product of our current macro-modeling methodology.
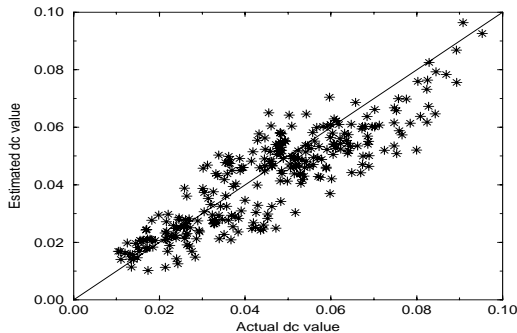
Figure 14 Actual vs. estimated DCT DC value for alu2
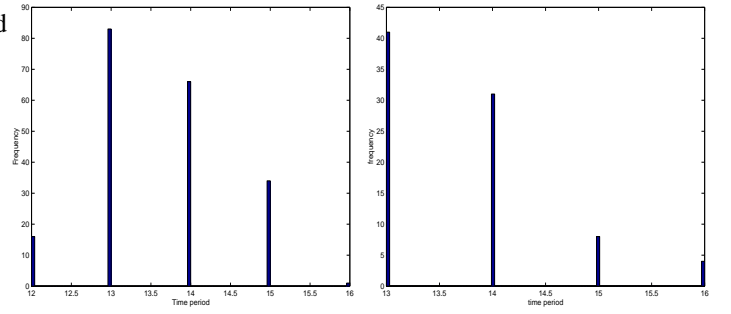
Figure 15. Distribution of the first time period for large Hamming distance, alu2

Figure 16. Dist. of the first time period, 40% to 60% inputs switching, alu2

3) Time Period : The variable time period for the DCT template is measured as the difference between successive maxima/minima of DCT of the actual current waveform. In our macro-modeling technique we estimate the first five time periods of the DCT template and this is denoted by T_i, $i \in \{1, 2, 3, 4, 5\}$. The choice of just five time periods is empirical and gives good accuracy because of the *energy compaction* property of the DCT discussed in section I.A. But, if one attempts to build a single comprehensive model for each of these time periods across all the possible input vector pairs the time period turns out to be the most difficult parameter to model. To make things worse, this parameter has the largest impact on the current model, because any error in time period causes significant change in the frequency spectrum. However, if we partition the model based on the fraction of primary inputs switching as proposed earlier, we find that within a partition there is not a significant variation in the time period.

Therefore for every partition, we compute during characterization a *nominal* value for each of the first five time periods. This is selected as the peak value of the distribution of observed time periods for that partition. In statistical terms, we choose the *mode* of the time period distribution as the time period for a given partition of the input vector pairs. The choice of the first five time periods is not a limitation of the model and can be increased, if there is a need. A typical histogram of the first time period for alu2 is shown in Fig. 15, and it can be seen that there is no significant variation in the time period. The triangular template provides a good model in cases of both large and small Hamming distances, but in some cases for small Hamming distance, the trapezoidal template is better. However, estimating the coefficients of the parameters of the triangular template is simpler and which would become clear after we discuss the trapezoidal template.

B. Trapezoidal Template

A trapezoidal template equation model seems to best fit the current waveforms for low Hamming distances, when the fraction of primary inputs switching is $\leq 60\%$. The trapezoidal template presented here is similar to that presented in [14] and is given by:

$$I_{\text{tra}}(k) \quad = \quad AD_1(k) \left[\cos \left(\frac{2\pi(k-1)}{T_i} \right) \right]$$

$$+ \quad AD_2(k)\left[\cos\left(2\pi(k-1)\omega\right)\right], k = 1, 2, \ldots \quad (19)$$

where k is again the sample index, T_i, $i \in \{1,2,3,4,5\}$, is the variable time period, A is the amplitude, ω is the frequency of the second cosine term and $D_1(k)$, $D_2(k)$ are the decay factors corresponding to the cosine terms. The DCT of a trapezoidal current waveform deviates from the simple decaying sinusoid which we observe for the triangular current waveform, as shown in Fig. 7. Therefore, we introduce a second cosine term in the trapezoidal template, as a single cosine term cannot capture this deviation. Since a trapezoidal waveform can be considered as a linear combination of two triangular waveforms, the proposed template function is equivalent to a linear combination of the DCT of two triangular waveforms. This is possible because DCT is a linear transform. Therefore the DCT of a linear combination of triangular waveforms in time domain is equal to sum of the individual DCT. The parameters of the trapezoidal template are motivated by the continuous Fourier transform of the piecewise-linear trapezoidal waveform, F_{tra}, discussed in section II.B, some empirical observations and the fact that a trapezoidal current waveform is a linear combination of two triangular waveforms. These observations help in simplifying the template function to just two cosine terms. The simplification is achieved by using the same concept of variable time period T_i as discussed in section III.A.3.

The six parameter models needed to completely specify the trapezoidal template as a function of the input vector pair are given below:

Amplitude: We use $I[1]$ as the amplitude in our template function, because the DCT appears to be a decaying sinusoid. This is estimated separately using linear regression and is motivated by the term A which appears in F_{tra} (refer to (12)). It is given by:

$$A = w(x) \quad (20)$$

where $w(x)$ is a polynomial function of the input vector pair.

Decay Factor ($D_1(k)$, $D_2(k)$)**:** These two terms are motivated by the inverse square dependence on frequency seen in F_{tra} (refer to (12)) and are given by:

$$D_1(k) = \frac{p(x)k}{r(x)k^2 + s(x)}, \quad D_2(k) = \frac{q(x)k}{r(x)k^2 + s(x)} \quad (21)$$

where $p(x)$, $q(x)$, $r(x)$ and $s(x)$ are polynomial functions of the variables associated with the input vector pair.

Time Period, Frequency (T_i, ω)**:** Since the DCT template uses two cosine terms we need two frequency terms. The philosophy behind using two cosine terms is that one cosine term would produce a basic decaying sinusoid (as in the triangular case) and the other would account for the deviations from the decaying sinusoid that are seen at low Hamming distance. Therefore, the variable time period T_i is used with one of the two cosine terms, the one which is aimed at generating the decaying sinusoid. The frequency ω of the other cosine term is modeled as a polynomial function of the variables associated with the input vector pair. The variable time period T_i also helps in simplifying the model to some extent by reducing the model to just two cosine terms instead of at least four as suggested by the form of F_{tra}. The variable time period terms

T_i are approximated as the difference between consecutive maxima (or minima) of the DCT of the current waveform. As in the triangular case the time period does not vary much for a given partition, as shown in Fig. 16. Therefore, we use the same method as described in section III.A.3. to get an estimate of the first five time periods. Basically, from the distribution of time periods for a given Hamming distance, we choose the time period value which occurs most often (in statistical terms, we choose the *mode* of the time period distribution as the time period for a given Hamming distance). The first five time periods are enough to get an estimate of the dominant terms of the DCT, because of energy compaction. The frequency of the other cosine term is given by:

$$\omega = t(x) \quad (22)$$

where $t(x)$ is a polynomial function of the variables associated with the input vector pair.

DC Value ($I(0)$) **:** The first sample of the DCT of the current waveform corresponds to the DC value. This is estimated separately for the same reasons discussed in section III.A.2 and is given by:

$$I[0] = u(x) \quad (23)$$

where $u(x)$ is a polynomial function of the input vector pair. The coefficients of $u(x)$ are estimated using linear regression on a set of data points obtained from sample current waveforms.

Since the trapezoidal model is usually used with low Hamming distances, x corresponds to the vector $[x_1 \ x_2 \ x_3]$ for all the polynomial functions used in the trapezoidal template. The polynomial functions $p(x)$, $q(x)$, $r(x)$, $s(x)$, $t(x)$, $u(x)$ and $w(x)$ used to model the various parameters of the template have the same form, given by:

$$\Pi_{i=1}^{p}(\alpha_{i1}x1 + \beta_{i2}x2 + \gamma_{i3}x3 + \delta_i) \quad (24)$$

where p is the number of primary inputs. These parameters are then substituted in the DCT template (19), (except for A and $I[0]$) and the coefficients of the polynomials are estimated simultaneously using non-linear regression to construct the current macro-model. As mentioned earlier, A and $I[0]$ are estimated separately using linear regression. The polynomial function obtained for A after linear-regression is substituted in the template function before the estimation of other parameters using non-linear regression. This technique simplifies the non-linear regression, and reduces the number of parameters to be estimated using non-linear regression. Moreover, since the value of A can be obtained easily, it is simpler to estimate it directly using linear-regression.

Thus, we can estimate the parameters of the triangular and trapezoidal template, given a set of sample current waveforms. But, in order to construct the current macro-model automatically (without any user intervention), we need a technique to:

1. Determine the sample size on the fly during characterization. Since we use regression to construct the model, we need an appropriate sample size to build the model. We cannot use all the possible input vector pairs to build the model because of

the exponential nature of possible input vector pairs. A logic block with p primary inputs has 2^{2p} possible input vector pairs.

2. Determine the appropriate template function to be used for a given partition of input vectors, while constructing the model. We have to chose either a triangular, or a trapezoidal template, or possibly decide about partitioning the time domain current waveform.
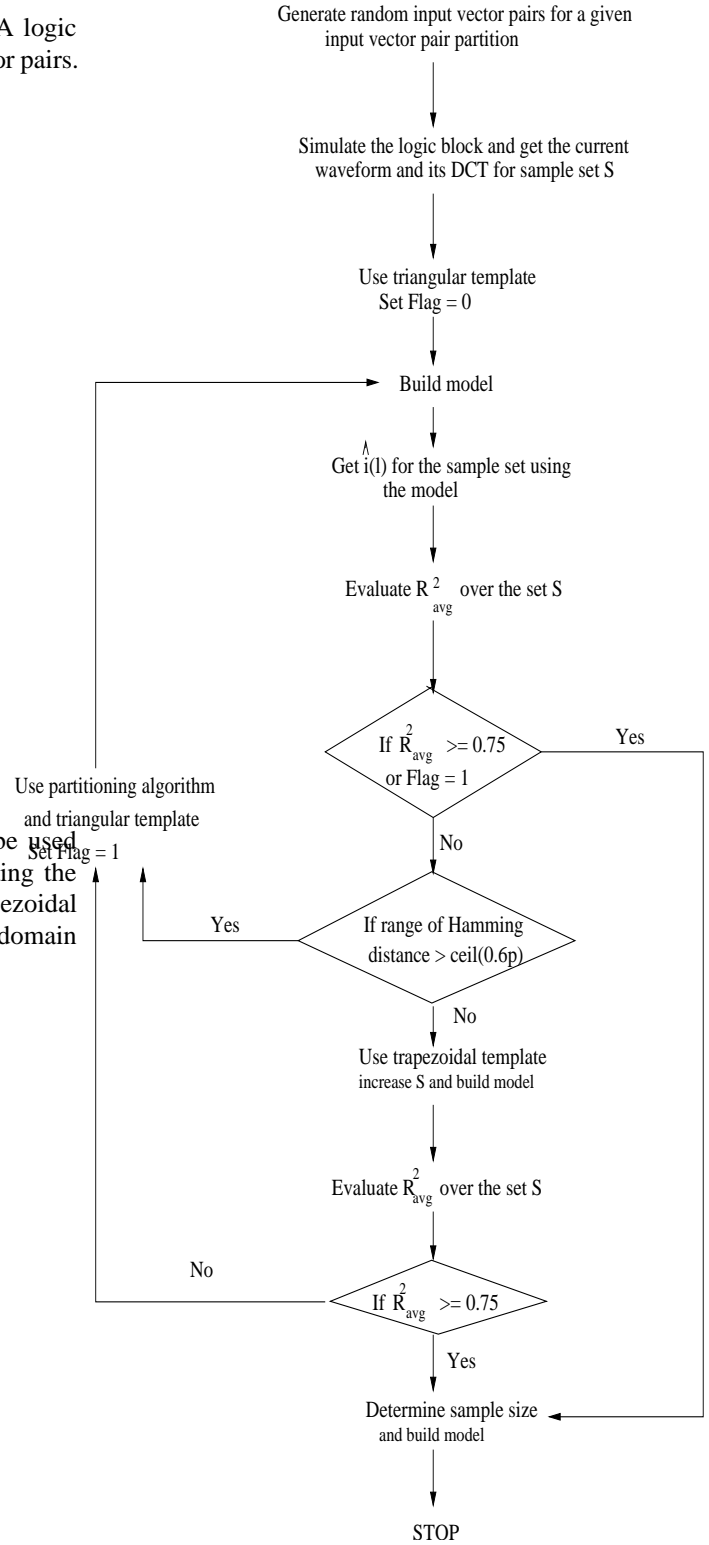
Generate random input vector pairs for a given input vector pair partition

Simulate the logic block and get the current waveform and its DCT for sample set S

Use triangular template
Set Flag = 0

Build model

Get î(l) for the sample set using the model

Evaluate R^2_{avg} over the set S

If R^2_{avg} >= 0.75 or Flag = 1   Yes

No

If range of Hamming distance > ceil(0.6p)   Yes

Use partitioning algorithm and triangular template
Set Flag = 1

No

Use trapezoidal template
increase S and build model

Evaluate R^2_{avg} over the set S

If R^2_{avg} >= 0.75   No

Yes

Determine sample size and build model

STOP

Figure 17. Characterization flow

C. Characterization Flow

In this section, we propose a characterization flow that can be used to generate current macro-models for a combinational logic block, given its low-level description and a circuit simulator. In our case, the low-level description corresponds to a SPICE description of the circuit and HSPICE as the

In order to accomplish the above two tasks, we need some quantitative metrics and an automatic current model characterization flow based on such metrics. In the next, section we present such a macro-model characterization flow.

circuit simulator. In order to automate the model development flow, we need to choose on the fly the correct template and also the number of simulations needed to build the model. This leads us to the need for a set of quantitative metrics which can be used to decide the appropriate template and the number of simulations. In this regard, we propose to use a metric borrowed from statistics, called the *coefficient of determination* [14], denoted by R^2 and given by:

$$R^2 = 1 - \frac{\sum_{l=0}^{N-1} \|i(l) - \widehat{i}(l)\|^2}{\sum_{l=0}^{N-1} \|i(l) - \bar{i}\|^2} \tag{25}$$

where $i(l)$ is the actual current waveform sample, $\bar{i}$ is the mean of the samples of the actual current waveform, $\widehat{i}(l)$ is the estimated current waveform sample, and here N is the length of the current waveform sequence without zero padding. The coefficient of determination R^2 measures the proportion of variability of the dependent variable explained by regression on predictor variables. The closer R^2 is to 1, the more accurate the model is. Since it is very difficult to achieve a value closer to 1, we use a threshold of $R^2 > 0.75$ to test the accuracy of the model. Actually, we found that if $R^2 > 0.75$, the current estimates were fairly accurate. We use this metric to choose between the two templates or decide upon partitioning. The steps of the automatic current characterization flow based on this metric are given below. The objective of this flow is to minimize the chances of building multiple models using different templates and is based on the observations made while analyzing several current waveforms. In this flow, we will assume that we have a sample size determination algorithm, which will be discussed later in the next section.

Step 1: Simulate the circuit using the circuit simulator for a set of randomly generated input vector pairs for a given input vector pair partition. Get the time domain current waveforms and the corresponding DCT. The set of input vector pairs used is called the initial characterization set and denoted by S. The size of the initial characterization set is a function of the number of primary inputs, p of the logic block. This comes from the fact that the number of data points for regression should be at least equal to the number of coefficients in the model. Since in our modeling flow we estimate the A and $I[0]$ separately, the initial number of simulations, is chosen based on the number of coefficients in the polynomial function for A or $I[0]$.

Step 2: By default, use the triangular template to build the model, since it occurs most frequently. The model parameters can be estimated using the methodology described in section III.A. Since the model estimates the DCT of the current waveforms in the sample set S, get the corresponding time-domain current waveform samples, $\widehat{i}(l)$ for the sample set S using the model.

Step 3: Evaluate R^2 for each current waveform and get the average R^2, denoted by R^2_{avg}. We call it the the average coefficient of determination and it is given by:

$$R^2_{\text{avg}} = \frac{1}{|S|} \sum_{m=1}^{|S|} R^2(m) \tag{26}$$

where $|S|$ is the size of the initial characterization set.

Step 4: If $R^2_{\text{avg}} > 0.75$, then the template is correct and use the *sample size determination algorithm* to get the correct sample size and stop, else go to step 5.

Step 5: Check the range of Hamming distance of input vectors under consideration, if the model is being built for large Hamming distance (more than 60% of primary inputs switching), use the partitioning algorithm to partition the time domain current waveform and use a triangular template for each partition. If the model is being built for small Hamming distance use the trapezoidal template and increase the size of S, because the trapezoidal template has more coefficients.

Step 6: Build the model again depending on the decision made in step 5. Get the corresponding R^2_{avg}. If for large Hamming distance R^2_{avg} does not improve, assume that the current model is the best one can achieve with this technique. One can try the trapezoidal template for individual partitions and build the model again. But our experimental results show that such a case should not arise. If the model is for small Hamming distance go to step 7, else go to step 8.

Step 7: If for small Hamming distance R^2_{avg} does not improve even with trapezoidal template use partitioning. After partitioning the time domain current waveform use the triangular template for each partition and build the model and evaluate R^2_{avg}. It should improve, if it doesn't then one can assume that it is the best one can achieve with this technique. One can again try using the trapezoidal template for each partition, but we have not come across such a scenario.

Step 8: Basically by step 6 or 7 one should have the correct model template and therefore one can use the *sample size determination algorithm* to get the correct sample size and build the model.

The above steps are based on our current macro-modeling experience and the objective is to get the correct template the first time in majority of cases. Since the triangular template occurs most of the time, we use it first. The threshold value for R^2_{avg} can be set by the user but the value 0.75 suggested above is also again based on observations made during the characterization of current models for various logic blocks, some of which are listed in Table II. A flow-chart depicting the above characterization flow is shown in Fig. 17. In the next section, we present a simple algorithm to determine an adequate sample size and complete the building of the model.

1) Sample Size: Once the correct template is known, we need to know the size of the sample set for characterization and go on to build the model. The size of initial characterization set is a function of p, the number of primary inputs. But we need a large enough sample set so that it adequately represents all the vector pairs for a given partition and gives a good value of the parameters. If the characterization sample set adequately represents the entire set (the entire set of input vector pairs for a given partition will be just referred to as the population, a term borrowed from statistics) or at least a majority of vector pairs for a given partition, then the model can be used for estimating the current waveform of other elements of the population. It is to be noted that including the entire population in the characterization set is infeasible because of the large population size as discussed earlier. In this

regard we developed a simple algorithm based on a number of quantitative metrics of goodness of fit. The algorithm is based on the simple philosophy that as the size of the characterization set increases, the ability of the macro-model to estimate the current waveforms for random input vector pairs which are not a part of the characterization set improves. The intuitive reason for this is the fact that as the size of the characterization set increases the model gets *trained* for more elements of the population and thus its ability to predict the behavior of the population improves. The steps of the algorithm to determine the size of the sample set are outlined below:

Step 1: Get the correct template and the model built using the initial characterization set S from the characterization flow described in section III.C.

Step 2: Generate another random sample of input vector pairs called T for the given input vector pair partition. Simulate the logic block using HSPICE (or any circuit simulator) to get the current waveforms for the set of vector pairs in T.

Step 3: Test the model on the set of random input vector pairs, denoted by T, generated in step 2. Comparison is done using three metrics: **a)** Average relative error in peak current estimation, e_{pavg}. **b)** Average relative error in the time instant at which the peak occurs, e_{tavg}. **c)** Coefficient of determination, R^2_{avg}.

These error metrics are easy to compute and give a good estimate of the accuracy of the model. The relative errors have been described in section IV. The size of the test set T is user defined in our case we use $|T| = |S|$, where $|S|$ is the size of the initial characterization set, obtained at the beginning of the characterization flow. The sample set T is called the test set and remains fixed during the algorithm.

Step 4: Compare the metrics with user defined thresholds. In our case, we use less that 20% relative error for peak current and the time instant at which peak occurs. A reasonable threshold for the coefficient of determination is that it should be greater than 0.75. If the metrics satisfy the threshold criterion, then stop. If any or all of the metrics are above threshold and not improving since last iteration, then stop, else go to step 5.

Step 5: Increase the size of the characterization set S, by a user defined amount, and build the model again. In our case we found that we need to increment $|S|$ by at least 10. So we used the following technique to get the increment value.

a) Calculate the number of Hamming distances covered in each input vector pair partition, denoted by H_{num}. **b)** if $H_{num} \geq 10$, $|S| = |S| + H_{num}$, else $|S| = |S| + \lceil (10/H_{num}) \rceil H_{num}$.

In order to build the model, use the previous solution as the initial guess for the non-linear regression part of the characterization flow. There can be multiple flavors for increasing the size of the characterization, but in our case we do some more HSPICE simulations and increase the size of S, and not tamper with the test set T (basically the model is tested on the same set each time). After building the model go back to step 3.
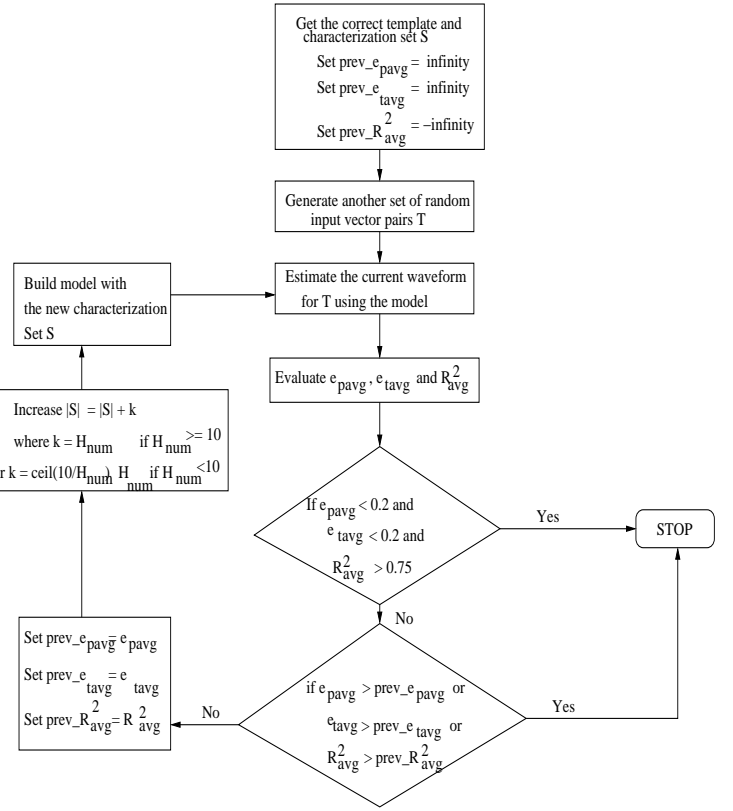

Figure 18. Sample size determination

At the end of this algorithm we have the characterization set and the model. Though the model is tested on different sample of random input vector pairs, it may not represent the entire population. But we believe that randomization of the sample set T alleviates some of the problem. Thus, it is possible to build current macro-models for a logic block without any user intervention. In Fig. 18 we present a flow chart for the sample size determination algorithm. The characterization flow was used in conjunction with the sample size determination algorithm to generate current macro-models for various benchmark circuits. In the next section we present some experimental results that illustrate the validity of our approach.
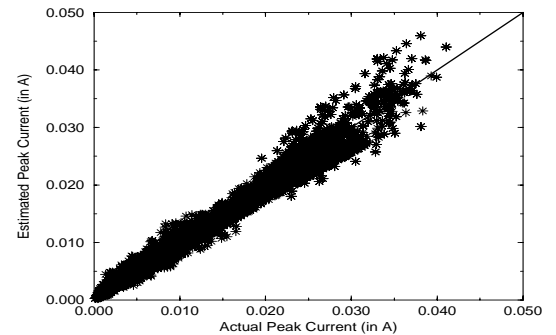

Figure 19. $i_{pactual}$ vs i_{pest} for ckts. listed in Table II

TABLE II

BENCHMARK CIRCUITS USED IN FIGS. 19–29

Circuit	#I	#O	#Cells	RMS_{avg} (A)	$e_{p,\mathrm{avg}}$	$e_{p,\mathrm{max}}$
c880	60	26	383	.0045	17.23%	27.90%
alu2	10	4	368	.0037	19.20%	35.74%
c432	36	7	217	.0028	12.43%	29.73%
cu	14	11	48	5.4423e-04	18.38%	31.93%
f51ml	8	3	105	0.0011	19.21%	39.43%
mux	21	1	91	4.4271e-04	17.95%	33.16%
random8	8	1	158	9.3857e-4	12.76%	37.11%
parity	16	1	68	7.7149e-04	15.37%	41.98%
vdao	17	27	341	0.0049	16.22%	25.67%
c499	41	32	202	0.0055	13.28%	14.52%
pcler8	27	17	101	9.8475e-04	8.47%	13.26%
sct	19	7	83	6.4798e-04	9.21%	15.67%
x2	10	3	50	5.4249e-04	14.35%	47.22%

TABLE III

ERROR METRICS FOR BENCHMARK CIRCUITS USED IN FIGS. 19–29

Circuit	$abse_{p,\mathrm{max}}$ (A)	$e_{t,\mathrm{avg}}$	R^2_{avg}	#sim.
c880	0.0009	10.20%	0.8047	3000
alu2	0.0007	8.13%	0.7721	650
c432	0.0024	11.46%	0.8117	2300
cu	0.0003	2.32%	0.8674	550
f51ml	0.0005	2.12%	0.7918	320
mux	0.0002	4.23%	0.8845	800
random8	0.0006	4.38%	0.8391	1000
parity	0.0004	3.98%	0.9034	640
vdao	0.0018	8.89%	0.7840	900
c499	0.0021	7.69%	0.8279	1500
pcler8	0.0010	5.21%	0.9021	900
sct	0.0009	6.31%	0.8967	600
x2	0.0001	2.43%	0.8734	500

TABLE IV

CHARACTERIZATION DATA FOR BENCHMARK CIRCUITS

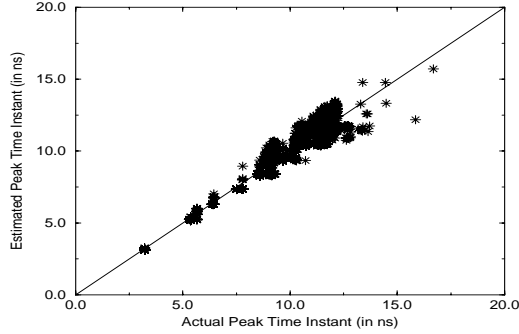| Circuit	model char.(days)	model eval.(hr)	$e_{di/dt,\mathrm{avg}}$ (A/ns)
c880	3.36	2.2	12.94%
alu2	1.19	0.52	14.89%
c432	2.7	1.93	17.38%
cu	0.17	0.47	22.56%
f51ml	0.35	0.39	20.25%
mux	0.67	0.63	17.38%
random8	1.05	1.04	12.83%
parity	0.39	0.48	14.20%
vdao	1.3	0.90	12.34%
c499	1.85	1.50	10.24%
pcler8	0.92	0.83	13.34%
sct	0.49	0.51	15.89%
x2	0.19	0.42	14.20%

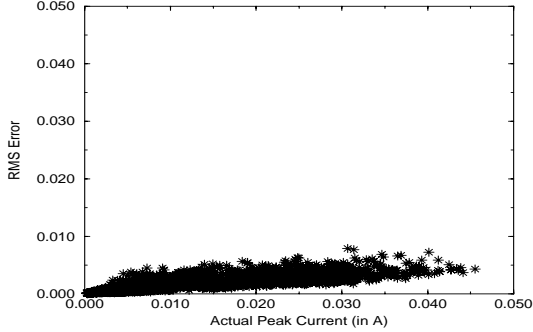Figure 20. $i_{t\mathrm{actual}}$ vs $i_{t\mathrm{est}}$ for ckts. listed in Table II

Figure 21. RMS error vs. peak current for ckts. listed in Table II

IV. EXPERIMENTAL RESULTS

A set of randomly generated vectors were used to simulate various benchmark circuits in HSPICE. The circuits are shown in Table II, where we have listed the number of inputs, outputs, and cells for each circuit. The choice of circuits was influenced by the fact that we are using HSPICE to build the models, and that therefore large circuits with large input counts would require unacceptable simulation times. In order to overcome this limitation, one can presumably use more efficient simulators, like PowerMill, or gate-level simulation techniques such as [15]. The resulting current waveforms were used to construct the current macro-model for the respective

circuits using the proposed approach. In Table III, the last column shows the total number of simulations (number of vector pairs) needed to build the macro-model for all the input vector partitions, for each circuit.

The current macro-models obtained after the characterization flow were then tested for accuracy using a different set of randomly generated vector pairs. The resulting current waveforms from the macro-model were evaluated using two types of criteria: quantitative and qualitative. Among the quantitative measures, we used the *Root Mean Square Error* (RMS) to compare the estimated waveform with the actual waveforms, given by:

$$RMSE = \sqrt{\frac{1}{K}\sum_{l=0}^{K-1}\|i(l)-\widehat{i}(l)\|^2} \qquad (27)$$

where $i(l)$ is the actual current waveform, and $\widehat{i}(l)$ is the estimated current waveform, and K is the length of the current waveform sequence without zero-padding. Table II, shows the average RMS error for various benchmark circuits, under the column "RMS_{avg}", which is computed as:

$$RMS_{\mathrm{avg}} = \frac{1}{P}\sum_{m=1}^{P} RMSE(m) \qquad (28)$$

where P is the number of vectors used to test the macro-model. In order to give some intuition as to the goodness of these average error numbers, and some measure of spread, we compared the RMS error for each vector pair with the

actual peak current. Such a comparison can used to determine the quantity of error, and the significance of the error. In Fig. 21, we show the combined RMS error vs peak current plot for the benchmark circuits listed in Table II. The plot shows that the RMS error is more or less bounded across the various current peak values for the benchmark circuits, and increases very slowly with the current peak. It should also be noted that in this plot, the slope of the line from the origin to that point is the relative percentage error of that point, with respect to the peak current value. Since we use regression to estimate the parameters of our macro-model, we also use the coefficient of determination to test the accuracy of the macro-model. The coefficient of determination (R^2), as explained earlier, measures the proportion of variability of the dependent variable explained by regression on predictor variables. The closer R^2 is to 1, the better the model captures the current waveform. Since R^2 is a statistic for each current waveform, we use R^2_{avg}, the average coefficient of determination. In Table III, we have included the average coefficient under the column R^2_{avg}.

An estimate of the time instant and the value of the peak current for various blocks would facilitate a fast power bus analysis. Therefore we used the relative error in peak current value and the time at which the peak occurs as another set of quantitative metrics to test the accuracy of our macro-model. The relative error is defined as:

$$err = \frac{\|x_{\mathrm{actual}} - x_{\mathrm{est}}\|}{x_{\mathrm{actual}}} \qquad (29)$$

where x is either the peak current, i_p, or the time instant at which the peak occurs, t_p. In Table II, $e_{p,\mathrm{avg}}$ denotes the average error in peak current estimation, and in Table III $e_{t,\mathrm{avg}}$ denotes the average error in the estimation of the time instant at which the peak occurs. The average error in each case is computed using:

$$e_{x,\mathrm{avg}} = 100\frac{1}{P}\sum_{i=1}^{P}\mathrm{err}_i \qquad (30)$$

Table II also includes the maximum relative error in peak current estimation for each benchmark circuit, denoted by $e_{p,\mathrm{max}}$. In some cases maximum relative long is close to 40%, therefore we have included the absolute error in peak current estimation at which the maximum relative error occurs in Table III. The absolute error in peak current is denoted by $abse_{p,\mathrm{max}}$. Comparison of the absolute error and the maximum error in peak current shows that, maximum relative error occurs when the absolute error is very low. Therefore, maximum error occurs when the magnitude of peak current is very low, and even though absolute error in peak current estimation is small it gets magnified when using the relative error metric. But overall, Table II shows that the average error in peak current estimation over the set of vectors is less than 20% for all the circuits considered. Also Table III shows that the average error in t_p is less that 15% in all cases. In order to show that the $i_{p,\mathrm{est}}$ and $t_{p,\mathrm{est}}$ can be estimated accurately for different vector pairs, using our macro-model, we have also included the corresponding correlation plots of the estimated values versus actual values in Figs. 19 and 20. In Fig. 19, we show a combined plot of $i_{p,\mathrm{actual}}$ v.s. $i_{p,\mathrm{est}}$ for the benchmark circuits listed in Table II, and similarly in Fig. 20 we show a combined plot of $t_{p,\mathrm{actual}}$ v.s. $t_{p,\mathrm{est}}$. The correlation plots are close to linear. The time instant plots show fewer number of points because most of the points are superimposed on top of each other. Table IV, shows the model characterization and evaluation times. It also includes the average relative error in the estimation of peak current slope, in its fourth column. The peak current slope is defined as the ratio of peak current to the time instant at which the peak occurs, and in the given context it can be be calculated as i_p/t_p. The model characterization (denoted in the second column) time is entirely dominated by circuit simulation time, and the model evaluation time (denoted in the third column) is dominated by the inverse DCT evaluation time. The DCT sample length was constant across all the benchmark circuits, therefore the model evaluation time is a function of the number of current samples generated for each benchmark circuit. These runtimes were obtained on a Sun UltraSparc-II, 450MHz machine. The model characterization and evaluation times correspond to the number of current samples (number of simulations) specified in the last column of Table III. In Fig. 23, we show a plot of absolute error in peak current vs the peak current. In this plot the slope of a line from origin to a given point, represents the relative error in peak current estimation. It can be seen, that the absolute error increases very slowly with the current peak. Finally, in Figs. 23–29, we present a qualitative comparison of the actual and estimated current waveform for the benchmark circuits listed in Table 2, for some example Hamming distances. The current waveforms for qualitative comparison were obtained by simulating the circuit for a given vector pair with a certain Hamming distance, as shown in the figures. In this comparison, we superimpose the the estimated current waveform on top of the actual current waveform obtained from HSPICE for randomly generated vector pairs. This is a qualitative measure because the plots are mostly useful for visual comparison. The comparison plots show good accuracy in most cases. In Fig. 23, we show the current waveform plot for alu2 where only 3 out of 10 inputs are switching. The figure shows some deviation from the actual plot because the Hamming distance is low. The same observation can be made regarding other low Hamming distance waveforms. Thus, small Hamming distance waveforms show more deviation from the actual current waveform, because for large Hamming distances the input variables are mostly confined to just two values. In case of low Hamming distances, the inputs can take any of four values. Moreover when a small fraction of inputs are switching, the current waveform significantly depends on what inputs are switching, which leads to large variations in the current amplitude. However, the current magnitude itself is much lower in case of low Hamming distance, so that the absolute error is actually small.
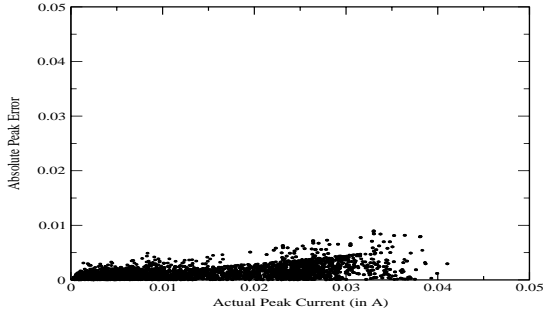
**Figure 22.** Abs. error in peak current vs. peak current for ckts. listed in Table II
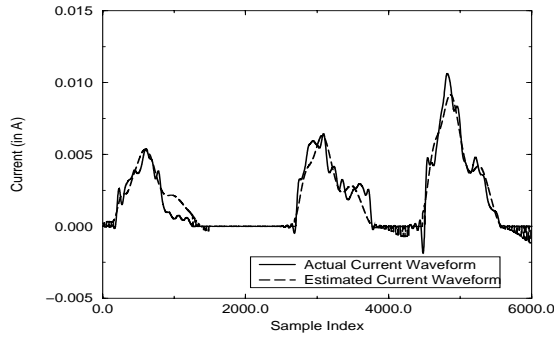
Figure 23. Act. vs. est. current waveform for alu2, Ham. dist=3

In Figs. 26 and 27, we show the results obtained for some skewed transitions. Figure 26 is obtained by switching the top thirty bits alternately from low to high and high to low. In Fig. 27 the first waveform is obtained, by switching half the bits from low to high and the second waveform is obtained by switching half the bits from high to low. In Figs. 28 and 29, we show the results obtained by assuming different arrival times for the various inputs. The input arrival times were randomly varied from 0ns to 1ns. It can be seen that variations in input arrival times reduces the current peak. Therefore, if the model is characterized assuming identical arrival times for all the inputs, it may lead to some inaccuracies. But Figs. 28 and 29 also show that, at a high-level of abstraction, the errors due to different arrival times may be tolerable. Even though for lower Hamming distances we see some deviations from the actual current waveforms and the model ignores the different arrival times of various inputs, we must point out that the model in fact works very well, capturing the required current waveforms in a high-level black box macro-model.
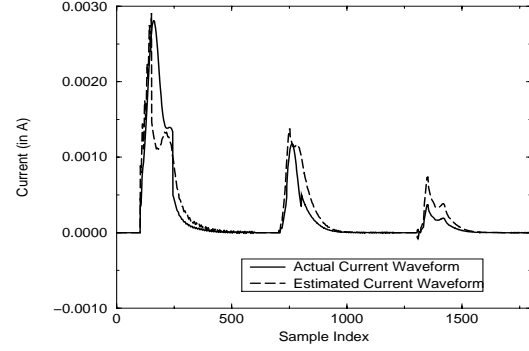
Figure 24. Act. vs. est. current waveform for cu, Ham. dist=2
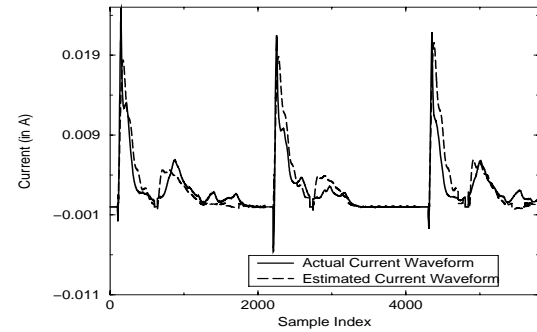
Figure 25. Act. vs. est. current waveform for c432, Ham. dist=32
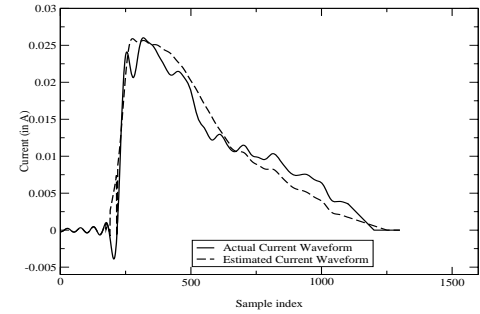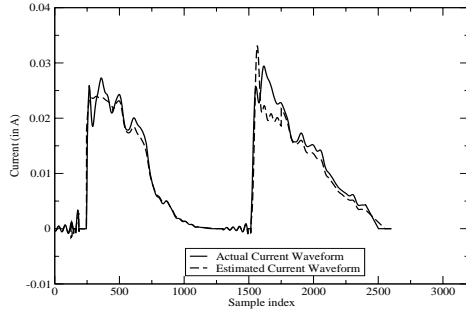

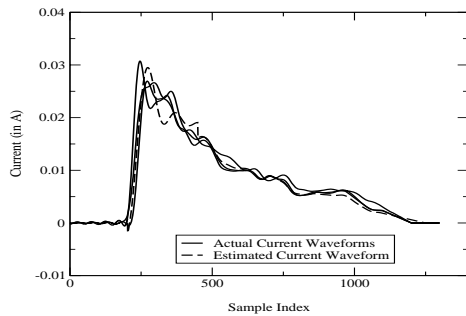**Figure 26.** Act. vs. est. current waveform for c880, Ham. dist=30

V. CONCLUSION

In order to enable early block-level analysis of the power grid, when using hard IP blocks, we have proposed a cycle-based current waveform modeling technique for combinational logic blocks that involves predicting the frequency transform from the input vector pairs, and using the inverse transformation to get back the time-domain waveform. The use of frequency domain analysis is motivated by our observation that, while the time domain waveforms show large variations in shape (making them hard to model in the time domain), the frequency domain transforms (such as DCT) show much less variation and are mostly limited to variations in their parameter values. For the DCT, we have shown that it is
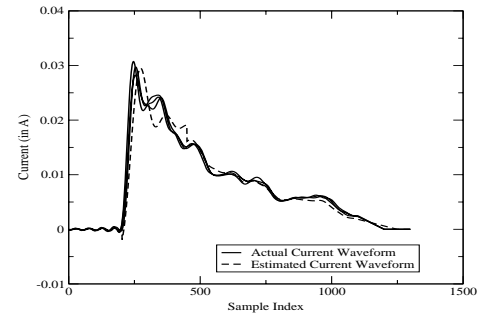
possible to build models for its parameters in terms of the input vector pairs. The models include coefficients whose values are computed by a process of characterization based on HSPICE simulations on a number of randomly generated vector pairs. While we continue to improve the estimation and validate this technique on more circuits, the data so-far show that this type of estimation is indeed possible, enabling early block-level simulation (hence fast, and for a large number of vectors) of power grids under transient current conditions.

Figure 29. Est. current waveform with diff. arrival times for c880, Ham. dist=32

Figure 27. Act. vs. est. current waveform for c880, Ham. dist=30

Figure 28. Est. current waveform with diff. arrival times for c880, Ham. dist=30

REFERENCES

[1] A. Dharchoudhury, R. Panda, D. Blaauw, R. Vaidyanathan, B. Tutuianu, and D. Bearden, "Design and analysis of power distribution networks in PowerPC microprocessor," *ACM/IEEE Design Automation Conference*, pp. 738 - 743, June 1998

[2] S. R. Powell and P. M. Chau, " Estimating Power Dissipation of VLSI signal Processing Chips: The PFA technique," *VLSI Signal Processing IV*, pp. 250-259, 1990.

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

[4] A. Raghunathan, S. Dey and N. K. Jha, "Register-Transfer Level Estimation Techniques for Switching Activity and Power Consumption," *IEEE International Conference on Computer-Aided Design*, pp. 158-165, November 1996.

[5] S. Gupta and F. N. Najm, "Analytical models for RTL power estimation of combinational and sequential circuits," *IEEE Transactions on Computer-Aided Design*, vol. 19, no. 7, pp. 808–814, July 2000.

[6] A. Bogliolo and L. Benini, "Node Sampling: a Robust RTL Power Modeling Approach," *IEEE International Conference on Computer-Aided Design*, pp. 461-467, November 1998.

[7] Q. Qiu, Q. Wu, Chih-S. Ding, and M. Pedram, "Cycle-accurate macro-models for RT-level power analysis," *IEEE Trans. VLSI Systems*, vol. 6, pp. 520-528, no. 4, December 1998.

[8] S. Gupta and F. N. Najm, "Energy-per-cycle estimation at RTL," *IEEE International Symposium on Low Power Electronics and Design*, SanDiego, CA, pp.21-126, August 1999.

[9] A. V. Oppenheim and R. W. Schafer, "Discrete-Time Signal Processing," *Prentice-Hall, Inc*, First edition, Nov. 1990.

[10] A. K. Jain, "The Fundamentals Of Digital Image Processing," *Prentice-Hall, Inc*, 1989.

[11] R. F. Gunst and R. L. Mason, "Regression Analysis And Its Application, A Data-oriented approach," *Marcel Dekker, Inc*, 1980.

[12] S. Bodapati and F. N. Najm, "Frequency-Domain Supply Current Macro-Model," *IEEE International Symposium on Low Power Electronics and Design*, Huntington Beach, CA, pp. 295-298, August 2001.

[13] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0," *Microelectronics Center of North Carolina, Tech. Rep.*, 1991.

[14] S. Weisberg, *Applied linear regression, 2nd Edition.* New York: John Wiley & Sons, 1985.

[15] A. Bogliolo, L. Benini, G. D. Micheli and B. Ricco, "Gate-Level Current Waveform Simulation Of CMOS Integrated Circuits," *IEEE International Symposium on Low Power Electronics and Design*, Monterey , CA, pp. 109-112, August 1996.

[16] S. Bodapati and F. N. Najm, "High-level current macro-model for power-grid analysis," *ACM/IEEE Design Automation Conference*, New Orleans, LA ,pp. 385-390, June 2002.

PLACE PHOTO HERE

Srinivas Bodapati received the B.Tech (Honors) degree in Electrical Engineering from the Indian Institute of Technology, Kharagpur in 1995, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering (ECE) from the University of Illinois at Urbana-Champaign (UIUC) in 2000 and 2003, respectively.

He has worked at Cadence Design Systems, India from 1995 to 1997 and at Texas Instruments, India from 1997 to 1998. Since 2003, he has been at Intel Corporation in Santa Clara, CA. His research interests are in power estimation and modeling, signal integrity and circuit reliability analysis.

PLACE PHOTO HERE

Farid N. Najm received the B.E. degree in Electrical Engineering from the American University of Beirut (AUB) in 1983, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering (ECE) from the University of Illinois at Urbana-Champaign (UIUC) in 1986 and 1989, respectively. He worked with Texas Instruments in Dallas, TX, 1989-1992, then joined the ECE Department at UIUC as an Assistant Professor, becoming Associate Professor in 1997. In 1999, he joined the ECE Department at the University of Toronto, where he is now Professor and Vice-Chair of ECE.

Dr. Najm is a Fellow of the IEEE, and is Associate Editor for the IEEE Transactions on CAD. He received the IEEE Transactions on CAD Best Paper Award in 1992, the NSF Research Initiation Award in 1993, the NSF CAREER Award in 1996, and was Associate Editor for the IEEE Transactions on VLSI 1997–2002. He served as General Chairman for the 1999 International Symposium on Low-Power Electronics and Design (ISLPED-99), and as Technical Program Co-Chairman for ISLPED-98. He has also served on the technical committees of ICCAD, DAC, CICC, ISQED, and ISLPED. Dr. Najm has co-authored the text "Failure Mechanisms in Semiconductor Devices," 2nd Ed., John Wiley & Sons, 1997. His research is on computer-aided design (CAD) for integrated circuits, with an emphasis on circuit level issues related to power dissipation, timing, and reliability.