BOTTOM-UP HIGH-LEVEL CURRENT MACRO-MODELS FOR LOGIC BLOCKS

SRINIVAS BODAPATI

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801
University of Illinois at Urbana-Champaign
4. TITLE AND SUBTITLE
Bottom-Up High-Level Current Macro-Models for Logic Blocks

6. AUTHOR(S)
Bodapati, Srinivas

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W Main St
Urbana, IL 61801

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
SRC
Brighton Hall, Suite 120
1101 Slater Rd
Durham, NC 27703

11. SUPPLEMENTARY NOTES
Approved for public release; distribution unlimited.

13. ABSTRACT (Maximum 200 words)
This research addresses the problem of current estimation at a high level of abstraction, useful in the early design and analysis of power distribution networks. These current estimates can be used at register transfer (RT) level to design block-level power distribution networks, and also to obtain time based true transient power. We target bottom-up current macro-modeling, which is useful when one is reusing a previously designed logic block, so that all the internal structural details of the circuit are known. In this case, one develops a current macro-model for the block that can be used to estimate the current drawn by the block without performing a more expensive transistor or gate-level simulation.

High-level current estimation capability is required to provide early warning of any top-level power distribution problems like voltage drop, electromigration before the circuit-level design has been specified. Because, by the time the design is specified at the gate or transistor levels, it may be too late and expensive to fix problems associated with the power distribution network (power grid). It may require significant redesign effort and time, to fix problems associated with the power grid.

14. SUBJECT TERMS
Current macro-model; power-grid; IR-drop; electromigration; bottom-up; DCT; frequency domain current estimation

15. NUMBER OF PAGES
91

16. PRICE CODE
UL

17. SECURITY CLASSIFICATION OR REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

NSN 7540-01-280-5500
BOTTOM-UP HIGH-LEVEL CURRENT MACRO-MODELS
FOR LOGIC BLOCKS

BY
SRINIVAS BODAPATI
B.Tech., Indian Institute of Technology, Kharagpur, 1995
M.S., University of Illinois at Urbana-Champaign, 2000

THESIS
Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2003

Urbana, Illinois
This research addresses the problem of current estimation at a high level of abstraction, useful in the early design and analysis of power distribution networks. These current estimates can be used at register transfer (RT) level to design block-level power distribution networks, and also to obtain time based true transient power. We target bottom-up current macro-modeling, which is useful when one is reusing a previously designed logic block, so that all the internal structural details of the circuit are known. In this case, one develops a current macro-model for the block that can be used to estimate the current drawn by the block without performing a more expensive transistor or gate-level simulation.

High-level current estimation capability is required to provide early warning of any top-level power distribution problems like voltage drop, electromigration before the circuit-level design has been specified. Because, by the time the design is specified at the gate or transistor levels, it may be too late and expensive to fix problems associated with the power distribution network (power grid). It may require significant redesign effort and time, to fix problems associated with the power grid.

High-level power-grid design and analysis requires a current macro-model for different logic blocks that are both easy to use and automatically constructed. The main contribution of this research is the development of current macro-models for combinational logic blocks in the frequency domain. The current waveforms obtained from the macro-models can be used to estimate energy for every input vector pair (energy per cycle) and peak-current for every input vector pair (peak current per cycle), apart from being useful in analyzing power grids at higher levels of abstraction. Another contribution of this research is the automatic construction of these macro-models based on a simple characterization flow.
To my family
ACKNOWLEDGMENTS

I would like to express my gratitude and deep appreciation for my academic adviser, Dr. Farid Najm, for his support and guidance during the time of this study. I will always be indebted to him for providing me the opportunity to work with him when I was lost and looking for some direction. He has been a tremendous mentor, and it has been a privilege to work with him and be associated with him. My association with him has been one of the best things that has happened to me. I have learned from him more than the subject matter of this research. Through his kindness and help, he instilled in me the confidence and zeal to succeed. Thank you very much, Dr. Najm. I would also like to thank my committee members, Dr. Elyse Rosenbaum, Dr. Jose E. Schutt-Aine, and Dr. Naresh R. Shanbhag, for their advice, patience, and suggestions in the course of this work.

I would like to thank my father, uncle Mausaaji, aunt Ammalu, and sister Madhu for their constant support and encouragement. I am grateful to them for their invaluable guidance and love at the time of need. I am also indebted to my late mother who instilled the true meaning of hard work very early in my life and inspired me to reach this stage.

I would also like to thank my friends Sudhakar, Subodh, Kavel, and Srini for their many technical and nontechnical discussions. I would like to extend special thanks to Subodh and Sudhakar for their invaluable help during the course of this work.

I would also like to extend special thanks to my in-laws, sister-in-law, grandmother, and aunt Jaya for the infinite help they have extended to me during the course of this work. Last, but not least, I would like to acknowledge the support of my wife Sravanthi. I will always owe her for her constant encouragement, love, and support that saw me through the difficult times.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Current Estimation</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Dissertation Overview</td>
<td>8</td>
</tr>
<tr>
<td>2 DISCRETE COSINE TRANSFORM</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Discrete Cosine Transform</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Properties of Current Waveform DCT</td>
<td>15</td>
</tr>
<tr>
<td>2.3 DCT and Fourier Transform</td>
<td>16</td>
</tr>
<tr>
<td>3 ANALYSIS OF CURRENT WAVEFORMS</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Triangular Current Waveform</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Trapezoidal Current Waveform</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Multiple Peak Current Waveform</td>
<td>27</td>
</tr>
<tr>
<td>4 MACRO-MODEL CONSTRUCTION</td>
<td>34</td>
</tr>
<tr>
<td>4.1 Dummy Coding</td>
<td>35</td>
</tr>
<tr>
<td>4.2 Triangular Template</td>
<td>39</td>
</tr>
<tr>
<td>4.2.1 Decay factor</td>
<td>40</td>
</tr>
<tr>
<td>4.2.2 Amplitude and DC component</td>
<td>41</td>
</tr>
<tr>
<td>4.2.3 Period</td>
<td>43</td>
</tr>
<tr>
<td>4.3 Trapezoidal Template</td>
<td>44</td>
</tr>
<tr>
<td>4.4 Characterization Flow</td>
<td>48</td>
</tr>
<tr>
<td>4.4.1 Sample size</td>
<td>50</td>
</tr>
<tr>
<td>5 EXPERIMENTARY RESULTS</td>
<td>55</td>
</tr>
<tr>
<td>6 CONCLUSIONS AND FUTURE WORK</td>
<td>78</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>82</td>
</tr>
<tr>
<td>VITA</td>
<td>85</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

As the minimum feature size of very large scale integrated (VLSI) circuits continues to shrink, and as device densities continue to increase several-fold, the increased power dissipation of theses devices has begun to pose serious concerns with regard to reliability and marketability.

In the case of microprocessor circuits, computing power has increased tremendously, primarily by increasing the device count and the clock frequency on the chip. On the other hand, efforts to push the computing throughput to its extreme are causing the architecture of the processors to become deeply pipelined. Heavy pipelining of the circuit means that the devices on the circuit participate more actively in the computations. All of the above situations, viz. the higher device count, the higher clock frequency, and the higher switching rate lead to the increased power consumption of integrated circuits to undesirable levels.

The high levels of power dissipation result in high chip temperature, which, if unchecked, can lead to reliability problems. Also, high temperature degrades the performance of the chip as some of the device parameters are affected by temperature. Thus, the chip might be forced to operate at speeds lower than what it was originally designed for. In order to overcome these problems affecting reliability and performance, one has to resort to expensive packaging. This in turn affects the cost of the chip, and hence profitability.

In the case of digital signal processing chips and microprocessors used in portable applications, smaller and longer-life batteries are desirable. This situation again requires that the circuits consume very low power. Also, in case of desktop applications, there is a need for low power chips to keep the cooling cost low.

Thus, from both a 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 has been 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 dramatic increase in supply currents. Today’s high-end microprocessors, for example, can consume supply current of 100 A or more, and it is getting worse. The
Figure 1.1 ITRS-2001 roadmap for power dissipation

International Technology Roadmap for Semiconductors (ITRS-2001) [1] indicates that the total power supply current delivered to a high-performance integrated circuit will grow to about 200 A in 2006, and almost 500 A in 2012. This is the result of increasing power consumption and decreasing supply voltage, as shown in Figs. 1.1 and 1.2. These large current values coupled with the lower power supply voltages and thinner wires employed in deep submicron designs adversely affect the robustness of the power distribution network, which we refer to simply as the power grid. The resistance (and in the future, inductance) of the metal lines in the power grid cause large voltage drops at the devices due to high instantaneous currents. These supply voltage variations not only affect the power grid integrity but also prolong the timing closure loop. A large voltage drop slows down the devices significantly and reduces the noise margin of circuits, which might lead to functional failures/soft errors. In fact, a 5% drop in the supply voltage of a gate can cause a 15% increase in the gate delay [2]. This large increase in delay, if not accounted for, can lead to delay failures. Since the supply voltage is being reduced for various reasons (to reduce power, to maintain gate oxide integrity, etc.), a reduction in the supply voltage reduces
Figure 1.2 ITRS-2001 roadmap for supply voltage

noise margin. At 1 V supply, a 200 mV drop becomes a whopping 20% of the supply, which adversely affects the noise margin.

From the ITRS projection for power and voltage one can obtain the projections for current flowing in the power grid as shown in Fig. 1.3. The ITRS projection indicates hundreds of amperes of current flowing through the power grid. These sustained large currents flowing in the power grid, coupled with thinner wires, increase the current density in the power grid. Therefore, metal lines in the power grid become more prone to the formation of voids and shorts to nearby wires due to electromigration caused by the large sustained currents. Electromigration causes the atoms in the metal to gradually shift in the direction of sustained current, wearing out metal and leading to shorts and voids. Large current densities in the power grid may also cause failures due to joule heating of the metal lines. As a result, power grid analysis and design is now an important concern during the integrated circuit (IC) design flow. But the sheer size of the power grid makes the design and analysis of a power grid very difficult. Traditionally, the size of the power grid was easy to manage, as it was confined to only one or two levels of metal. But today, power grids are huge (a
design with 10 million transistors may have a power grid with 30 million branches), and they require a significant number of routing tracks and metal layers. Actually the power grid is an extremely large network of nonlinear devices and linear interconnects forming the power grid. Accurate analysis of the power grid therefore requires a simultaneous solution of a network of linear and nonlinear components. This takes into account the coupled nature of the circuit and provides an accurate picture, but it also limits the size of the circuit that can be handled.

But for digital circuits, a more efficient two-step techniques is used. The first step is to calculate the current consumption for all circuit elements assuming ideal power supply values. The second step is to use computed loading currents to calculate the current distribution in the power grid. The resulting voltage drop can be fed back into the first step to improve the accuracy of the current data drawn from the power grid. But usually, one iteration loop is sufficient to get realistic voltage drop and current data for the power grid [2]. The separation into two steps allows the exploitation of specialized algorithms, which in turn improves the speed and capacity.
Although the above challenges have attracted enough interest in the area of power grid verification and analysis, most of the focus has been on the post-layout verification of the power grid when the entire chip design is complete. Unfortunately, power grid problems revealed at this stage are usually very expensive and difficult to fix because of the increase in design/circuit/layout complexity. Now, the physical design of the power grid is related to (and has a significant impact on) the design and layout decisions made for the chip blocks. Therefore, it is important to do early design planning of the power grid in order to reduce the chances of having to redesign large parts of it [3]. Early design of the power grid is needed to determine the location of the clean VDD/GND pads, nominal pitches and widths of metal layers, via styles (point or bar vias), and parameters of the chip package etc [3]. In this context there is a need for new techniques that aid in the early design and planning of the power grid. One of the key technologies needed for early planning of the power grid is the availability of fairly accurate and fast current estimates at a much higher level of abstraction (basically early in the design cycle). Actually, some commercial tools [4] are available that can be used at any stage of the physical design process but they still are not very helpful in making design decisions very early in the design cycle. Moreover they still depend on good current estimates early in the physical design process to enable the design and analysis of power grid.

Most of the existing current estimation techniques work at the transistor level SPICE, Powermill [5] or gate-level. However, these simulators are too slow, and current estimations obtained at this level to detect power grid problems are very expensive to fix. In the following we present a brief overview of the various current estimation techniques proposed to work at lower levels of circuit abstraction.

1.1 Current Estimation

In order to reduce the complexity of solving a nonlinear network while analyzing the power grid, it is decoupled into a nonlinear circuit and linear interconnect network. The nonlinear circuit is simulated using various transistor-level or gate-level circuit simulators to get the appropriate currents, and then these currents are applied to the linear interconnect network to obtain the node voltages and branch currents [3], [6]. Some techniques have been proposed to solve the large linear interconnect network [7], [8], but the current estimation is
still obtained using various gate or transistor level analysis tools like SPICE, Powermill [5], IRSIM [9]. In this section we discuss some of these current estimation techniques and tools.

Powermill is a commercial transistor-level simulator for the simulation of current and power, where the transistor characteristics are captured using a piecewise-linear model in a look-up table. This reduces the model computation time. Powermill uses event-driven techniques to decouple a large circuit into collections of smaller ones that are evaluated separately. The decoupling is usually done based on channel-connected regions that are evaluated using numerical integration techniques. Thus, Powermill reduces runtime compared to SPICE by not solving the companion models used by circuit simulators. IRSIM is again another transistor level circuit simulator similar to Powermill. In IRSIM the circuit is represented as a set of nodes interconnected with transistors. Nodes are modeled as capacitors to ground while transistors are modeled as bidirectional switches with finite ON resistance which is a function of transistor size and type. The circuit is again simulated using event-driven techniques.

Along with the transistor level current estimation techniques, various gate-level current estimation techniques have also been proposed. The gate-level current simulators discussed in [10]-[16] are all essentially based on the following observation: the current drawn by a complex CMOS gate for any given input transition has almost the same behavior as the current drawn by an elementary gate with the same driving capability and switching capacitance. Hence, only a small set of reference gates actually need to be characterized, while any other gate simply needs to be reduced into the equivalent elementary one whenever a transition occurs at its inputs. These techniques provide good approximations of the current behavior of gates with single input transitions, but they lose accuracy when dealing with internal charge redistributions and multiple transitions. Moreover, gate-collapsing techniques are usually not compatible with logic-level design tools. In [17], the time-domain behavior of the current drawn by a CMOS gate is modeled as an asymmetric triangular pulse. The parameters of the triangular pulse are a function of the input slew and output load. The modeling is done for all the cells in a predefined cell library (usually an application specific integrated circuit (ASIC) cell library), instead of a few reference cells as suggested in [10]-[16]. These models are then used along with an event-driven logic simulator (Verilog-XL) to compose the overall time domain current waveform of the circuit.
The problem with both gate-level and transistor-level current simulators is that they are computationally very expensive. Moreover, they require a gate-level or transistor level description of the design to be useful. Therefore, they cannot be used in the design of a power grid very early in the design cycle. Also at such a low level description it is very expensive both computationally and economically to explore design trade-offs with respect to the design of a power grid. Hence, in order to avoid costly redesign steps, current estimation techniques are required that enable current simulations at a higher level of abstraction. This will provide the designer with more flexibility to explore design trade-offs early in design process thus reducing the design cost and time.

Apart from these gate-level current estimation techniques, two types of methods have been proposed for estimating the maximal currents at the gate level. In the search-based approach, the input-pattern is derived for which the sum of the peak currents of the gates is maximal [12], [18]. In the pattern-independent approach [19], [20] the maximal current of a logic circuit at each time of the clock cycle is computed.

Most of the maximal current estimation techniques work towards providing an upper/lower bound for the maximum instantaneous current, but for large circuits the bounds are not very tight. Thus, the estimates are either overly pessimistic or optimistic and may not be very helpful in making optimal design decisions. Moreover in the presence of inductance, the maximum voltage drop may not occur at the maximum current; therefore, it is important to have an estimate of the entire time domain current waveform.

Thus there is a need for fast and accurate current estimates at a high level of abstraction for early planning and design of the power grid. In general, such current estimates would be difficult if the design description is only a Boolean function, with no information on the structure, number of gates/nodes, etc. In [3] the authors have used a simple area based DC estimate of the current, which is based on the current estimate of a previous chip and scaled accordingly, for early design and planning of the power grid. In [21] the authors have proposed current models for already existing functional blocks based on simulation. In this technique, the current drawn from a functional block is modeled as a triangular wave or trapezoidal wave, depending on the ratio of average current and peak current. But this model is not very accurate and cannot be used to generate current waveforms for input vectors pairs beyond the ones used to simulate the circuit. In order to improve this technique
the authors also propose using a piecewise-linear model for the current waveform obtained after actual simulation. One major drawback of this model is that it always needs circuit simulation (therefore a low-level description) to get the current waveforms and thus cannot be used at a high level of abstraction.

In an environment where design blocks are being reused (hard intellectual property (IP) blocks), it might be possible to build accurate block-level models that can give the current waveform drawn by a logic block in response to a given input vector stream. Such a modeling technique is usually referred to as a bottom-up technique, where one attempts to measure the actual current waveform drawn by an existing implementation (of a logic block) in response to an input vector pair, and produce a model based on the measurements. In other words, this technique attempts to build current macro-models for accurate current estimation. These current macro-models can be used to perform early and fast block-level analysis (simulation) of the currents and voltages in the power grid without performing a more expensive gate-level or transistor-level simulation. Clearly, this approach is best suited for designs that are built using a library-based approach.

1.2 Dissertation Overview

In this dissertation, we propose such a bottom-up current waveform macro-model for (combinational) logic blocks. The proposed current macro-model captures the dependence of the current waveform per-cycle, on the vector pair applied at block inputs. Previous work in bottom-up macro-modeling has targeted either the average power [22]–[26] or energy-per-cycle [27], [28]. 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 a macro-modeling approach 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 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 lower-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, we invert the transform to obtain the time-domain current waveform.

More specifically, we use the discrete cosine transform (DCT), for our model. The reasons for using DCT will become clear in subsequent chapters. In our approach we use a template equation for 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, the triangular wave and the trapezoidal wave. The parameters of the template function are modeled as a function of the input vector pair. The input vector pair is classified based on the fraction of primary inputs switching. The parameters of the DCT template are modeled as a function of the input vector pair for each component of the classification using a different set of equations.

Given a low-level (in our case transistor-level) description of a CMOS combinational circuit, we present a characterization process that can be used to build current macro-models. These models can be used with a behavioral level logic simulator to obtain the current waveform. These waveforms can then be used with a linear solver to analyze the power-grid.

The dissertation is organized as follows. In Chapter 2 we describe the frequency transforms that will be used as the basis for inferring a DCT model template, and also discuss some of the properties of DCT which are relevant to current waveforms. In Chapter 3, we illustrate what typical shapes these transforms take in practice. We also present analytical expressions which approximate these transform shapes reasonably well. In Chapter 4, we present the DCT model template that we have used, and describe the model construction process. In Chapter 5 we present experimental results that illustrate the validity of this approach, and finally, in Chapter 6 we summarize and present future directions.
CHAPTER 2

DISCRETE COSINE TRANSFORM

In this chapter, we present the discrete cosine transform (DCT) and its relationship with the continuous Fourier transform. This relationship forms the basis for inferring a DCT model template. This model template is used to capture the DCT samples of the current waveform in the form of an analytical expression, and will be discussed in Chapter 4. We also present some example current waveforms and their corresponding DCT to show that large variations in waveform shapes in the time domain translate to variations mostly in the parameters of the frequency-domain transforms, but not in their overall shape. This observation forms the basis of our current macro-modeling technique in frequency domain as it allows us to use a generic template function to model the DCT of a highly irregular time domain current waveform. The generic template is governed by a given set of parameters that vary from one vector pair to another. Therefore, these parameters of the template function are modeled as a function of the input vector which is discussed in detail in subsequent chapters. But before we go into the details of the template, we present the DCT, which forms the basis of the current macro-modeling technique.

2.1 Discrete Cosine Transform

The current waveform obtained from circuit simulation is a discrete-time signal, because most of the circuit simulators report current values at discrete time instants. The interval between any two time instant is usually the step-size specified in the transient simulation deck. Thus the current waveform (actually the samples) obtained from a circuit simulator like HSPICE can be considered to be obtained from the periodic sampling of a continuous-time current waveform, i.e,

\[ i[n] = i_c(nT), \quad 0 \leq n \leq N - 1 \]  \hspace{1cm} (2.1)

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 [29]. Usually \( T \) is equal to the (fixed) step size specified in the transient analysis. In our simulations we used
a time step of 0.01 ns; thus, it is also the sampling time period in our case. In some cases the circuit simulator step size may not be uniform in spite of a uniform step size specified in the simulation deck. In such cases one can artificially generate uniform samples of the current waveform. Therefore without loss of generality one can assume a uniform step size as far as circuit simulation is concerned.

The above definition of current waveform relates the actual current waveform which is a discrete signal to a continuous hypothetical signal. This enables us to establish a relation between the DCT of the discrete signal and the continuous Fourier transform of the corresponding continuous signal. This relationship is exploited to gain some insight into the form of the DCT model template, by approximating the current waveform with piecewise-linear waveforms. The discrete samples of the piecewise-linear waveform approximate the current samples. The DCT of such samples can be easily related to the Fourier transform of the corresponding piecewise-linear waveform using the generic relationship shown in this chapter. Such a relationship is useful because DCT is a discrete transform. Samples obtained from DCT do not give enough insight into the form of the actual analytical expression from which the samples have been generated. But if DCT could be related to the continuous Fourier transform, it might be possible to get an analytical expression for the DCT samples, which in turn could be used to develop the actual current macro-model. Now in order to establish the relationship between DCT and Fourier transform, we start with defining the DCT of a discrete time current signal. The one-dimensional DCT [30] of a sequence \( \{i[n], 0 \leq n \leq N - 1\} \) is defined as

\[
I[k] = \alpha(k) \sum_{n=0}^{N-1} i[n] \cos \left[ \frac{\pi (2n + 1) k}{2N} \right], \quad 0 \leq k \leq N - 1 \tag{2.2}
\]

where

\[
\alpha(0) = \sqrt{\frac{1}{N}}, \quad \alpha(k) = \sqrt{\frac{2}{N}} \quad \text{for} \quad 1 \leq k \leq N - 1
\]

and the inverse transformation is given by

\[
i[n] = \sum_{k=0}^{N-1} \alpha(k) I[k] \cos \left[ \frac{\pi (2n + 1) k}{2N} \right], \quad 0 \leq n \leq N - 1 \tag{2.3}
\]

The basic premise for using DCT in our macro-modeling is the fact that large variations in the time domain current waveform translate to variations mostly in the parameters of
the DCT, but not in its overall shape. In order to justify this basic premise we present some current waveforms and their corresponding DCT. In Fig. 2.1, we show 10,000 sample points in the time domain current waveform for the MCNC benchmark circuit alu2 [31], even though the actual current waveform is nonzero for less than 2000 samples (which corresponds to 20 ns for our sampling period of 0.01 ns). Basically, we have increased the number of samples in the current waveform by padding zeros at the end of the sequence. This is referred to as zero padding, and its main advantage is that it increases the resolution of the DCT, and gives a smoother curve, which helps in getting an insight into the form of macro-model (explained later in this section). The resolution of DCT increases because DCT is related to discrete Fourier transform (DFT) which in turn is related to the samples of discrete time Fourier transform (DTFT) as discussed in Section 2.3. Now zero padding in the time domain is equivalent to a higher sampling rate of the DTFT to get the corresponding DFT and hence the DCT [29]. In Fig. 2.2, we present the DCT of the corresponding zero-padded time domain current waveform. We show only 100 points of the 10,000 point DCT for clarity.

![Figure 2.1](image1.png) ![Figure 2.2](image2.png)

**Figure 2.1** Current waveform for alu2  **Figure 2.2** DCT of the current waveform

In Fig. 2.1, the actual current waveform is not very clear; therefore, in Figs. 2.3–2.12 we show the time domain current waveform without zero padding and the corresponding DCT obtained after zero padding for MCNC benchmark circuits mux and vdao [31]. One can verify from these plots our earlier assertion that though the shape of the time domain current waveform is highly irregular, the shape of the DCT of the current waveform obtained after zero padding is smooth, which makes it an excellent candidate for model construction. Though all the DCT plots have 10,000 points, we have shown the first 100 points of the 10,000 point DCT for clarity.
Finally, in Fig. 2.13 we compare the DCT of a current waveform with and without zero padding. It is obvious from the figure that with zero padding, we get a smooth DCT curve. This is important as it also gives some insight into the form of the analytical expression from which these samples have been obtained and thus gives an intuition into the form of the macro-model template. The DCT plots obtained after zero padding, along with analytical expressions derived in Chapter 3, form the basis for inferring the template function discussed in Chapter 4.
There are some other properties of the DCT of the current waveform that make it an attractive choice for macro-model construction. In the next section we discuss these important properties.
2.2 Properties of Current Waveform DCT

Some properties of the DCT of the current waveform relevant to macro-modeling are as follows:

1. **Energy-per-cycle property:** The DC sample, \( I[0] \), of the current waveform DCT corresponds to energy per cycle [27], [28] for the given input vector pair. In fact,

\[
I[0] = \frac{\text{EPC}}{V_{dd} \Delta t \sqrt{N}}
\]  

(2.4)

where EPC is the energy per cycle for the given input vector pair, \( V_{dd} \) is the supply voltage, \( \Delta t \) is the sampling time period, and \( N \) is the length of the zero-padded sequence. 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.

2. **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. The error affects all the time domain samples, but the impact of this error gets distributed over several samples. Specifically, in our case, even though we evaluate a 10 000 point DCT of the current waveform, we can get a reasonable estimate of the time domain current waveform from the first few (100 to 200) dominant terms of the DCT, as shown in Fig. 2.14.

In this plot we show the actual current waveform, and the approximate current waveform obtained by just keeping the first 200 dominant terms of the DCT and taking their inverse. The remaining terms of the DCT are reduced to zero. This is also an important property of the DCT which we use in our macro-modeling technique discussed in Chapter 4.

Because DCT is a discrete transform, it does not give any insight into the form of the analytical expression needed to construct the current macro-model. The only insight into the form of the current macro-model obtained using DCT is through the smooth curves shown in Section 2.1. Therefore, we exploit the relation between DCT and continuous Fourier transform to get an insight into the form that the DCT current macro-model should take. For example, we consider a piecewise-linear triangular waveform, an approximation of
the actual current waveform and construct its Fourier transform as an analytical closed-form expression. The form of this Fourier transform suggests what functional form one should use for the DCT model template. This relationship between DCT and continuous Fourier transform is possible because of Eq. (2.1), as without a corresponding continuous signal for a discrete signal one cannot relate the DCT and continuous Fourier transform. In the next section we examine this relationship between DCT and continuous Fourier transform.

2.3 DCT and Fourier Transform

The DCT of an $N$-sample (here we do not distinguish between the actual current sample length or the zero-padded sequence) sequence is closely related to the DFT, which in turn is a sampled version of the DTFT. The DTFT of a discrete time signal is related to the Fourier transform (FT) of the corresponding continuous signal. The exact relationship among all these transforms will be explained here.

The $N$-sample DCT of a discrete signal is related to a $2N$-sample DFT $[30]$ 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 of the new sequence, and saving $N$ terms of it. The even extension of the discrete time current waveform $i[n]$ is
defined as

\[ i'[n] = \begin{cases} 
  i[n] & n = 0, 1, \ldots, N - 1 \\
  i[2N - 1 - n] & n = N, N + 1, \ldots, 2N - 1 
\end{cases} \quad (2.5) \]

The 2N-point DFT of \( i'[n] \), by definition [30], is given by

\[
I'[k] = \frac{1}{\sqrt{2N}} \sum_{n=0}^{2N-1} i'[n] e^{-j(2k\pi n/2N)}
\]

\[
= \frac{1}{\sqrt{2N}} [i[0] + i[0] e^{-j(2k\pi(2N-1)/2N)} + \ldots + i[N - 1] e^{-j(2k\pi(N-1)/2N)} + i[N - 1] e^{-j(2k\pi(N)/2N)}] 
\]

\[
= \frac{1}{\sqrt{2N}} e^{j(k\pi/2N)} [i[0] e^{-j(k\pi/2N)} + i[0] e^{-j(2k\pi) + j(k\pi)/2N} + \ldots + i[N - 1] e^{-j(2k\pi N/2N) + j(k\pi)/2N} + i[N - 1] e^{-j(2k\pi N/2N) - j(k\pi)/2N}] 
\]

\[
= \sqrt{\frac{2}{N}} e^{j(k\pi/2N)} \sum_{n=0}^{N-1} i[n] \cos \left( \frac{(2n + 1)k\pi}{2N} \right) 
\quad (2.6)
\]

Equation (2.6) holds because \( e^{-j(2k\pi N/2N)} = e^{j(2k\pi N/2N)} \), and this relationship is used in deriving Eq. (2.6) (actually third step). Equation (2.6) shows that 2N-point DFT and DCT are related, except for the extra phase term that appears when we evaluate the DFT of an even extended sequence. They also differ in the scaling of \( I[0] \).

It is also known that the DFT is related to DTFT [29]. 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 2\( \pi \), multiplied by a constant factor to make DFT an orthonormal transform. Thus by definition,

\[
I'[k] = \frac{1}{\sqrt{2N}} I_d(\omega) \bigg|_{\omega=\frac{2\pi k}{2N}} \quad k = 0, 1, \ldots, 2N - 1 
\quad (2.7)
\]

where \( I_d(\omega) \) is the DTFT of the sequence \( \{i'_n\} \) and is defined as

\[
I_d(\omega) = \sum_{n=0}^{2N-1} i'[n] e^{-j\omega n} 
\quad (2.8)
\]

where \( \omega \) 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 FT of the continuous waveform, denoted by \( I_c(\Omega) \), is given by [29]:

\[
I_d(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} I_c \left( \frac{\omega + 2\pi n}{T} \right) 
\quad (2.9)
\]
where $\Omega$ has been replaced by $(\omega + \frac{2\pi n}{T})$. When $n = 0$, which corresponds to a DTFT between $-\pi \leq \omega \leq \pi$, we get $\Omega = \frac{\omega}{T}$ [29].

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 signal from which the given discrete-time samples were taken. The DCT of the current waveform also has some desirable properties, especially the energy compaction property which aid in current macro-modeling. Of course, the most significant feature of the DCT of the current waveform is the fact that the shape of the DCT of a current waveform does not vary significantly, even though the actual current waveform varies. In the next chapter, we present the two typical shapes of the current waveform that normally occur in practice. We approximate these current waveforms with piecewise-linear approximations and exploit the relationship between DCT and Fourier transform to derive analytical expressions for the DCT of these approximations. These analytical expressions along with the smooth DCT plots are used to gain insight into the form of the actual current macro-model template function.
CHAPTER 3

ANALYSIS OF CURRENT WAVEFORMS

In the previous chapter we established a relationship between the DCT of a discrete time signal and the Fourier transform of the corresponding continuous time signal. In this chapter, we exploit this relationship to derive analytical expressions for the Fourier transform of simplified representations of typical current waveforms. The form of these analytical expressions gives insight into the actual DCT model template. This DCT model template is finally used to capture the DCT of the current waveform as a function of the input vector as discussed in Chapter 4. Now, in order to derive the analytical expressions we first identify the various types of current waveform shapes that one commonly sees in practice, and approximate them with regular piecewise-linear waveforms. For such regular waveforms it is possible to derive closed form analytical expressions for the Fourier transform.

A CMOS combinational logic circuit draws current over one cycle in response to a vector pair at its inputs. After simulating various MCNC [31] and ISCAS [32] benchmark circuits with HSPICE, we found that the various current waveform shapes typically fall into one of the three categories:

1. **Approximately triangular**: In this case, the time domain current waveform has a triangular geometry, and the DCT of the current waveform is similar to the DCT of a regular triangular wave.

2. **Approximately trapezoidal**: In this case, the time domain current waveform has a trapezoidal geometry, and the DCT of the current waveform is similar to the DCT of a regular trapezoid.

3. **Multiple peaks**: In this case the time domain current waveform has distinct multiple peaks which are separated in time. We use partitioning in the time-domain to convert such a waveform to a sequence of waveforms that are usually triangular (but can be trapezoidal), and then model them accordingly.

In the following, we explore the FT of the triangular and the trapezoidal current waveforms, which will be used to infer reasonable forms for our DCT model template. In Section 3.3 we present the FT of a simple two peak current waveform and explain why
partitioning is a better option. We also discuss the partitioning algorithm used to partition multiple peak current waveforms, which makes modeling multiple peaks easier.

3.1 Triangular Current Waveform

A typical triangular current waveform and its piecewise-linear triangular approximation are shown in Fig. 3.1. Before taking the DCT, we increase the waveform time duration and assume zero values for the waveform over the time extension. This, as discussed earlier, is called 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. It also gives a lot of insight into the form of the current macro-model. The smooth DCT plots, along with the analytical expressions to be derived in this chapter help us define the generic template function for current macro-model, as explained in Chapter 4. Therefore, 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 10000 sample points in Fig. 3.1.

![Figure 3.1 Typical triangular current waveform and its approximation](image)

As discussed in Chapter 2, the DCT of a discrete time signal is related to the FT of the even extension of the corresponding continuous time signal. In our case we use the FT of the piecewise-linear approximation to infer the DCT model template. Therefore, we construct an even extension of this triangular waveform and derive the corresponding continuous FT, as shown in Fig. 3.2.
The triangular approximation to the current waveform shown in Fig. 3.1 can be analytically expressed as

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

(3.1)

where \(a, b, c,\) and \(d\) are the dimensions shown in Fig. 3.2. The FT of the even extension of \(f_{\text{tri}}(t)\), denoted by \(F_{\text{tri}}(\Omega)\) is given by (very easy to derive)

\[
 F_{\text{tri}}(\Omega) = A e^{-j\Omega(a+b+c+d)} \left[ \frac{2 \cos \Omega(b+c) - 2 \cos \Omega(a+b+c)}{a\Omega^2} + \frac{2 \cos \Omega(b+c) - 2 \cos \Omega(c)}{b\Omega^2} \right]
\]  

(3.2)

where \(\Omega\) is the frequency in the analog (continuous) domain and is related to \(\omega\), the digital frequency by \(\omega = \Omega T\). In our case, we have \(T = 0.01\) ns and \(N = 10\) 000 before the even extension. After extension, we have \(2N\) samples therefore \(\omega = 2\pi k/2N = \pi k/10000\) 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 \(a + b + c + d = 100\) ns. Now using the set of equations described in Chapter 2, we can relate \(F_{\text{tri}}(\Omega)\) to the DCT, of the samples of the piecewise-linear triangular approximation. Comparing the DCT values obtained through \(F_{\text{tri}}(\Omega)\) with the DCT of the actual current waveform, one can verify whether \(F_{\text{tri}}(\Omega)\) can be used to get insight into the actual DCT model template.
The DTFT of the samples of the piecewise-linear triangular waveform, over the interval \([0, \pi]\) is given by

\[
F_{\text{trid}}(\omega) = \frac{1}{T} F_{\text{tri}} \left( \frac{\omega}{T} \right)
\]  

(3.3)

where \(F_{\text{trid}}(\omega)\) is the DTFT of the samples of the even extension of \(f_{\text{tri}}\). Actually, the above equation is just an approximation, because \(F_{\text{tri}}(\Omega)\) is not a bandlimited signal, we would have aliasing [29] from the high frequency components of \(F_{\text{tri}}(\Omega)\). But one can ignore aliasing here because the high frequency terms are decaying with the square of frequency, and the amount of aliasing is negligible as it would verified later when we compare the actual DCT of the samples to the values obtained from \(F_{\text{tri}}(\Omega)\). The DCT of the samples of the piecewise-linear triangular waveform obtained from Eq. (3.3) are given by

\[
F_{\text{tri}}[k] = \frac{1}{T\sqrt{2N}} F_{\text{tri}} \left( \frac{\pi k 10^9}{100} \right) \quad k = 1, 2, N - 1
\]

\[
= \frac{1}{2T\sqrt{N}} F_{\text{tri}} \left( \frac{\pi k 10^9}{100} \right) \quad k = 0
\]  

(3.4)

The phase term has been ignored in Eq. (3.4) because it is subsumed when we go from continuous FT to DTFT and DFT. In Eq. (3.4), the DCT sample corresponding to \(k = 0\) is obtained by taking the limit, as shown below:

\[
\lim_{k \to 0} \frac{1}{2T\sqrt{N}} F_{\text{tri}} \left( \frac{\pi k 10^9}{100} \right)
\]  

(3.5)

Basically, the DCT samples for the piecewise-linear waveform can be obtained from the corresponding continuous transform \(F_{\text{tri}}\), if we ignore aliasing. In Fig. 3.3, we show the DCT of the current waveform, the DCT of the triangular waveform approximation, and the plot of the DCT samples obtained from continuous transform. The plot shows that the DCT of the triangular waveform approximation and the DCT plot of the current waveform are very similar. The DCT plot of the current waveform has a very regular shape (as discussed in Chapter 2), making it a good candidate for model construction, which is one reason why we chose the DCT over other transforms. Compared to DFT (discrete Fourier transform) DCT has only real terms, which also makes the model simpler, because with DFT one has to estimate both the real and complex values of the DFT samples.

Figure 3.3 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. Thus, the current macro-model is an analytical expression that captures the DCT samples of the current waveform as a function of the input vector pair. Now, in order to infer the form of this template, we compare the plot of DCT samples obtained from continuous transform to the DCT plot of the current waveform and the DCT plot of the piecewise-linear triangular approximation. The plot shows that the samples obtained from the continuous transform of the piecewise-linear approximation and its corresponding DCT are almost identical; therefore, one can ignore aliasing. Compared to the DCT of the actual current waveform, it is obvious that except for the scale factor, plot of the samples obtained from Eq. (3.4) have the right shape and thus may help us define the form of the current macro-model template function.

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 period, the plots in Fig. 3.3 show that in case of a current waveform and its approximations (both discrete and continuous transform plots) the period is varying (we can use the difference between consecutive/alternate zero crossings or the difference between consecutive maxima or minima to measure the 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 period, as will be discussed in Chapter 4. The fact that the DCT sample for a piecewise-linear approximation decays with the square of the frequency is an important observation. This turns out to be very useful while defining the generic current macro-model for the triangular current waveform, as will be discussed in Chapter 4.

### 3.2 Trapezoidal Current Waveform

In some cases, the current waveform has trapezoidal shape, as shown in Fig. 3.4, which also shows a piecewise linear trapezoidal approximation to the current waveform. In Fig. 3.4 we again show a zero padded current waveform for obvious reasons. It is our observation that the trapezoidal current waveform usually occurs for low Hamming distance input vector pair, for some circuits only. Low Hamming distance means that less than 60% of the primary inputs are switching. Triangular current waveform occurs most frequently for the majority of benchmark circuits as shown in Chapter 5. The equation of the trapezoidal wave shown in Fig. 3.4 is given by:

$$f_{\text{tra}}(t) = \begin{cases} 
0, & \text{for } 0 \leq t \leq a; \\
A(t-a)/b, & \text{for } a \leq t \leq a+b; \\
A, & \text{for } a+b \leq t \leq a+b+c; \\
A(a+b+c+d-t)/d, & \text{for } a+b+c \leq t \leq a+b+c+d; \\
0, & \text{for } a+b+c+d \leq t \leq a+b+c+d+e.
\end{cases} \quad (3.6)$$

where the dimensions are illustrated in Fig. 3.5, which also shows an even extension of the piecewise-linear trapezoidal approximation.

The FT of the even extension of $f_{\text{tra}}(t)$, denoted by $F_{\text{tra}}(\Omega)$, is given by

$$F_{\text{tra}}(\Omega) = Ae^{j\Omega(a+b+c+d+e)} \left[ \frac{2 \cos \Omega(c+d+e) - 2 \cos \Omega(b+c+d+e)}{b\Omega^2} \right] + Ae^{j\Omega(a+b+c+d+e)} \left[ \frac{2 \cos \Omega(e+d) - 2 \cos \Omega(e)}{d\Omega^2} \right] \quad (3.7)$$

24
Figure 3.4 Typical trapezoidal current waveform and its approximation

Figure 3.5 Diagram explaining $f_{\text{tra}}(t)$

where $\Omega$ is the frequency in the analog (continuous) domain and is related to $\omega$, as discussed in Section 3.1. As before, we can obtain the DCT samples for the piecewise-linear trapezoidal waveform from $F_{\text{tra}}$, using the following equation:

$$F_{\text{tra}}[k] = \begin{cases} \frac{1}{T\sqrt{2N}}F_{\text{tra}} \left( \frac{\pi k 10^9}{100} \right) & k = 1, 2, N - 1 \\ \frac{1}{2T\sqrt{N}}F_{\text{tra}} \left( \frac{\pi k 10^9}{100} \right) & k = 0 \end{cases}$$

(3.8)
In Eq. (3.8), the DCT sample corresponding to $k = 0$ is obtained by taking the limit, as shown below:

$$\lim_{k \to 0} \frac{1}{2T\sqrt{N}} F_{\text{tra}} \left( \frac{\pi k 10^0}{100} \right)$$

(3.9)

The DCT samples, obtained from Eq. (3.7) are again real for the values of $\Omega$ under consideration. In Fig. 3.6, we show the DCT of the current waveform, the DCT of the trapezoidal waveform approximation, and the plot of the continuous transform. Again it can be seen that the DCT of the actual current waveform compares well with the DCT of the piecewise-linear approximation. Except for a scale factor and different zero-crossing points, the two DCT plots match very well. It can also be seen that the DCT samples obtained from the Fourier transform of the piecewise-linear approximation and its actual DCT are almost identical, thus one can ignore aliasing in this case also. This implies that the continuous FT $F_{\text{tra}}$ and the actual current waveform DCT have very similar behavior at the sample frequencies under consideration. Thus, one can use the continuous transform analytical expression to infer the form of the DCT template for the actual current waveform.

![Figure 3.6 DCT of a typical trapezoidal waveform](image)
The most obvious inference that one can draw from the continuous transform, is the quadratic decay in the amplitude of the DCT, in terms of frequency (similar to that for the triangular current waveform). If we compare the plots in Fig. 3.3 and Fig. 3.6, 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 period to simplify our model, as we do in the case of triangular waveforms (see Chapter 4). 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 is also based on the observation of the plots. Figure 3.6, 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 period. In the case of trapezoidal waveforms, we use the difference between maxima (or minima) as the 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 Chapter 4.

3.3 Multiple Peak Current Waveform

When we have current waveforms with multiple peaks that 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. 3.7, we show a typical multiple peak current waveform with two distinct peaks which are far apart.

The current waveform shown in Fig. 3.7 can obviously be considered a summation of two triangular waves. The DCT of the multiple peak current waveform obtained after zero padding is shown in Fig. 3.8. Again here we show the first 100 points of the DCT for clarity.

In Fig. 3.8, we can see that the DCT is distorted. Therefore, the templates, discussed in Sections 3.1 and 3.2 cannot be applied directly as they cannot capture the distortions in the DCT of the current waveform. Since these distortions in the DCT values are occurring at the dominant frequencies, it is important to capture these variations in the DCT values, to get
Figure 3.7 Typical multiple peak current waveform

Figure 3.8 DCT of a multiple peak current waveform

A good estimate of the current waveform. One solution to the problem of modeling multiple peak current waveforms is to again exploit the relationship between DCT and continuous Fourier transform, to get an insight into the form of the DCT model template required.
for multiple peak waveforms. In this context, we can consider the simplest multiple peak current waveform which is a two peak current waveform and construct a piecewise-linear approximation to it, as shown in Fig. 3.9. In Fig. 3.9, we also show an even extension to the piecewise-linear approximation. The two peak piecewise-linear approximation to the multiple peak current waveform can be analytically expressed as

\[
 f_{\text{mul}}(t) = \begin{cases} 
 0, & \text{for } 0 \leq t \leq f; \\
 A(t - f)/a, & \text{for } f \leq t \leq a + f; \\
 A(a + b + f - t)/b, & \text{for } a + f \leq t \leq a + b + f; \\
 B(t - (a + b + f))/c, & \text{for } a + b + f \leq t \leq a + b + c + f; \\
 B(a + b + c + d + f - t)/d, & \text{for } a + b + c + f \leq t \leq a + b + c + d + f; \\
 0, & \text{for } a + b + c + d + f \leq t \leq a + b + c + d + f + e.
\end{cases}
\] (3.10)

where \(a, b, c, d, e,\) and \(f\) are the dimensions shown in Fig. 3.9, and \(A\) and \(B\) are the peak values of the two peak piecewise-linear approximation.

![Figure 3.9 Piecewise-linear approximation to two peak current waveform](image)

The FT of the even extension of \(f_{\text{mul}}(t)\), denoted by \(F_{\text{mul}}(\Omega)\) is given by

\[
 F_{\text{mul}}(\Omega) = B e^{-\jmath \Omega (a+b+c+d+e+f)} \left[ \frac{2 \cos \Omega (d+e) - 2 \cos \Omega (c+d+e)}{c \Omega^2} \right] + B e^{-\jmath \Omega (a+b+c+d+e+f)} \left[ \frac{2 \cos \Omega (d+e) - 2 \cos \Omega (e)}{d \Omega^2} \right]
\]

29
\[ + Ae^{-j\Omega(b+c+d+e+f)} \left[ \frac{2\cos\Omega(b + c + d + e) - 2\cos\Omega(a + b + c + d + e)}{a\Omega^2} \right] \\
+ Ae^{-j\Omega(b+c+d+e+f)} \left[ \frac{2\cos\Omega(b + c + d + e) - 2\cos\Omega(c + d + e)}{b\Omega^2} \right] \] (3.11)

The above equation confirms the fact that the DCT of a two-peak waveform is obtained from the summation of the DCTs of two time shifted single peak waveforms. The above equation can be generalized for more than two peaks easily, but the problem with Eq. (3.10) is that it has a lot of terms which make the task of inferring a simple DCT model template really difficult. One can also use the above equation as it is and try to estimate the various parameters of the template using regression as in the case of the other two templates (discussed in Chapter 4). Estimating the parameters of a template as described in (3.11) using regression is very difficult computationally as it involves nonlinear parameter estimation. There are also accuracy issues involved while using such a model to predict current waveforms for various input vector pairs. Usually the basic philosophy of regression is to keep the model simple [33], and the above equation from a regression perspective is not simple. Another issue while using such a template is that all the current waveforms for a given Hamming distance and for a given circuit are not multiple peak, which makes the above template even difficult to use. In general, there may be more than two peaks, and we might need a different template function to model each such cases depending on the number of peaks. This would make the model construction cumbersome as one would have to chose from several template functions before arriving at the correct template. Therefore, we propose partitioning the time domain current waveform with multiple peaks. Partitioning converts the multiple peak time domain current waveform into a sequence of waveforms that are either approximately triangular or approximately trapezoidal. From a practical point view, we have not come across multiple peak current waveforms which are trapezoidal, so we will not discuss them in detail in this dissertation, but the techniques described in this dissertation can be applied to multiple peak trapezoidal current waveforms too.

In the proposed partitioning technique we exploit the fact that DCT is a linear transform. Therefore, DCT of a multiple peak current waveform is the summation of the DCT of individual time-shifted single peak waveforms. Thus, one can partition the time domain waveform into subintervals so that each subinterval contains a single peak, and then build a model for each of the time shifted waveforms using the templates discussed in Section 3.1.
(or Section 3.2). The flow chart for the partitioning algorithm is shown in Fig. 3.10, and the steps of the algorithm 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 and therefore can be done in $O(n)$. 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., $i[n] \leq 0.01i_{pg}$ and note the index $n$. Set $p_{index} = n$. This is a partition. This also is $O(n)$.

**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. This is also $O(n)$.

**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 $i_p \geq 0.15i_{pg}$, go to step 5, else stop (partitioning is not required).

**Step 5:** Set index $= index + 1$. Go to Step 2. The whole algorithm is repeated until there are significant peaks. If $P_n$ is the number of significant peaks, the partitioning algorithm is of the order $O(P_n n)$.

With the above partitioning algorithm, 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. 3.7 was partitioned using the above algorithm, and the DCT of the resulting partitions is shown in Fig. 3.11. We can see from Fig. 3.11 that the DCT of partitioned current waveforms does not show any obvious distortion and appears to be a decaying sinusoid. Therefore one can use the triangular DCT template (discussed in Section 3.1) to model each partition of the two peak current waveform. The partitions obtained from the above algorithm can be treated as the time shifted waveforms which constitute the multiple peak current waveform. It is also obvious from Fig. 3.11 that the DCT of the time-shifted/partitioned current waveforms interfere among each other, and thus cause distortions in the resulting DCT of the multiple peak current waveform. The DCT of the multiple peak current waveform is a linear combination of the DCTs of the time-shifted/partitioned current waveforms.

Basically with time-domain partitioning we try to identify the various time-shifted single peak waveforms, which actually constitute the multiple peak current waveform, and model the DCT of the constituent elements. This approach is better than creating a new template function because with partitioning we keep the number of current macro-model templates to only two, instead of introducing a multitude of templates to model the various multiple peak current waveforms. Since in the current macro-model characterization phase one needs to automatically detect the appropriate template depending on an error criterion (see Chapter 4), it makes sense to minimize the number of templates and reduce the characterization time spent in identifying the appropriate template, through various metrics (see Chapter 4).

In summary, we need to infer appropriate DCT templates for the triangular and trape-
**Figure 3.11** DCT of a multiple peak current waveform after partitioning.

*Zoidal* current waveforms only based on the analytical expression derived in Sections 3.1 and 3.2. The DCT template function for these two templates can also be used with multiple peak current waveforms after time-domain partitioning. In the next chapter we present the actual DCT template functions for the two typical current waveform shapes identified in this chapter and also describe the model construction process, which basically includes estimating the parameters of the DCT template function.
CHAPTER 4
MACRO-MODEL CONSTRUCTION

In the previous two chapters we presented the frequency transforms that enable current macro-modeling and illustrated the typical shapes these transforms take in practice. We also derived analytical expressions for these typical transform shapes by approximating them with piecewise-linear triangular and trapezoidal waveforms. These analytical expressions form the basis for inferring the DCT templates of actual current waveforms. In this chapter we present the DCT template functions for the actual current waveforms and a methodology to estimate the parameters of these templates as a function of the input vector pair using regression. We also discuss an integrated macro-model development flow that can be used to construct current macro-models for logic blocks without any user intervention.

The primary objective of current macro-modeling is to develop a compact analytical expression for the DCT samples of the current waveform as a function of some independent and easy to evaluate variables. In this context we need to identify the various variables that determine the current drawn by a logic block from the power supply. The primary constraint on the selection of these variables is the fact that they should be easy to evaluate for the model to be useful. Now the current drawn by a logic block from the supply rails is primarily a function of the transitions occurring at its primary inputs. Therefore, we use the input vector pair as an independent variable in our current macro-model. There are other secondary variables like the slew rate of the transitions occurring at the primary inputs, the output load capacitance driven by the logic block, different arrival times of the signals at the primary inputs. Some of these variables are very difficult to determine at a higher level of abstraction and some can be accounted for using various techniques. Some of these techniques will be discussed in Chapter 6. Since the proposed current macro-modeling technique is used to build current models for existing implementations of the circuit only, we ignore any circuit level detail while identifying the variables. Otherwise the current drawn by a logic block is a function of the individual nodes switching inside the circuit in response to transitions at the primary inputs. Since the switching of the internal nodes is also a function of the transitions at the primary inputs, we use the primary input vector pair as
the only independent variable. The current signature specific to the circuit (which is due to the transitions of the internal nodes) is captured during the model characterization process. Thus, once the model is generated, current waveforms can be obtained from the model by just evaluating the primary inputs of the logic block. This can be achieved easily with a behavioral simulation of the design where the logic block is instantiated.

The model characterization process involves the estimation of the DCT template parameters as a function of the input vector. The estimation of the parameters of the DCT template function is done using regression and it requires the following steps:

**Step 1:** Identifying the proper functions to model the parameters of a given DCT template.

**Step 2:** Mapping a given input vector pair to a set of variables which is amenable to regression, because we cannot use the input vector pair directly as an independent variable in an analytical expression. Therefore, we need a mapping from the input vector pair to a set of variables that become the independent variables of the functions identified in step 1.

In the next section, we present such a mapping of input vector pair to set of variables using *dummy coding* which enables estimating the parameters of the DCT template as a function of the input vector pair. In Sections 4.2 and 4.3 we present the DCT templates for triangular and trapezoidal current waveforms. Finally, in Section 4.4 we present the actual characterization flow which enables the construction of the current macro-model for a given logic block automatically.

### 4.1 Dummy Coding

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 \( \{0 \rightarrow 1, 1 \rightarrow 0, 1 \rightarrow 0, 0 \rightarrow 0\} \). We treat the set of four possible transitions at
each primary input as a categorical variable [34]. 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* and the new variables are called *dummy variables* or *indicator variables*. In our case, the transition at a given input node can take four possible values, therefore we need three variables. The three variables denoted by \( x_1, x_2 \) and \( x_3 \) are used to capture the four possible transitions that can occur at a given input node. The dummy coding for the 4 levels of our categorical variable is shown in Table 4.1 and is very similar to that used in [27].

**Table 4.1** Dummy coding for transitions at a single node

<table>
<thead>
<tr>
<th>Input Transition</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0 \rightarrow 1 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( 1 \rightarrow 0 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( 1 \rightarrow 1 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( 0 \rightarrow 0 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In Table 4.1, each input transition is mapped to a vector of \([x_1 \; x_2 \; x_3]\). The reason we need such a mapping is that it prevents any inherent bias among the categorical variables and their impact on the dependent variable, as discussed in [34]. The mapping shown in Table 4.1, 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. In spite of all the above advantages of using dummy variables, the above mapping has a clear drawback that it increases the number of variables as well as the number of regression coefficients. But we have found that, depending on the fraction of primary inputs switching, we can introduce a less expensive solution, as follows. For the case where more than 60% of the inputs are switching (which we refer to as *large Hamming distance*) most of the inputs undergo \( 0 \rightarrow 1 \) or \( 1 \rightarrow 0 \) transition. Therefore, we can assume that that the categorical variable has just two levels, so that a single variable is required to represent the transitions at a primary input. It should be noted that for small Hamming distance (a small fraction of
primary inputs switching) very few inputs undergo $0 \rightarrow 1$ or $1 \rightarrow 0$, but we still cannot use any approximation, because each of those transitions is very significant. The current drawn from the supply in case of low Hamming distance vector pairs is very sensitive (actually, it is due to the small fraction of inputs switching) to the small fraction of inputs switching. The mapping for large Hamming distance is shown in Table 4.2, where each transition is mapped to a single variable denoted by $y$. The mapping shown Table 4.2, has all the drawbacks we mentioned above. It imposes an assumed order on the impact these variables would have on the current waveform. For example, a $0 \rightarrow 0$ transition impacts the current waveform four times more than a $0 \rightarrow 1$ transition. But in the case of large Hamming distance, we can live with this fact because most of the transitions are either $0 \rightarrow 1$ or $1 \rightarrow 0$. Since the coefficients are estimated through random vector generation, the impact of this bias is significantly reduced.

**Table 4.2 Coding for large Hamming distances**

<table>
<thead>
<tr>
<th>Input Transition</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0 \rightarrow 1$</td>
<td>1</td>
</tr>
<tr>
<td>$1 \rightarrow 0$</td>
<td>2</td>
</tr>
<tr>
<td>$1 \rightarrow 1$</td>
<td>3</td>
</tr>
<tr>
<td>$0 \rightarrow 0$</td>
<td>4</td>
</tr>
</tbody>
</table>

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. 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 coefficients 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 [35], [36].

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 five 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 the 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 $\lfloor 0.1p \rfloor$, and the same holds for rest of the partitions.
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 in Chapter 5 that 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. 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.

4.2 Triangular Template

The triangular template is a simplified as well as a modified version of \( F_{\text{tri}}(\Omega) \) (refer to Eq. (3.2), page 22). The simplification is achieved because we explicitly incorporate the effect of the changing time period as discussed in Chapter 3. We estimate the first few time periods (in our case, five) directly (see Section 4.2.3) and use them in a single cosine term, whose other parameters do not change. This is possible because of the energy compaction property discussed in Chapter 2. Therefore, one needs to estimate the first few (100 to 200 samples) dominant samples to get a reasonable estimate of the current waveform as shown in Chapter 2. The template equation for the DCT of the triangular current waveform is therefore given by

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

(4.1)

where \( k \) is the sample index. In order to completely specify the above template function, we need to define the following four parameters as a function of the variables associated with the input vector pair:

- **Decay factor** (\( D(k) \)): This is motivated by the continuous FT of the piecewise-linear triangular waveform \( F_{\text{tri}} \) discussed in Chapter 3. Since DCT samples obtained from \( F_{\text{tri}} \) decay with the square of frequency, we include a \( 1/k^2 \) dependence in this function, among other things, as explained in Section 4.2.1.

- **Amplitude** (\( A \)): The first largest sample of the DCT of the current waveform is taken as the amplitude, and then all the sample points are measured with respect to this reference. Usually, \( I[1] \), corresponds to the amplitude of \( I_{\text{tri}}(k) \). The polynomial expression used to model the amplitude is discussed in Section 4.2.2.

- **DC value** (\( I(0) \)): This is the first sample of the current waveform DCT and is not included in the template function Eq. (4.1), discussed earlier. Estimating this term separately simplifies the template function \( I_{\text{tri}}(k) \) as discussed in Section 4.2.2.
**Period** ($T_i$): It is the variable period of the decaying sinusoid which is a characteristic of the DCT of the triangular current waveform as discussed in Chapter 3. In our approach we estimate the first few periods, as explained in Section 4.2.3.

### 4.2.1 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 $F_{tri}$:

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

(4.2)

where $x$ is either $y$ (a vector of length one, refer to Table 4.2) or the vector $[x_1 \ x_2 \ x_3]$ (refer to Table 4.1), 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. The three polynomial functions $f(x)$, $g(x)$, and $h(x)$ have the same form:

$$\prod_{i=1}^{p} (\alpha_i y_i + \beta_i), \text{ for large Hamming distance}$$

(4.3)

$$\prod_{i=1}^{p} (\alpha_{i1} x_{1,i} + \alpha_{i2} x_{2,i} + \alpha_{i3} x_{3,i} + \beta_i), \text{ for small Hamming distance}$$

(4.4)

where $p$ is the number of primary inputs. With the above polynomial functions, one can account for possible interaction between the input variables, through cross-product terms, without significantly increasing the number of coefficients compared to a linear model. The coefficients of these polynomials are obtained by substituting these polynomial functions into Eq. (4.2) and using nonlinear 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. 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 consider the maxima and minima only in the first five cycles during regression. Some typical decay factor curves and their corresponding estimates obtained using our model are shown in Fig. 4.1. These were obtained from the alu2 MCNC benchmark circuit [31].
Figure 4.1 Decay factor and its estimate for five different current waveforms

4.2.2 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 pairs. With partitioning, we can model both $A$ and $I(0)$ along the lines of the polynomial functions discussed in Section 4.2.1:

$$\Pi_{i=1}^p (\alpha_i x_{1,i} + \beta_i), \text{ for large Hamming distance}$$  \hspace{1cm} (4.5)

$$\Pi_{i=1}^p (\alpha_{i1} x_{1,i} + \alpha_{i2} x_{2,i} + \alpha_{i3} x_{3,i} + \beta_i), \text{ for small Hamming distance}$$  \hspace{1cm} (4.6)

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 4.1 and 4.2, respectively. 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 usually, 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 both $A$ 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 Eq. (4.5) and Eq. (4.6).

![Graph](image)

**Figure 4.2** Actual amplitude vs. estimated amplitude, for alu2

Estimating the DC value separately, simplifies the DCT template function for the current waveform $I_{\text{tri}}$. Otherwise, for $k = 0$ the template function would have required a phase angle $\phi$ as a function of the input vector pair. This would have required another non-linear regression to estimate $\phi$, which is computationally more expensive than simple linear regression. Moreover the analysis in Chapter 3 shows that the DC value corresponds to $k \to 0$, and it is difficult to model numerically within an analytical expression. A typical plot of actual DCT amplitude versus estimated DCT amplitude is shown in Fig 4.2 and a similar plot for DC value is shown in Fig. 4.3. As discussed in Chapter 2, $I[0]$ corresponds to the *energy per cycle* for a given input vector pair, and it is a byproduct of our current macro-modeling methodology.
4.2.3 Period

The variable 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 periods of the DCT template, and this is denoted by \( T_i, \ i \in \{1,2,3,4,5\} \). The choice of just five periods is empirical and gives good accuracy because of the energy compaction property of the DCT discussed in Section 2.1. But, if one attempts to build a single comprehensive model for each of these periods across all the possible input vector pairs the 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 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 period. Therefore for every partition, we compute during characterization a nominal value for each of the first five periods. This is selected as the peak value of the distribution of observed periods for that partition. In statistical terms, we choose the mode of the period distribution as the
period for a given partition of the input vector pairs. The choice of the first five periods is not a limitation of the model and can be increased if there is a need. A typical histogram of the first period for alu2 is shown in Fig. 4.4, and it can be seen that there is no significant variation in the 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 it would be obvious after we discuss the trapezoidal template.

Figure 4.4 Distribution of the first period for large Hamming distance, alu2

4.3 Trapezoidal Template

A trapezoidal template model seems to best fit the current waveforms for low Hamming distances, when the fraction of primary inputs switching is \( \leq 60\% \). Our trapezoidal template is given by

\[
I_{\text{tra}}(k) = A \left[ D_1(k) \cos\left(\frac{2\pi(k - 1)}{T_i}\right) + D_2(k) \cos(2\pi(k - 1)\omega) \right], \quad k = 1, 2, \ldots \tag{4.7}
\]

where \( k \) is again the sample index, \( T_i, i \in \{1, 2, 3, 4, 5\} \), is the variable period, \( A \) is the amplitude, \( \omega \) is the frequency of the second cosine term, and \( D_1(k) \) and \( D_2(k) \) are the decay factors corresponding to the cosine terms. Thus, compared to the triangular template, the trapezoidal template has more parameters. The DCT of a trapezoidal current waveform deviates from the simple decaying sinusoid which we observe for the triangular current waveform, as shown in Chapter 3. Therefore, we introduce a second cosine term in the trapezoidal template, as a single cosine term cannot capture this deviation. The second
cosine term introduces extra parameters to the trapezoidal template. The parameters of the trapezoidal template are motivated by the continuous Fourier transform of the piecewise-linear trapezoidal waveform $F_{\text{tra}}$ discussed in Chapter 3 and some empirical observations. These observations help in simplifying the template function to just two cosine terms. The simplification is achieved by using the same concept of variable period $T_i$ as discussed in Section 4.2.

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

Amplitude ($A$): This is the largest sample of the current waveform DCT and corresponds to $I[1]$ as it is obvious from Eq. (4.7). This is estimated separately using linear regression and is motivated by the term $A$ which appears in $F_{\text{tra}}$ (refer to Eq. (3.7)). It is given by

$$A = w(x)$$

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 the inverse square dependence on frequency seen in $F_{\text{tra}}$ (refer to Eq. (3.7)):

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

where $p(x), q(x), r(x),$ and $s(x)$ are also polynomial functions of the variables associated with the input vector pair. The frequency terms in the numerator of $D_1$ and $D_2$ ($p(x)k$ and $q(x)k$) are needed to maintain a good fit between the DCT samples obtained from HSPICE simulations and the DCT model template. The need for such empirical modifications arise because we are using just two cosine terms to model the DCT of the current waveform, whereas the continuous Fourier transform expression $F_{\text{tra}}(\Omega)$ for a trapezoid waveform suggests that we might need four cosine terms. Since four cosine terms may be too many (computationally expensive model as more parameters would have to estimated) and since a single cosine term cannot capture the deviation from a simple decaying sinusoid, we used only two cosine terms in our template and tried to obtain a good fit with some empirical modifications.

Period, frequency ($T_i, \omega$): 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 period $T_i$ is used with one of the two cosine terms, the one which is aimed at generating the decaying sinusoid. The frequency $\omega$ of the other cosine term is modeled as a polynomial function of the variables associated with the input vector pair. The variable 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 $F_{\text{tra}}$. The variable 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 period does not vary much for a given input vector pair partition, as shown in Fig. 4.5. Thus, we use the same method as described in Section 4.2.3 to get an estimate of the first five periods. Basically, from the distribution of periods for a given input vector partition, we choose the period value that occurs most often (in statistical terms, we choose the mode of the period distribution as the period for a given input vector pair partition). The first five 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

$$\omega = t(x)$$  \hspace{1cm} (4.10)

![Figure 4.5 Distribution of the first period for alu2, 40% to 60% inputs switching](image)

Figure 4.5 Distribution of the first period for alu2, 40% to 60% inputs switching
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 4.2.2 and is given by \(I[0] = u(x)\), 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.

Thus all the parameters of the trapezoidal template are modeled as a function of the variables associated with the input vector pair. Since the trapezoidal model is usually used with low Hamming distances, \(x\) corresponds to the vector \([x1 x2 x3]\) 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;

\[
\Pi_{i=1}^{p}(\alpha_{i1}x1 + \beta_{i2}x2 + \gamma_{i3}x3 + \delta_{i})
\]  \hspace{1cm} (4.11)

where \(p\) is the number of primary inputs. A polynomial of this form is substituted in the corresponding parameter definitions. These parameters are then substituted in the DCT template (except for \(A\) and \(I[0]\)), and the coefficients of the polynomials are estimated simultaneously using nonlinear 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 nonlinear regression. This technique simplifies the nonlinear regression, and reduces the number of parameters to be estimated using nonlinear regression, which is computationally expensive. 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 do the following:

1. Determine the sample size on the fly during the characterization phase. 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 from triangular, trapezoidal template or even decide about partitioning the time domain current waveform.

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.

### 4.4 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 SPICE description of the circuit and HSPICE as the circuit simulator. In order to automate the model development flow, we need to decide 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 that 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 [33], denoted by $R^2$ and given by

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

where $i(l)$ is the actual current waveform sample, $\bar{i}$ is the mean of the samples of the actual current waveform, $\hat{i}(l)$ 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 one the more accurate the model is. Since it is very difficult to achieve a value closer to one, 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 chose between the two templates or decide upon partitioning. The steps of the automatic current characterization flow based on this metric is give 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 is 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 determined by 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 4.1. Since the model estimates the DCT of the current waveforms in the sample set $S$, get the corresponding time-domain current waveform samples, $\hat{I}(l)$ for the sample set $S$ using the model.

**Step 3:** Evaluate $R^2$ for each current sample and get the average $R^2$, denoted by $R_{\text{avg}}^2$. We call it the average coefficient of determination and it is given by:

$$R_{\text{avg}}^2 = \frac{1}{|S|} \sum_{m=1}^{S} R^2(m) \quad (4.13)$$

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

**Step 4:** If $R_{\text{avg}}^2 > 0.75$, then the template is correct and use *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 vector pairs, 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 $S$ as the trapezoidal template has more coefficients.

**Step 6:** Build the model again depending on the decision made in step 5. Get the corresponding $R_{\text{avg}}^2$. If, for large Hamming distance, $R_{\text{avg}}^2$ does not improve (it should not
happen), 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. Now if the model is for small Hamming distance go to step 7, else go to step 8.

**Step 7**: If, for small Hamming distance, $R_{\text{avg}}^2$ 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 $R_{\text{avg}}^2$. It should improve. If it does not (which, again, should not happen), 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.

The above steps are based on our current macro-modeling experience, and the objective is to get the correct template the first time in a majority of cases. Since the triangular template occurs most of the time, we use it first. The threshold value for $R_{\text{avg}}^2$ 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 (see Chapter 5). A flow-chart depicting the above characterization flow is shown in Fig. 4.6. In the next section we present a simple algorithm to determine an adequate sample size.

### 4.4.1 Sample size

Once the correct template is known, we need to know the size of the sample set for characterization. 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. If the characterization sample set adequately represents the entire set (the entire set of input vector pairs for a given partition would be referred to as the population, a term borrowed from statistics), or at least the 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. Note 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 some quantitative metrics.
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 4.4.

**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. The model is tested on a different sample set because if one starts with a polynomial function whose number of coefficients is equal to the size of data, one usually gets a very good fit (actually exact fit in case of linear regression) and thus the error metrics derived for such a case can be misleading, as far the prediction of current waveforms is concerned. Therefore, the model is tested on a different sample set by estimating the current waveforms of the set $T$ and comparing them with the actual current waveforms. Comparison is done using three metrics:

1. **Average relative error in peak current estimation** $e_{pavg}$.
2. **Average relative error in the time instant at which the peak occurs** $e_{tavg}$.
3. **Coefficient of determination** $R^2_{avg}$.

These errors metrics are easy to compute and give a good estimate of the accuracy of the model. The relative errors have been described in Chapter 5. The comparison on a test set $T$ is based on the philosophy discussed earlier, i.e., the error metric would improve if the characterization set size increases. 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 the user defined threshold. 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 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:

1. **Calculate the number of Hamming distances** covered in each input vector pair partition, denoted by $H_{num}$.
2. 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 nonlinear
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 do not tamper with the test set $T$ (basically the model is tested on the same set each time). After building the model go to step 3.

![Flow chart for sample size determination](image)

**Figure 4.7** 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. 4.7 we present a flow chart for the sample size determination algorithm.

Thus, in this chapter we presented the two DCT model templates for current macro-
modeling, along with the model construction process. The model construction process includes, estimating the parameters of the DCT template model using regression. We also presented a model development flow to construct current macro-models for logic blocks without any user intervention. These black-box current models enable current waveform estimation as a function of the variables associated with the input vector pair and can be used for high-level current estimation. In the next chapter, we present the experimental results that illustrate the validity of these black-box models.
CHAPTER 5

EXPERIMENTAL RESULTS

In the previous chapter, we described the DCT model template for current macro-modeling along with the model construction process. In this chapter we present some experimental results that illustrate the validity of our approach. In order to test the accuracy of our macro-model, a set of randomly generated vectors were used to simulate various MCNC [31] and ISCAS-85 [32] benchmark circuits in HSPICE. The circuits are shown in Table 5.1, where we have listed the number of inputs, outputs, gates 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 [5], or gate-level simulation techniques such as [17]. The resulting current waveforms were used to construct the current macro-model for the respective circuits using the approach discussed in Chapter 4. In Table 5.1, the last column shows the total number of simulations (number of vector pairs) needed to build the macro-model for all possible Hamming distances, for each circuit. It can be seen that for circuits with large input count, we need many simulations. This can at first be surprising, but it is actually not totally unexpected given the large number of possible Boolean vector pairs with even a moderate number of inputs.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>#I</th>
<th>#O</th>
<th>#Comp</th>
<th>RMS$_{avg}$</th>
<th>$e_{pavg}$</th>
<th>$e_{tavg}$</th>
<th>$R^2_{avg}$</th>
<th>#of simulations</th>
</tr>
</thead>
<tbody>
<tr>
<td>c880</td>
<td>60</td>
<td>26</td>
<td>383</td>
<td>0.0045</td>
<td>17.23%</td>
<td>10.20%</td>
<td>0.8047</td>
<td>3000</td>
</tr>
<tr>
<td>alu2</td>
<td>10</td>
<td>4</td>
<td>368</td>
<td>0.0037</td>
<td>19.20%</td>
<td>8.13%</td>
<td>0.7721</td>
<td>650</td>
</tr>
<tr>
<td>c432</td>
<td>36</td>
<td>7</td>
<td>217</td>
<td>0.0028</td>
<td>12.43%</td>
<td>11.46%</td>
<td>0.8117</td>
<td>2300</td>
</tr>
<tr>
<td>cu</td>
<td>14</td>
<td>11</td>
<td>48</td>
<td>5.4423e-04</td>
<td>18.38%</td>
<td>2.32%</td>
<td>0.8674</td>
<td>550</td>
</tr>
<tr>
<td>f51ml</td>
<td>8</td>
<td>3</td>
<td>105</td>
<td>0.0011</td>
<td>19.21%</td>
<td>2.12%</td>
<td>0.7918</td>
<td>320</td>
</tr>
<tr>
<td>mux</td>
<td>21</td>
<td>1</td>
<td>91</td>
<td>4.4271e-04</td>
<td>17.95%</td>
<td>4.23%</td>
<td>0.8845</td>
<td>800</td>
</tr>
<tr>
<td>random8</td>
<td>8</td>
<td>1</td>
<td>158</td>
<td>9.3857e-04</td>
<td>12.76%</td>
<td>4.38%</td>
<td>0.8391</td>
<td>1000</td>
</tr>
<tr>
<td>parity</td>
<td>16</td>
<td>1</td>
<td>68</td>
<td>7.7149e-04</td>
<td>15.37%</td>
<td>3.98%</td>
<td>0.9034</td>
<td>640</td>
</tr>
<tr>
<td>vdao</td>
<td>17</td>
<td>27</td>
<td>341</td>
<td>0.0049</td>
<td>16.22%</td>
<td>8.89%</td>
<td>0.7840</td>
<td>900</td>
</tr>
<tr>
<td>c499</td>
<td>41</td>
<td>32</td>
<td>202</td>
<td>0.0055</td>
<td>13.28%</td>
<td>7.69%</td>
<td>0.8279</td>
<td>1500</td>
</tr>
<tr>
<td>pcler8</td>
<td>27</td>
<td>17</td>
<td>101</td>
<td>9.8475e-04</td>
<td>8.47%</td>
<td>5.21%</td>
<td>0.9021</td>
<td>900</td>
</tr>
<tr>
<td>sct</td>
<td>19</td>
<td>7</td>
<td>83</td>
<td>6.4798e-04</td>
<td>9.21%</td>
<td>6.31%</td>
<td>0.8967</td>
<td>600</td>
</tr>
<tr>
<td>x2</td>
<td>10</td>
<td>3</td>
<td>50</td>
<td>5.4249e-04</td>
<td>14.35%</td>
<td>2.43%</td>
<td>0.8734</td>
<td>500</td>
</tr>
</tbody>
</table>
The current macro-models obtained after the characterization flow, were 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 (RMSE) to compare the estimated waveform with the actual waveforms, given by:

\[ RMSE = \sqrt{\frac{1}{K} \sum_{l=0}^{K-1} ||i(l) - \hat{i}(l)||^2} \]  

(5.1)

where \(i(l)\) is the actual current waveform, and \(\hat{i}(l)\) is the estimated current waveform, and \(K\) is the actual length of the current waveform sequence without zero padding. Table 5.1, shows the average RMSE for various benchmark circuits, under the column \(RMS_{\text{avg}}\), which is computed as

\[ RMS_{\text{avg}} = \frac{1}{P} \sum_{m=1}^{P} RMSE(m) \]  

(5.2)

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 RMSE for each vector pair with the actual peak current. Such a comparison can be used to determine the quantity of error and the significance of the error. For instance, if the error is large, but the current amplitude is very small, the error would not be too significant. In Fig. 5.1 we show the combined RMSE versus peak current plot for the benchmark circuits listed in Table 5.1. The plot shows that the RMSE is more or less bounded across the various current peak values for the benchmark circuits and that it increases very slowly with 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 accuracy of the macro-model. The coefficient of determination \((R^2)\) as explained in Chapter 4, 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 \(R^2_{\text{avg}}\), the average coefficient of determination. In Table 5.1, we have included the average coefficient under the column \(R^2_{\text{avg}}\).
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

$$err = \frac{||i_{x\text{actual}} - i_{x\text{est}}||}{i_{x\text{actual}}}$$  \hspace{1cm} (5.3)$$

where $i_x$ is either the peak current $i_p$ or the time instant at which the peak occurs $i_t$. In Table 5.1, $e_{p\text{avg}}$ denotes the average error in peak current estimation, and $e_{t\text{avg}}$ 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

$$e_{x\text{avg}} = 100 \frac{1}{P} \sum_{m=1}^{P} err(m)$$  \hspace{1cm} (5.4)$$

![Figure 5.1 RMSE vs. peak current for circuits listed in Table 5.1](image1)

![Figure 5.2 $i_{p\text{actual}}$ vs. $i_{p\text{est}}$ for circuits listed in Table 5.1](image2)

Table 5.1 shows that the average error in peak current estimation is less that 20% for all the circuits considered. It also shows that the average error in $i_t$ is less that 15% in all cases. In order to show that the $i_{p\text{est}}$ and $i_{t\text{est}}$ 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. 5.2 and 5.3. In Fig. 5.2, we show a combined plot of $i_{p\text{actual}}$ versus $i_{p\text{est}}$ for the benchmark circuits listed in Table 5.1, and
similarly in Fig. 5.3 we show a combined plot of $i_{\text{actual}}$ versus $i_{\text{test}}$. The correlation plots are mostly linear. The time instant plots show fewer number of points because most of the points are superimposed on top of each other.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure}
\caption{$i_{\text{actual}}$ vs. $i_{\text{test}}$ for circuits listed in Table 5.1}
\end{figure}

Finally, in Figs. 5.4–5.41, we present a qualitative comparison of the actual and estimated current waveform for the benchmark circuits listed in Table 5.1 for some example Hamming distances. 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. 5.4, we show the current waveform plot for alu2 where only 4 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. Having said all this, 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 5.4** Actual vs. estimated current waveform for alu2, Hamming distance = 4

**Figure 5.5** Actual vs. estimated current waveform for alu2, Hamming distance = 3
Figure 5.6 Actual vs. estimated current waveform for alu2, Hamming distance = 7

Figure 5.7 Actual vs. estimated current waveform for alu2, Hamming distance = 8
Figure 5.8 Actual vs. estimated current waveform for c432, Hamming distance = 10

Figure 5.9 Actual vs. estimated current waveform for c432, Hamming distance = 32
Figure 5.10 Actual vs. estimated current waveform for c880, Hamming distance = 21

Figure 5.11 Actual vs. estimated current waveform for c880, Hamming distance = 41
Figure 5.12 Actual vs. estimated current waveform for c880, Hamming distance = 55

Figure 5.13 Actual vs. estimated current waveform for cu, Hamming distance = 2
Figure 5.14 Actual vs. estimated current waveform for cu, Hamming distance = 6

Figure 5.15 Actual vs. estimated current waveform for cu, Hamming distance = 10
**Figure 5.16** Actual vs. estimated current waveform for cu, Hamming distance = 12

**Figure 5.17** Actual vs. estimated current waveform for parity, Hamming distance = 4
Figure 5.18 Actual vs. estimated current waveform for parity, Hamming distance = 14

Figure 5.19 Actual vs. estimated current waveform for random8, Hamming distance = 4
Figure 5.20 Actual vs. estimated current waveform for random8, Hamming distance = 6

Figure 5.21 Actual vs. estimated current waveform for x2, Hamming distance = 4
Figure 5.22 Actual vs. estimated current waveform for x2, Hamming distance = 8

Figure 5.23 Actual vs. estimated current waveform for f51m, Hamming distance = 7
Figure 5.24 Actual vs. estimated current waveform for mux, Hamming distance = 19

Figure 5.25 Actual vs. estimated current waveform for mux, Hamming distance = 4
Figure 5.26 Actual vs. estimated current waveform for vdao, Hamming distance = 10

Figure 5.27 Actual vs. estimated current waveform for alu2, Hamming distance = 2
Figure 5.28 Actual vs. estimated current waveform for alu2, Hamming distance = 2

Figure 5.29 Actual vs. estimated current waveform for alu2, Hamming distance = 2
Figure 5.30 Actual vs. estimated current waveform for alu2, Hamming distance = 3

Figure 5.31 Actual vs. estimated current waveform for alu2, Hamming distance = 3
Figure 5.32 Actual vs. estimated current waveform for alu2, Hamming distance = 3

Figure 5.33 Actual vs. estimated current waveform for alu2, Hamming distance = 3
Figure 5.34 Actual vs. estimated current waveform for c432, Hamming distance = 26

Figure 5.35 Actual vs. estimated current waveform for c432, Hamming distance = 26
Figure 5.36 Actual vs. estimated current waveform for c432, Hamming distance = 28

Figure 5.37 Actual vs. estimated current waveform for c432, Hamming distance = 28
Figure 5.38 Actual vs. estimated current waveform for c432, Hamming distance = 30

Figure 5.39 Actual vs. estimated current waveform for c499, Hamming distance = 35
Figure 5.40 Actual vs. estimated current waveform for pcler8, Hamming distance = 19

Figure 5.41 Actual vs. estimated current waveform for sct, Hamming distance = 14
CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Integrated circuits are drawing increasingly large currents from the supply, which is seriously complicating the design of the on-chip power/ground distribution network. Today’s high-end microprocessors, for example, can consume supply current of over 100 A, and it is getting worse. Therefore, it is important to do early design planning of the power grid in order to reduce expensive iterations at a later stage in the design cycle. To enable this, in an environment where design blocks are being reused (hard IP blocks), we have proposed a cycle-based current waveform modeling technique, for various logic blocks. This technique is based on modeling the discrete cosine transform (DCT) of the current waveform as a function of the input vector pairs applied to the primary inputs of a logic block. One can use inverse DCT to get the time-domain current waveform. The use of DCT 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 DCT of the current waveform shows much less variation. Actually large variations in the time-domain current waveform translate to variations mostly in the parameters of the DCT of the current waveform, not in its overall shape.

The DCT of the current waveform, for a given logic block has been modeled using analytical expressions. Two types of analytical expressions have been proposed to capture the two types of current waveform shapes that one commonly sees in practice, namely triangular current waveform and the trapezoidal current waveform. We used the relationship between DCT and continuous Fourier transform to infer the form the analytical expressions. The time domain current waveforms (both trapezoidal and triangular) were approximated with a piecewise-linear waveform. The Fourier transform expressions obtained for the even extension of these piecewise-linear waveforms was used get an insight into the the form of the actual current macro-model. Along with the triangular and trapezoidal current waveforms, we also discussed multiple peak current waveforms, that occur in practice. We proposed a simple partitioning algorithm to partition multiple peak current waveforms into their constituent single peak waveforms. This enables us to model the DCT of the constituent
elements of the multiple peak current waveform using the triangular or trapezoidal templates instead of deriving a single complicated template function for multiple peak waveforms.

The parameters of the template functions corresponding to the DCT of the current waveform were modeled as a function of the input vector pair, using regression. In order to formulate the polynomial function used to model the parameters, we presented a mapping from the Boolean vector pair to a set of dummy variables. We also used zero padding in the time domain to make the DCT smooth and amenable to regression. In order to account for the large variation among the parameters of the DCT template function, across input vectors, for a given logic block, we developed a classification for the input vector pairs, and constructed a separate model for each classification. The input vector pairs were partitioned into five groups depending on the fraction of primary inputs switching, and a different analytical expression (based on the two template functions) was constructed for each partition.

In order to construct the macro-models automatically without any user intervention, we presented a simple characterization flow to detect the correct template function and determine the correct sample size for performing regression. The characterization flow used a quantitative metric called the coefficient of determination to chose between the two template functions. It also used this metric to decide about partitioning. The basic philosophy of the characterization flow was to minimize the number of iterations needed to construct the model. The characterization flow used a sample size determination algorithm to get an adequate sample size to perform regression. This was needed because the exponential nature of the possible input vectors pairs for a given logic block, makes it virtually impossible to build a model using all possible input vector pairs. But the size of sample set used to build the model should be large enough to predict current waveforms for vector pairs outside the sample set. Therefore, we developed a sample size determination algorithm.

We implemented the current macro-model characterization flow and built models for many combinational benchmark circuits. The models were tested for accuracy on a different set of random input vector pairs, using various quantitative metrics like relative error in peak current estimation, relative error in time instant at which peak occurs, and coefficient of determination. The average relative error for both peak current estimation and the time instant were less than 20%. All the macro-models were constructed and tested using
HSPICE, which is universally acknowledged as a very accurate circuit simulator.

We have thus far summarized the contribution of this dissertation. We will now present some of the directions in which this dissertation can be extended. The proposed macro-modeling approach has been tested on combinational logic only. Our modeling approach may have to be modified slightly for sequential circuits where the supply current is a function of the input vector pair as well as the state transitions. In such a case it may be worthwhile to use the state transitions along with the input vector pair as the predictor variables. Since, sequential circuits also have a clock input, it may have to considered separately from the primary inputs and internal states of the sequential circuit. Also, in order to use the state information for current macro-modeling, all the states of the sequential logic should be observable.

The supply current drawn by a circuit is a function of input pattern and the I/O conditions also [17]. The I/O conditions here correspond to the slope of the input transitions, arrival times of the inputs and the output load. The proposed macro-model is a function of the input pattern and does not account for the I/O conditions. Because at the logic level (or at higher level of abstractions where the proposed macro-models would be used), signal slopes are neither represented nor propagated. The arrival times are also not very accurate in the absence of interconnect delay. Since the current macro-models can also be used at a lower level of abstraction, it would be worthwhile to explore the impact of these parameters on the current waveform. The different arrival times of the various inputs may be considered as different vector pairs and modeled using the proposed approach. In this approach, one may specify a threshold time interval and any set of inputs arriving during that interval would be considered as a single vector and modeled using the proposed approach.

In [17], the authors discussed the impact of slope of input transitions and output load on the current waveform for logic gates. They found that the slope of input transition impacts the location of current peak. But during our preliminary study of this problem at the block level, we found that at the block level the location of the peak does not change significantly for a range of input slew rates. Therefore, it might be possible to use a single macro-model for a range of input slew. In order to be able to use the macro-models at lower level of abstractions, it would be worthwhile to further explore the impact of input slew rates on block level current waveforms.
Since the current drawn by a logic block is also a function of the load capacitance connected to the outputs of the logic block, it would interesting to explore the variation in current waveform as a function of the output load driven by a logic block. One possible solution to this problem could be to decouple the block current and the load current. This would be an approximation, but it would be worthwhile to explore. After decoupling, one can use the macro-models for the no load current waveform and then use the load capacitance and the output slew rate to get the load current drawn from the power grid. One can use timing models or static timing analysis to get the output slew rates and logic simulation to determine the outputs making a low to high transition. There might be some charge sharing between outputs making a high to low transition and low to high transition, which might be ignored by the above technique but it would still be worthwhile to explore such solutions in future. It would an interesting direction for future research.

So far we have developed macro-models for circuits with gate counts less than 400, because of the limitation of the circuit simulator. It would be interesting use this technique in a larger context and use it along with a linear solver to actually perform “dynamic” power-grid analysis.

Apart from being useful in the design an analysis of power grids, the proposed current macro-model technique can be used to estimate power, energy-per-cycle, or even power dissipated in the power grid due to the resistance of the metal lines (joule heating). It would be useful to see how the proposed technique can be used in different contexts of the VLSI design flow, to analyze some of the above issues.
REFERENCES


83


VITA

Srinivas Bodapati received the B.Tech (Honors) degree from the Indian Institute of Technology, Kharagpur, India, in 1995 and the M.S degree from University of Illinois at Urbana-Champaign in 2000, both in electrical engineering. Since August 1998 he has been a research assistant in the Coordinated Science Laboratory at University of Illinois at Urbana-Champaign. In the summer of 2001, he worked with Ambit group of Cadence Design Systems in San Jose, California. He has also worked at Cadence Design Systems, Noida, India from 1995 to 1997 and at Texas Instruments, Bangalore, India from 1997 to 1998. Srinivas Bodapati is currently a student member of the IEEE.