

1997

Complete-Range Activity-Based RTL Power Estimation

Nur Kurt-Karsilayan
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Electrical and Computer Engineering Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Kurt-Karsilayan, Nur, "Complete-Range Activity-Based RTL Power Estimation" (1997). *Dissertations and Theses*. Paper 6361.

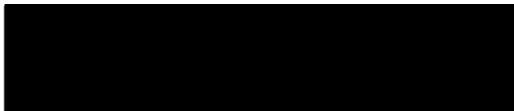
<https://doi.org/10.15760/etd.3507>

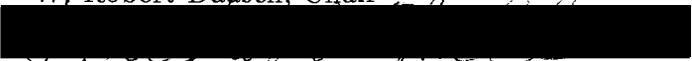
This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

THESIS APPROVAL

The abstract and thesis of Nur Kurt-Karsilayan for the Master of Science in Electrical Engineering were presented November 13, 1997, and accepted by the thesis committee and the department.

COMMITTEE APPROVALS:


W. Robert Daasch, Chair


Douglas V. Hall


Gerry Recktenwald
Representative of the Office of Graduate Studies

DEPARTMENT APPROVAL:


Lee W. Casperson, Chairman
Department of Electrical Engineering

ABSTRACT

An abstract of the thesis of Nur Kurt-Karsilayan for the Master of Science in Electrical Engineering presented November 13, 1997.

Title: Complete-Range Activity-Based RTL Power Estimation

In recent years, power consumption has become a major concern in the electronic industry. Power reduction can be accelerated in the design cycle by fast and accurate power estimation tools. Since the units of lower-levels of design abstraction are transistors or gates, power estimation becomes a slow process at these levels. Therefore designers need to have tools for fast and accurate power estimation at the higher levels of design abstraction such as register transfer level (RTL).

A novel RTL power estimation technique called CRAB-RPE will be presented in this thesis. The CRAB power model is built upon four important properties which most of the previous RTL models did not support at the same time. First, the model is based solely on the first and second-order primary input bit-level transition probabilities which provide detailed information about the primary input bit activity dependency of the circuit. Second, the model is based on the power characterization of a microarchitecture library with a complete range of primary input bit transition probabilities without any assumptions about this activity. Third, the pairwise spatial correlations of the primary input nodes are considered by including second-order crossterms of the primary input switching probabilities. Fourth, the first-order temporal correlations of the primary input bits are considered by including 1 to 1 and binary switching transition probabilities. With the proposed

model, fast power estimation can be achieved from input bit-level statistics without further simulation. The model was evaluated using the ISCAS combinational circuit benchmarks and other commonly used micro-architectural circuit blocks. Second-order terms were observed to be important for modeling the low bit activity effects on power dissipation. The CRAB power model returned under 5% of the low-level simulator estimates for either biased single, pair PIN statistics or uniform white noise, DBT-like data.

COMPLETE-RANGE ACTIVITY-BASED RTL POWER ESTIMATION

by
NUR KURT-KARSILAYAN

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Portland State University
1998

To my husband, Ilker.

Acknowledgements

First, I would like to acknowledge Mentor Graphics for its financial support. I would like to thank especially Jeff Jones, Jeff Wilson and Sky Soo at Mentor Graphics for their guidance and encouragements. I would also like to thank Du Nguyen who involved in the RTL Power Estimation project earlier.

I would like to thank my advisor, W. Robert Daasch for his guidance and the valuable discussions we had throughout the whole year. It was really fun to work with him.

Doug Hall and Gerald Recktenwald provided valuable feedback as members of my thesis committee that I would like to thank both.

I would also like to thank Prof. Rolf Schaumann for his invaluable support and encouragement.

Lastly, I would like to thank our family members who are thousands of miles away from us. I know that, despite that distance in miles between us, you all had faith in me which gave me power to finish this thesis.

And to my husband, Ilker. You were the one and only who shared my miserable and happy moods. Thank you for your infinite support.

Contents

Acknowledgements	i
List of Figures	iv
List of Tables	vi
1 Introduction	1
1.1 Power Dissipation Ingredients	2
1.2 Low Power Design Space	4
1.3 Low-Level Power Estimation	6
1.3.1 Simulation-Based Techniques	6
1.3.2 Analytical Techniques	7
2 Previous Work in RTL Power Estimation	10
2.1 Predictive Techniques	13
2.1.1 Complexity-Based Models	13
2.1.2 Entropy-Based Models	15
2.2 Descriptive Techniques	18
2.2.1 Fixed-Activity Models	19
2.2.2 Activity-Sensitive Models	20
2.3 Chapter Summary	23
3 CRAB-RPE: Complete-Range Activity-Based RTL Power Estimation	24
3.1 CRAB-RPM: CRAB RTL Power Model	25
3.2 CRAB-PC: CRAB Power Characterization	32
3.2.1 IVG: Input Vector Generator	34
3.2.2 MCE: Model Coefficient Extractor	35
3.2.3 Power Simulator	36
3.3 CRAB-PA: CRAB Power Analysis	36
3.3.1 HLS: High Level Synthesis	36
3.3.2 PE: Power Evaluation	38
3.3.3 Simulator	38
3.4 Limitations and Extensions	39

3.5	Chapter Summary	40
4	CRAB Model Evaluation	41
4.1	Experimental environment	42
4.1.1	CRAB-PC Implementation	43
4.1.2	CRAB-PE Implementation	54
4.2	CRAB Experimental Results	55
4.2.1	NUR.ALU Results	56
4.2.2	MAGCMP Results	59
4.2.3	ADD4 Results	61
4.2.4	Comparison of the Proposed Models with C17	64
4.2.5	C432 Results	71
4.2.6	C499 Results	74
4.2.7	C1908 Results	76
4.2.8	C6288 Results	79
4.3	Chapter Summary	81
5	Future Work	83
6	Conclusion	85
	Bibliography	87

List of Figures

1.1	Power estimation time vs level of abstraction.	2
1.2	Power dissipation ingredients in CMOS circuits.	3
2.1	Structural RTL representation of a chip.	11
2.2	General design flow for RTL power estimation.	12
3.1	BDD of the output switching function of a 2-input AND gate.	28
3.2	Second-order effect of a PIN on a primary output node.	31
3.3	Matrix representation of the CRAB power model for the character- ization phase.	33
3.4	CRAB Power Characterization phase.	34
3.5	CRAB Power Analysis phase.	37
4.1	Implementation of CRAB-PC.	44
4.2	c17.vhdl before modification.	46
4.3	c17.vhdl after modification.	47
4.4	Illustration of average power dependency on t_i^{sw} and t_i^{11}	49
4.5	PINSTAT Algorithm for the specification of PIN statistics.	50
4.6	Pascal Triangle.	50
4.7	The PIN statistics generated by PINSTAT(3,5%,95%).	52
4.8	Implementation of CRAB Power Evaluation.	55
4.9	Relative error vs. PIN statistics of the 18 CRAB-PE results for NUR.ALU with the first-order model.	57
4.10	Comparison of the proposed first-order model with CES and IRSIM power values.	58
4.11	Relative error vs. PIN statistics for the CRAB-PE of MAGCMP.	60
4.12	Relative error distribution for the LSS of ADD4 CRAB coefficients.	61
4.13	Relative error distribution of 48 CRAB-PE results for ADD4 with the original model.	63
4.14	Relative error distribution of 36 CRAB-PE results for ADD4 with the original model.	63
4.15	Relative error distribution of the LSS for C17 with the original model.	66
4.16	Relative error distribution of the LSS for C17 with the quadratic model.	67
4.17	Relative error distribution of the LSS for C17 with the third model.	67

4.18	Relative error distribution of the LSS for C17 with the CRAB model.	68
4.19	Relative error distribution of 41 CRAB-PE results for C17 with the original model.	68
4.20	Relative error distribution of 41 CRAB-PE results for C17 with the quadratic model.	69
4.21	Relative error distribution of 41 CRAB-PE results for C17 with the third model.	69
4.22	Relative error distribution of 41 CRAB-PE results for C17 with the CRAB model.	70
4.23	Relative error distribution of 10 CRAB-PE results using random (uniform white noise) data for C17.	70
4.24	Relative error distribution of the LSS for C432.	71
4.25	Relative error distribution of the 704 CRAB-PE results for C432.	72
4.26	Relative error distribution of the 680 CRAB-PE results for C432.	73
4.27	Relative error distribution of the 29 CRAB-PE results for C432 using uniform white noise data.	74
4.28	Relative error distribution of the LSS for C499.	75
4.29	Relative error distribution of the 827 CRAB-PE results for C499.	76
4.30	Relative error distribution of the LSS for C1908.	77
4.31	Relative error distribution of the 512 CRAB-PE results for C1908.	78
4.32	Relative error distribution of the LSS for C6288.	79
4.33	Relative error distribution of the 121 CRAB-PE results for C6288.	80

List of Tables

3.1	Notation.	26
3.2	Power models and number of coefficients.	30
4.1	Circuits for the experiments.	43
4.2	Number of pattern sets for each PIN bias.	51
4.3	Standard deviations from average power for different vector lengths.	53
4.4	The range and the number of four model coefficients.	65

Chapter 1

Introduction

In recent years, *power dissipation* has become an important design concern for CMOS circuits. Previously, the design specifications were based on area, performance, cost, and reliability. The change in requirements is the result of the remarkable growth of personal computing devices and wireless communications systems which demand high-speed computation and complex functionality with low power consumption [2]. In most of the applications, the overall aim of power reduction is to decrease system cost, such as cooling, packaging, and energy, and ensure long-term circuit reliability. On the other hand, peak power dissipation is a separate concern for determining the electrical limits of the design, battery type and power distribution network [4]. While technology, layout, gate, and circuit optimizations may offer power reductions of a factor of two at best, optimization at the register transfer level (RTL) and system level was shown to result in an order of magnitude reduction in *average power dissipation* [3]. The relationship between power estimation time versus the design abstraction level [1] is illustrated in Fig. 1.1. Clearly, there is a trade-off between the power estimation time cost and the level of design abstraction. Power estimation is faster at the higher levels of design abstraction.

In this thesis, models for average power dissipation will be investigated at the

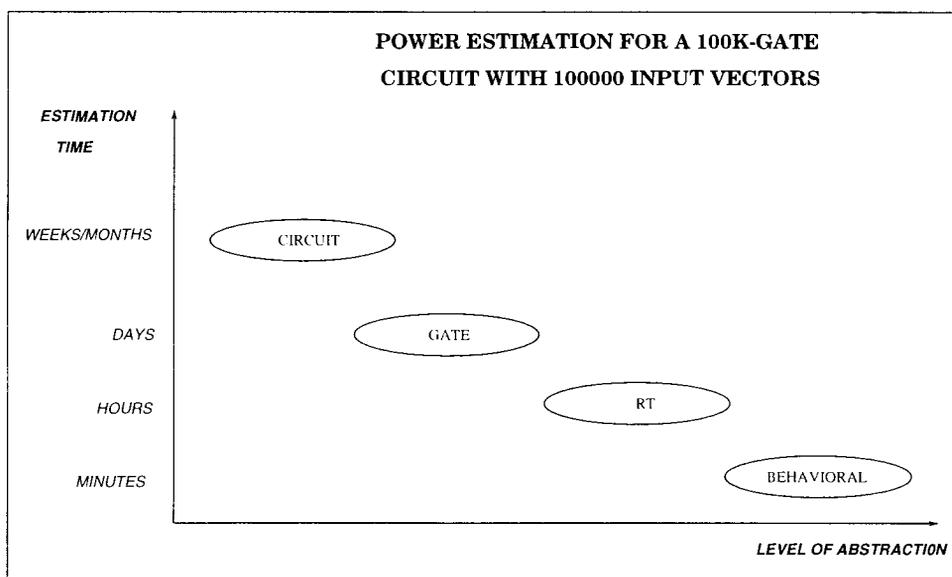


Figure 1.1: Power estimation time vs level of abstraction.

register transfer level of design abstraction. To explore the low power design space for digital CMOS circuits, the sources of power dissipation need to be specified and then methods to reduce power can be investigated.

1.1 Power Dissipation Ingredients

There are four main factors that contribute to power dissipation in digital CMOS circuits [5]:

- *Capacitive current* charges and discharges the capacitive load during output transitions which are shown with arrows labeled 1 in Fig. 1.2.
- *Short-circuit current* between the supply rails when both NMOS and PMOS are on during an input transition. This current is shown with the arrow labeled 2 in Fig. 1.2.
- *Leakage current* is determined by the fabrication technology. It is a reverse

bias current between MOS diffusion regions and the substrate in an MOS transistor and the sub-threshold conduction caused by the inversion charge that exists below the MOSFET threshold voltage. The leakage current is indicated with the arrow labeled 3 in Fig. 1.2. The actual power contribution of the substrate current is several orders of magnitude below other contributors [3].

- *Standby current* is drawn continuously from the power supply (e.g. pseudo-NMOS inverter). It is a small current and a small contribution to power dissipation.

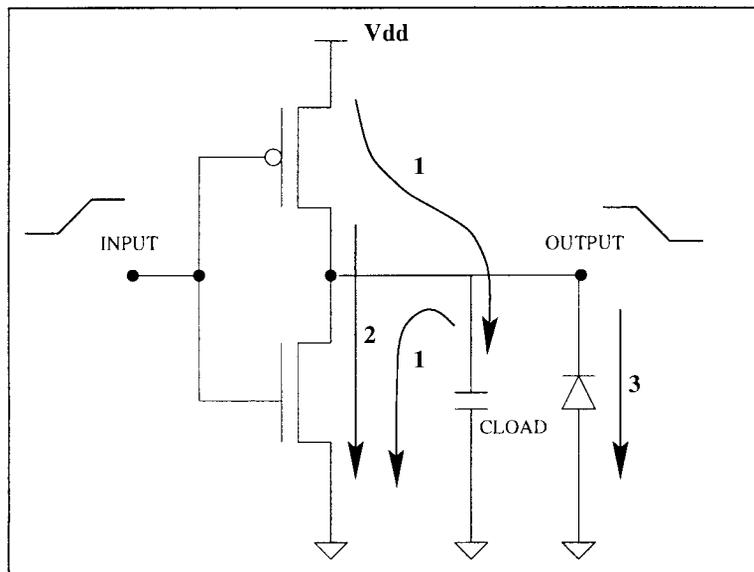


Figure 1.2: Power dissipation ingredients in CMOS circuits.

Total power dissipation in a CMOS circuit is the sum of the above power ingredients. *Dynamic power dissipation* is the sum of power dissipations caused by the capacitive and short-circuit currents [2]. Veendrick showed that if the gate sizes are selected such that the input and output rise/fall times are about equal, the short-circuit power dissipation will be less than 15% of the total power dissipation [6]. For modern CMOS circuits, the capacitive power is the dominant ingredient of the

total power. The average capacitive power of a CMOS circuit can be expressed as:

$$P_{ave} = \frac{1}{2} \cdot V_{dd}^2 \cdot C_{load} \cdot t^{sw} \cdot f_{clk} \quad (1.1)$$

where C_{load} is the load capacitance at the output node, t^{sw} is the switching activity of the output node (i.e. sum of 1 to 0 and 0 to 1 transition probabilities per clock cycle), V_{dd} is the supply voltage, and f_{clk} is the clock frequency. The $C_{load} \cdot t^{sw}$ product is sometimes called *switched capacitance* [3] and the $t^{sw} \cdot f_{clk}$ product is known as the *transition density* in [8].

The low power design space will be explored by describing the effects of each component (V_{dd} , C_{load} , t^{sw}) on the average power dissipation.

1.2 Low Power Design Space

The most effective component in the power reduction is supply voltage. Since the power is proportional to the square of the voltage, a factor of two reduction in voltage will result in four-fold power reduction. This is a global effect throughout the entire design. In many cases, there is a trade-off between the reduced voltage and circuit performance or area [2]. One processing technique for example, reduces supply voltage without sacrificing speed by decreasing of the threshold voltage of the devices.

Power dissipation also depends on the load capacitance of each node in the circuit. The transistor interconnect contributes to the load capacitance of the connecting nodes. Logic minimization, using smaller devices, fewer and shorter wires, resource sharing are several ways to reduce load capacitance [2]. Interconnect, for example, can be reduced by register sharing and common sub-function extraction or by effective placement and routing. The trade-off between the capacitive

loads and the performance of the circuits implies that the capacitances can not be reduced independently.

If the switching activity is minimized, power dissipation is reduced. Switching activity is composed of two components: Functional activity and glitch activity. Functional activity of a circuit depends directly on its logic function. Glitch activity is caused by the unwanted transitions that occur at a circuit node before the signal reaches its steady-state value.

The switching activity of the output node of a circuit depends on [2]:

1. Switching activities of the circuit inputs,
2. Spatial and temporal correlations among the circuit inputs,
3. Delay model,
4. Logic function of the circuit.

The first is related to the input pattern-dependency of power as discussed in [10]. Switching activity may be different for different circuits and various data representations. For example, switching activity of a finite state machine (FSM) varies between 0.08 and 0.18 [4]. For video signals, the most significant bits have switching activities of 0.1 whereas the least significant bits have 0.5. The commonly used *uniform white noise data* has a switching activity range 0.4-0.5.

The second is local and temporal correlations of the input signals. For example, if two bits of a word are always high at the same time then these bits are *spatially correlated*. The spatial correlation occurs for many internal nodes of a circuit by a mechanism called *re-convergent fanout* [10]. On the other hand, if there is a dependency among the values of an input signal in time domain (e.g a signal is 1 if and only if its previous value is 0) then the values for that particular bit are *temporally correlated*. For example, a feedback in a finite state machine often creates

temporally correlated signals [10]. It has been shown conclusively that spatial and temporal correlations will affect the switching activity of an output node [2]. The delay model used is also important for the calculation of the switching activity. If a zero-delay model is used, then only the *functional activity* can be obtained for the switching activity, on the other hand a general delay model can predict *glitch activity* in addition to *functional activity*. The logic function itself naturally also plays a role on the switching activity. For example, the functional activity of a 2-input XOR's output is $1/2$, when all (i.e. 16) possible input transitions are considered.

The design for low power is dependent upon power estimation and optimization tools that the designer uses to make critical design choices without expensive redesigns later in the design cycle. There are several levels of CMOS design abstraction. The two *lowest levels* use transistors and gates as building blocks of the circuit. At higher levels of abstraction, circuits are described as interconnected multi-function registers. This is generally known as the register transfer level.

1.3 Low-Level Power Estimation

The low-level power estimation techniques can be grouped into two categories: Simulation-based techniques or analytical techniques.

1.3.1 Simulation-Based Techniques

Transistor-level, simulation-based techniques simulate the circuit with a sample set of input vectors [2]. They are advantageous in terms of the accuracy with respect to real (silicon) power dissipation and they can handle various device mod-

els and different circuit design styles. Their disadvantages come from the fact that they are not scalable to VLSI levels because they require large memory and execution-time. PowerMill and IRSIM are the examples of this type of simulator [35, 36].

A second technique uses a hierarchy of simulators to achieve a reasonable accuracy and efficiency trade-off. Entice-Aspen [37] is based on this technique where Aspen computes the circuit-activity information and Entice computes the power characterization data.

Another simulation-based technique uses statistical sampling based on a Monte Carlo Simulation approach that solves the pattern dependence problem by an appropriate choice of input vectors [9]. However, this method does not include spatial correlations at the input.

1.3.2 Analytical Techniques

Analytical techniques propagate the circuit node signal probabilities or node transition densities of the primary input bits through the circuit with little or no simulation (i.e. when the primary input probabilities are provided by the user). These techniques can also be grouped in two categories depending on the static delay-model used.

One technique is based on a zero-delay assumption. In addition to this, the values of each input in consecutive clock cycles are assumed to be temporally independent. Based on these assumptions the switching activity (t^{sw}) of the circuit node can be expressed in terms of its signal probability (p) as

$$t^{sw} = 2 \cdot p \cdot (1 - p) \quad (1.2)$$

where p is the probability of a signal being 1. Ercolani et. al, presented a procedure for propagating signal probabilities from the circuit inputs toward the circuit outputs by considering only *pairwise spatial correlations* (i.e. two bits are considered to be locally dependent on each other). Marculescu et. al. [25] and Schneider et.al. [26] proposed to model the temporal correlation of two consecutive signals by a time-homogeneous two-state Markov chain. The various transition probabilities can be computed exactly using the ordered binary decision diagram (OBDD) representation of the logic function in terms of the primary input nodes. Marculescu also proposed in [25] a method to propagate the transition probabilities and correlation coefficients through a gate-level circuit.

The second group is based on a nonzero delay model. Glitches, that are not modeled in the zero-delay model, are present when these delay models are used. Najm presented an efficient algorithm using the Boolean difference operation ($P(\frac{\partial}{\partial})$) to propagate the primary input transition densities through the circuit [8]. The transition density of each output node (td_o) is computed in terms of the input transition densities (td_i) as

$$td_o = \sum_{i=1}^n P\left(\frac{\partial o}{\partial i}\right) \cdot td_i \quad (1.3)$$

Mehta et. al. [27] improved the accuracy of the transition density propagation by using higher order terms for the input transition densities (td_i) such as second-order products (i.e. $td_i \cdot td_j$).

All of the low-level power estimation techniques introduced thus far require time and memory constraints for very large designs to retain the accuracy of the power dissipation estimate. Today, designers want to make selections between two or more different pieces of circuit alternatives, based on power dissipation levels

of each. In this case, the high-level power estimation or optimization is more important than these lower levels of abstraction. For CMOS circuit design, today, an estimate of a register transfer level (RTL) power dissipation is preferred by the designers.

In chapter 2, the current CMOS register transfer level (RTL) power dissipation estimation techniques will be discussed. A new RTL power model called CRAB will be introduced in Chapter 3. Chapter 4 will report the experimental results for characterizing and evaluating the CRAB model. The future work and the conclusion remarks will conclude the thesis.

Chapter 2

Previous Work in RTL Power Estimation

Compared to lower levels (i.e. gate-level or transistor-level), estimation of average power at the register transfer level (RTL) has two superior features. First, power estimation is available at an earlier stage of the design. In the absence of RTL power estimation, synthesis of RTL design to gates followed by simulation with the gate-level power tools are necessary tasks that must be added to the design cycle. Second, power estimation should be faster, at the register transfer level. This second observation is because the units of lower level abstraction are transistors and gates whereas a register transfer level design can be described structurally by reusable micro-architectural blocks (e.g. adders, multipliers, control and memory etc.). Hence the granularity of an RTL description is larger which should yield faster power analysis. Fig. 2.1 shows the structural RTL representation of a chip [3]. *Micro-architectural block* has the same meaning as *RTL library component*, *RTL design sub-block*, *module* or *block* and these will be used interchangeably throughout the thesis.

The main component in the power dissipation Eq. 2.1 is the *switched capacitance* ($C_{load} \cdot t^{sw}$) which is the product of switching activity (t^{sw}) and physical

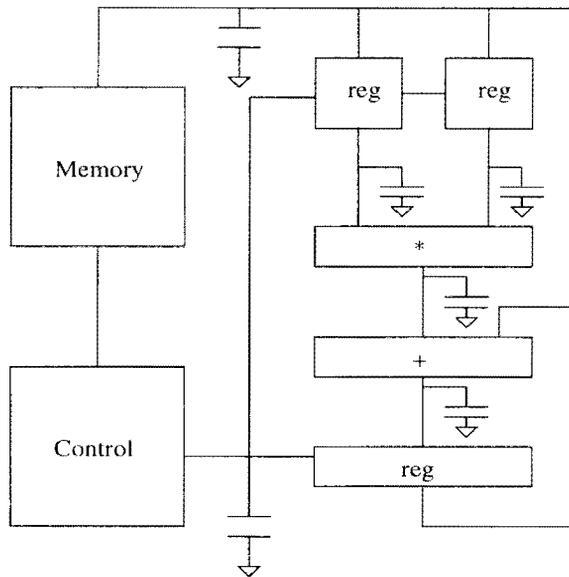


Figure 2.1: Structural RTL representation of a chip.

capacitance (C_{load}) of the micro-architectural block. Note that static power and short-circuit power are excluded in Eq. 2.1, since they contribute at most to 15% of the total power dissipation.

$$P_{ave} = \frac{1}{2} \cdot V_{dd}^2 \cdot C_{load} \cdot t^{sw} \cdot f_{clk} \quad (2.1)$$

Hence power modeling, characterization and evaluation is to be based on the total switched capacitance. The general flow from power modeling to power evaluation is depicted in Fig. 2.2. In order to find the total power dissipation of an RTL design, power contributions from each RTL design sub-block must be evaluated as shown in Fig. 2.2. This figure will be useful for the following sections.

There are two main approaches to model the *power dissipation* or *switched capacitance* of a CMOS circuit at the RT-Level. One is *predictive approach* which estimates power dissipation without a pre-characterization step. Most of these methods are based on either the complexity of the block or the entropy of primary

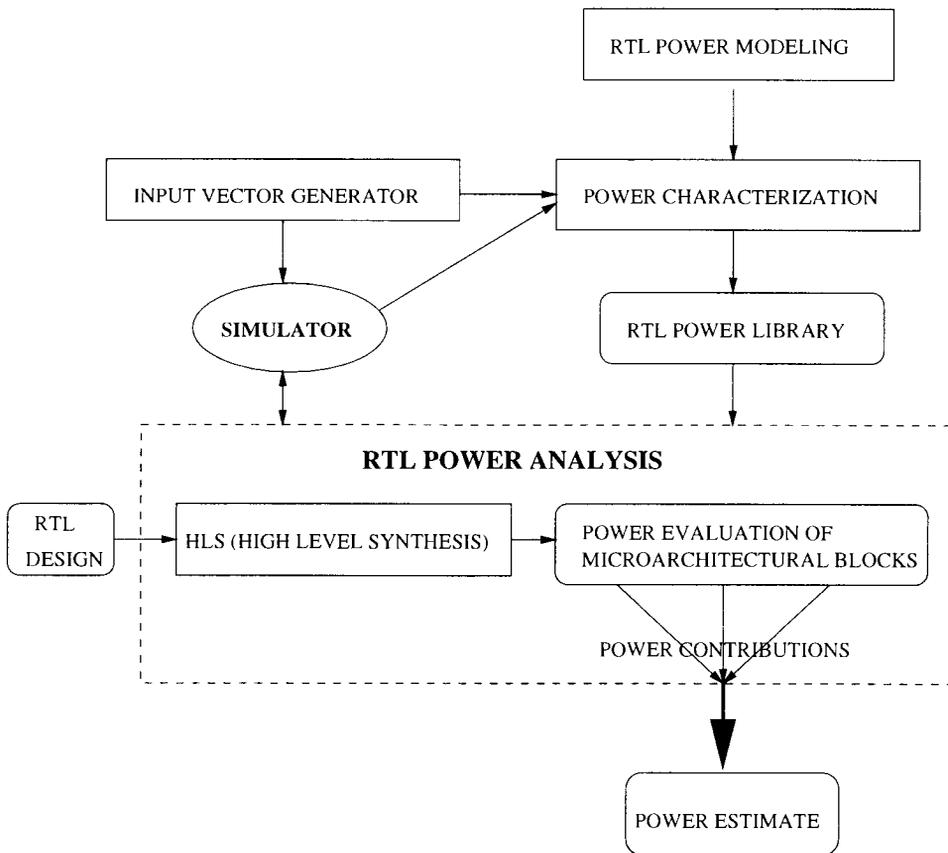


Figure 2.2: General design flow for RTL power estimation.

input and output nodes. Entropy is an information theoretic measure which specifies the amount of computational work. The other approach is *descriptive* and is based on a pre-characterization step. The model parameters obtained at the pre-characterization step are then used to estimate the power dissipation during a later power analysis step shown in Fig. 2.2. The details of the techniques will be discussed in the following sections.

2.1 Predictive Techniques

Previous predictive RTL power estimation work [7, 12, 13, 15] is based on modeling the switched capacitance or power without a pre-characterization step. The elimination of the characterization step in Fig. 2.2 results in saving the time for characterization in the design flow, although, the characterization step is visited only once for each micro-architectural block. Without such a step many predictive models return inaccurate power estimates when compared to lower level power estimates. Predictive techniques for power estimation of CMOS datapath circuits that have been proposed thus far depend on:

- Complexity-based Models or,
- Entropy-based Models.

2.1.1 Complexity-Based Models

Complexity-based approaches provide fast but inaccurate power values (when compared to low level power simulator results) by estimating the complexity of the circuit by a measure called *gate equivalent count* [12, 13]. Gate equivalent count specifies the average number of reference gates which are required to build a particular functional block.

The Chip Estimation System (CES) [12], is an example of a complexity-based predictive technique. The model for functional block average power is

$$P = GE \cdot (E_{typ} + V_{DD}^2 \cdot C_L) \cdot f \cdot A_{int} \quad (2.2)$$

where GE is the gate equivalent count for the functional block, E_{typ} is the typical power dissipation per MHz for a specific gate (e.g. NAND2 gate), C_L is the

estimated load capacity, f is the switching rate and A_{int} defines the percentage of internal gates switching per clock cycle.

CES depends on a direct relationship between the complexity of a chip architecture and the *gate equivalent count*. However, gate equivalent counts are based upon a single reference gate and do not provide reliable power estimates for different circuit styles and layout techniques. Furthermore, CES does not take the input activity into account. It predicts the same power estimate for extremely different applied input patterns.

Liu and Svensson presented a power model, conceptually similar to CES, for random logic circuits [13]. The average logic gate function was defined as a three input AND gate (buffered NAND) connected to three identical AND gates at the output node. The effective capacitance of the reference gate (C_{logicg}) excluding clock driving nodes is defined as in Eq. 2.3, where f_g is the average fan in and fan out, f_d is the duty factor, k_1 is the number of minimum size gate capacitances in one input node, k_2 is the number of the minimum size gate or diffusion capacitances on the buffer input node and C_{tr} is a minimum size NMOS transistor capacitance.

$$C_{logicg} = f_g f_d k_1 C_{tr} + f_d k_2 C_{tr} + 3 f_d C_{tr} + k_3 f_d C_{tr} \quad (2.3)$$

To obtain the total switched capacitance for the logic gate and, define the gate power, the clocked and unclocked node capacitances are added to C_{logicg} [13]. Like CES, this method does not consider the effects of input activity on a circuit's power dissipation.

Complexity-based predictive techniques are useful because they require only a little information about the circuit such as gate-equivalent counts or technology parameters. On the other hand, for various input activities, the power estimates

are equal which is not the case in reality because the average power varies with the type of input patterns applied.

2.1.2 Entropy-Based Models

Currently, several researchers have introduced entropy, an information theoretic measure to estimate circuit activity [7, 15]. The basic assumption is that there is a uniform distribution of transition activity over all nodes, such that average power can be written as a product of average transition activity and total capacitance.

$$P_{ave} \propto \sum_{i=1}^N C_i \cdot t_i \approx t \sum_{i=1}^N C_i \quad (2.4)$$

where C_i is the node capacitance, t_i is the node transition activity and i represents the gate output nodes in the circuit. In [7, 15], average transition activity t is defined as

$$t = \frac{1}{N} \sum_{i=1}^N t_i \quad (2.5)$$

The entropy of a random Boolean variable X is defined as

$$H(X) = p \cdot \log_2 \frac{1}{p} + (1 - p) \cdot \log_2 \frac{1}{(1 - p)} \quad (2.6)$$

where p represents the signal probability. In [7], the values of the signals are assumed to be independent for consecutive clock cycles. From this, the average node transition activity is

$$t = 2 \cdot p \cdot (1 - p) \quad (2.7)$$

When the normalized activity and normalized entropy are compared, they are nearly equal for the entire range $p[0,1]$. From this, the average entropy (H) replaces average node transition density (t) in Eq. 2.8.

$$P_{ave} \propto H \cdot \sum_{i=1}^N C_i \quad (2.8)$$

In [7], a model is derived for the average entropy as a function of input and output entropies. The final expression obtained for the average entropy H is shown in Eq. 2.9, where n is the number of primary inputs, m is the number of primary outputs, H_i is the total input entropy and H_o is the total output entropy.

$$H \approx \frac{2/3}{n+m} \cdot (H_i + 2H_o) \quad (2.9)$$

Nemani and Najm also propose an entropic bound for the area (i.e. total capacitance) of the circuit [7]. The area is largely determined by the number of gates in the circuit and can be used to estimate the total capacitance $\sum_{i=1}^N C_i$ in Eq. 2.8. Their proposed bound is computed using the output entropy and the number of input bits.

$$0.4 \cdot H_o \cdot \frac{n}{\log_{10} n} \leq A \leq 2 \cdot H_o \cdot (n \cdot \log_{10} n) \quad (2.10)$$

In comparison to ISCAS benchmarks, this bound appears to be too large for some circuits. Nemani and Najm improved the area estimation by using a measure that they called *average cube complexity* [14]. *Average cube complexity*, is the average literal count of the prime implicants of the function. But this measure has been used for single output functions only [14].

Despite the efforts of [7] and [14], the activity relative errors have been larger than 100% as well as the area of the circuit and number of gates. The average entropy proposed in [7] has underestimated the circuit activity for some examples. This is likely understood because the circuit activity depends on the functionality of the circuit as well as the type of data being processed. For example, the proposed

model will return the same result for two different functions that have identical *average entropy*, although the power dissipation of each functional block may be different. Additionally, the *average entropy* may be predicted to be the same for two different sets of input data applied to the same circuit, although the power dissipation may be extremely different for each applied data. Another concern in the evaluation of the proposed models is that the applied input vectors were selected from uniform white noise (UWN). Hence for correlated input streams, the relative error for either activity or area would possibly be larger and is currently untested.

Marculescu et. al. have made an effort to estimate switching activity based on both the entropy (or informational energy) and the distribution of nodes in the circuit [15]. Their technique requires additional, either *structural* or *functional* information about the circuit. If there is no information provided by the user the circuit is *simulated* to obtain the average entropy for different node distributions (uniform, linear and exponential) [15] throughout the circuit levels or depth.

The required *structural information* can be the number of internal nodes, the number of logic levels or the actual distribution of nodes throughout the levels of the circuit. Output entropy of the circuit is not effected by the three different distributions. It is a function of input entropy, a structural-based scaling factor and the number of logic levels. To use the *functional information* for the power estimation, two function-dependent transmission coefficients (HTC and ETC) are defined in [15]. These coefficients are used to estimate the output entropy (or output informational energy) of the function from the input entropy (or input informational energy). Marculescu et. al. also introduced adjustments for the coefficients depending on the nature of input patterns (i.e. input patterns other

than uniform white noise). Hence for each functional block, the input entropy (or input informational energy) can be propagated to the primary outputs by using the proposed transmission coefficients. Then average entropy can be evaluated depending on the input-output entropies and the circuit node distribution. Then the the average power is evaluated directly by using Eq. 2.8.

The above approach is limited in practice [15]. For example, the temporal correlation of primary inputs is not considered. The suggested transmission coefficients (ETC, HTC) are applicable only to single well-defined functions, multi-functions such as a complex ALU require substantial work to define the coefficients. Furthermore, the proposed technique is valid only for zero-delay model.

The entropy-based techniques introduced so far have several practical limitations. For example, they do not account for glitches, because they assume a zero-delay mode of operation. Another assumption is the independence of signal values in consecutive clock cycles which does not take temporally correlated input vectors into account. Finally, the node transition activity is assumed to be uniformly distributed over all nodes.

In summary, predictive techniques have not offered a reliable power estimation because they do not provide a reliable model of real hardware and nature of data being processed.

2.2 Descriptive Techniques

Descriptive techniques offer an approach based on the pre-characterization of different functional blocks and construction of an *RTL Power Library* as shown in Fig. 2.2. The descriptive techniques reviewed for this thesis can be grouped

into two classes depending on the effects that the input patterns have on power dissipation of a circuit [10]. *Fixed-activity Models* are based on the assumption of constant activity for a functional block whereas *Activity-sensitive Models* reflect the activity dependency of power dissipation.

2.2.1 Fixed-Activity Models

Power Factor Approximation (PFA) is one of the descriptive techniques based on fixed-activity modeling [16]. It is basically the descriptive version of the technique introduced by Liu and Svensson [3]. Compared to [3], PFA is superior in practice because it requires power characterization of each specific module stored in the RTL library.

The proposed PFA [16] model for a specific micro-architectural block is of the form

$$P_{ave} = \kappa \cdot HC \cdot f \quad (2.11)$$

where κ is the proportionality constant, HC is the hardware complexity term and f is the activation frequency. For instance, the hardware complexity (HC) of a multiplier is assumed to have a quadratic relation with its word-length, f is the frequency at which the multiplications are performed. κ is the empirically extracted constant for a specific technology and supply voltage.

The disadvantage of the model is that the activity constant κ is intended to capture the internal activity of the circuit for all set of applied input patterns. However this is not the case, since there is an inevitable influence of applied input activity on power dissipation. Hence PFA does not return reliable results for input stimuli other than uniform white noise or other pre-selected patterns.

2.2.2 Activity-Sensitive Models

Recently, several researchers have directed their efforts in RTL power modeling by considering the effects of input activity on internal activity [3, 17, 21, 22, 24].

ESP (Early design Stage Power and performance simulator) [17] is one of the activity-sensitive methods that accounts for the cumulative effects of input activity on the power dissipation of datapath circuits. The proposed average power model is as shown in Eq. 2.12 and is based on the assumption that a bit transition causes some parts of the circuit to become active.

$$P_{ave} = P_{constant} + n \cdot P_{change} \quad (2.12)$$

$P_{constant}$ is interpreted as the constant power which is independent of the transition activity, n is the number of transitional bits, P_{change} is the contribution of power per bit transition.

Although ESP takes the activity-sensitivity into account to some extent, it does not reflect the bit-positional effects on the power. The influence of all input bit activities on the power dissipation is considered in a cumulative manner.

SPA identifies and solves part of the above problem by a technique called the dual bit type (DBT) approach for datapath elements [3, 18, 19]. The method is based on the fact that there are two breakpoints (BP0, BP1) in the bit-level representation of the temporally correlated, two's complement data stream. The first region between BP0 and the least significant bit is called *data* which contains bits that behave like uniform white noise. The second region between BP1 and the most significant bit is called *signed* which reflects the temporal correlation coefficient of word-level statistics. Hence it is possible to relate the word-level statistics

to bit-level statistics such that two independent sets of capacitive coefficients can be introduced for signed and unsigned (data) parts of the word. Before the power analysis is performed, RT-Level simulation of the design is performed for typical input patterns. During this simulation, the activity of the signed parts and signals in the data parts are viewed and maintained. Additionally, statistical properties (i.e. μ, σ, ρ) of the word-level data are captured. The statistical properties are then used to specify the two break-point bit positions such that the power model for datapath elements uses.

An example of the proposed DBT average power model is [19]:

$$P_{ave} = (N_U \cdot C_U + N_S \cdot C_S) \cdot V_{dd}^2 \cdot f \quad (2.13)$$

where C_U and C_S are the empirically extracted coefficients to characterize the capacitance switched in the data (C_U) and sign (C_S) regions of different functional blocks and N_U and N_S are the number of bits in the data and sign regions, respectively.

The same researchers proposed a different model for the control part of an RTL design [20]. Unlike DBT, activity based control (ABC) model is based on the input-output transition and signal probability. The model for an FSM that is implemented in standard cells is

$$P_{ave} = (C_I \cdot t_I \cdot N_I \cdot N_M + C_o \cdot t_o \cdot N_o \cdot N_M) \cdot V_{dd}^2 \cdot f \quad (2.14)$$

where t_I, t_o are the transition probabilities and N_I, N_o are the number of inputs and outputs respectively. N_M represents the number of on minterms of the truth table.

DBT model is superior to the previous models since it accounts for different input activities. However, the proposed model is valid only for two's complement

data which is common but not a general data representation. Additionally, the characterization of a module requires some knowledge about the internal structure depending on the functionality and complexity which may not always be possible.

Ramprasad et. al. presented an analytical estimation technique for transition activity similar to DBT in the sense that they also used word-level statistics [22]. Their technique differs from DBT in two ways: computation of the break-points BP_0 and BP_1 and computation of the word-level transition activity.

In [21], Gupta and Najm suggested a power macro-model for a combinational circuit based on its input/output signal switching activity. Basically, during characterization, power values are stored in a three dimensional table. The three dimensions of the model are the *average input signal probability*, *average input transition density*, and *average output zero-delay transition density*. In contrast to DBT model, the characterization phase can be completed automatically with a unique model for every functional block.

The above model accounts for the activity in a cumulative manner (i.e. average signal and transition probabilities are considered). This suppresses the bitwise effects on the power dissipation, however each primary input bit may activate different portions of the circuit and cause different amount of power dissipation.

Pedram et. al. introduced bitwise transition effects on cycle power dissipation in [24]. The cycle-based power equation is formed by considering the first-order temporal correlations and third-order spatial correlations. The proposed methodology is based on four steps: Module equation form generation and variable selection, variable reduction, and population stratification.

This method is powerful in the sense that it introduces a variable reduction technique. Otherwise, to obtain a characterization, the required number of coef-

ficients would be the sum of $3N$ (for the first-order terms), $3N(3N-1)/2$ (for the second-order terms), $N(3N-1)(3N-2)$ (for the third-order terms), which is not a small number. On the other hand, the population size which is chosen as 80000 vector pairs is a very small subset of all possible vector pairs ($2^{2N} - 2^N$) for $N=10$ input bits. This model will be revisited in the next chapter.

2.3 Chapter Summary

This chapter summarized the proposed techniques so far in RTL power estimation area. Each technique brings both advantages and disadvantages to the power estimation problem. Predictive techniques and the majority of the descriptive techniques do not model the switched capacitance properly for various input activities such as spatiotemporal correlation of input bits or the bitwise activities. Only DBT [19] accounts for data representation and temporal correlation but it is limited to only two's complement type of data and it does not take the spatial correlation into account. Furthermore, it requires specific knowledge to define the DBT model for each micro-architectural block (i.e. a multiplier and an adder have different models depending on the internal structures). Independent of this thesis, the cycle accurate power model [24] relies on the same idea of modeling bitwise effects on the power dissipation. However, the selected number of vector pair populations is very small compared to the number of exhaustive vector pairs which will be discussed in Chapter 3. The novel model developed in this thesis does not have the above disadvantages and it will be presented in the next chapter.

Chapter 3

CRAB-RPE: Complete-Range Activity-Based RTL Power Estimation

In previous chapters, it was discussed that the amount of work required for RTL power estimation is considerably less than at the lower levels. In Chapter 2, current research in RTL power estimation was investigated and the limitations of each work were discussed. In this chapter, CRAB-RPE, a novel power estimation technique will be presented. The new estimation technique has four features which were not handled simultaneously in any other work.

- CRAB power model is based solely on the bit-level statistics of primary input nodes (PINs).
- No assumption is made about primary input activity. A *complete-range* of bit-level statistics is used for the characterization of each circuit block.
- Second-order terms of the bit-level statistics are used for improved power estimation accuracy and modeling pairwise spatial correlations.
- Temporal correlation of primary input bits is modeled by a lag-one Markov model.

Complete-range summarizes the desire to model the power dissipation for all possible input statistics. After the structural description of an RTL design (see Fig. 2.1) is obtained and simulated, the primary inputs of internal micro-architectural blocks may become spatially correlated because of re-convergent fan-out. It may be that some internal block PINs have low activity while others have high activity or maintain uniform white noise activity. Hence the micro-architectural blocks have to be characterized with a *complete-range* of input activity so that the power can be estimated for any PIN activity.

CRAB-RPE has two main phases which are built upon the CRAB RTL power model:

- CRAB Power Characterization (CRAB-PC)
- CRAB Power Analysis (CRAB-PA)

CRAB-PC and CRAB-PA are depicted in Figs. 3.4, 3.5, respectively. The oval represents either an input to the flow or an output from the flow. The rectangles represent the functional parts of the flow.

CRAB power model and CRAB-RPE phases will be explained in detail in the remaining part of this chapter and the CRAB-RPE features mentioned at the beginning of this chapter will become clearer. The notation used throughout this chapter is summarized in Table 3.1.

3.1 CRAB-RPM: CRAB RTL Power Model

Average power dissipation of a CMOS circuit depends on the nature of applied input vectors [10] or can be defined as the weighted average of cycle-power values. *Cycle power* is the power dissipation caused by the difference of initial and final

Term	Meaning
PIN	Primary Input Node
PON	Primary Output Node
N	Number of PINs per block
L	Length of an input vector set
Switching Prob.	1 to 0 or 0 to 1 transition probability
t_i^{sw}	Switching probability of the i th PIN
t_i^{11}	1 to 1 transition probability of the i th PIN
t_i^{00}	0 to 0 transition probability of the i th PIN
$t_i^{sw} \cdot t_j^{sw}$	Second-order cross-term of the PIN switching probabilities
$k_i^{11}, k_i^{00}, k_i^{sw}$	Model coefficients for $t_i^{11}, t_i^{00}, t_i^{sw}$ respectively
k_{qi}^{sw}	Model coefficient for the quadratic terms $(t_i^{sw})^2$
k_{ij}	Model coefficient for the cross-terms $t_i^{sw} \cdot t_j^{sw}$

Table 3.1: Notation.

state of the input vectors applied to a module. Let us assume that the module has N PINs, hence there are 2^N exhaustive vectors that can be used to test the functionality of the block. Out of 2^N vectors there are $\frac{2^N \cdot (2^N - 1)}{2}$ pair combinations. Since the order of a vector pair is important for power, the total number of exhaustive vector pairs for the module is twice the number of pair sets which is $2^{2N} - 2^N$. One way of estimating the average power dissipation of a block would be to store the cycle-power values for exhaustive vector pairs and retrieve them during subsequent analysis phase. But the number of exhaustive vector pairs exceeds 1,000,000 for 10 PINs. Hence this is not a practical approach for power estimation. Another way would be to characterize the power of a module by using statistical properties such as transition probabilities of the PINs. The transition probabilities of circuit nodes are affected by the spatiotemporal correlations. Spatiotemporal correlation addresses both temporal and local correlations of signals [10, 25]. Researchers have explored ways for estimating switching activity of all circuit nodes

for spatiotemporally correlated signals [8, 25, 26, 28].

In [25], Marculescu et. al. present *switching activity analysis of all circuit nodes* for spatiotemporally correlated signals. The assumptions for the method are zero-delay mode of operation and lag-one Markov Chain Model for the temporal correlation of signals. It is also stated that some constants, so-called *transition correlation coefficients*, can be used for propagation of transition probabilities from primary inputs to any internal or primary output node. In [26], P. Schneider and his co-workers proposed an improved, practical version of Marculescu's work. Mehta et. al. present another *switching activity estimation method* in [27]. The Boolean difference which was first suggested in [8] is used in this paper to define the output *transition density* of a gate in terms of its first-order and higher-order input transition densities.

Three of the above techniques consider the effect of signal correlations on the switching activity of all nodes at the *gate-level* description of digital CMOS circuits. Concurrently with CRAB-RPE technique, Pedram et. al. [28] presented a *cycle-accurate RTL power model* which considers spatial and temporal correlations through third and first order, respectively. The model is based on variable selection, variable reduction and population stratification using linear regression statistics.

The core idea of the CRAB RTL power model originated from the *switching activity analysis* techniques for spatiotemporally correlated signals at the gate-level. Referring to the previous papers [8, 25, 26, 28], it can be easily shown that at the gate-level, any internal or output node transition of a digital circuit can be expressed in terms of its primary input bit-level transitions. However, at the RT-Level of a circuit, internal nodes are not available. But it is known

that each internal node activity depends on each PIN activity. If every internal node and primary output node transition is a function of PIN transitions then **the power dissipation can be completely expressed as a weighted sum of PIN transition probabilities.**

During a clock cycle, a Boolean signal can make only four possible transitions. The four transition probabilities add up to 1.

$$t^{11} + t^{sw} + t^{00} = 1 \tag{3.1}$$

where $t^{sw} = t^{01} + t^{10}$ and superscripts represent the logical transitions.

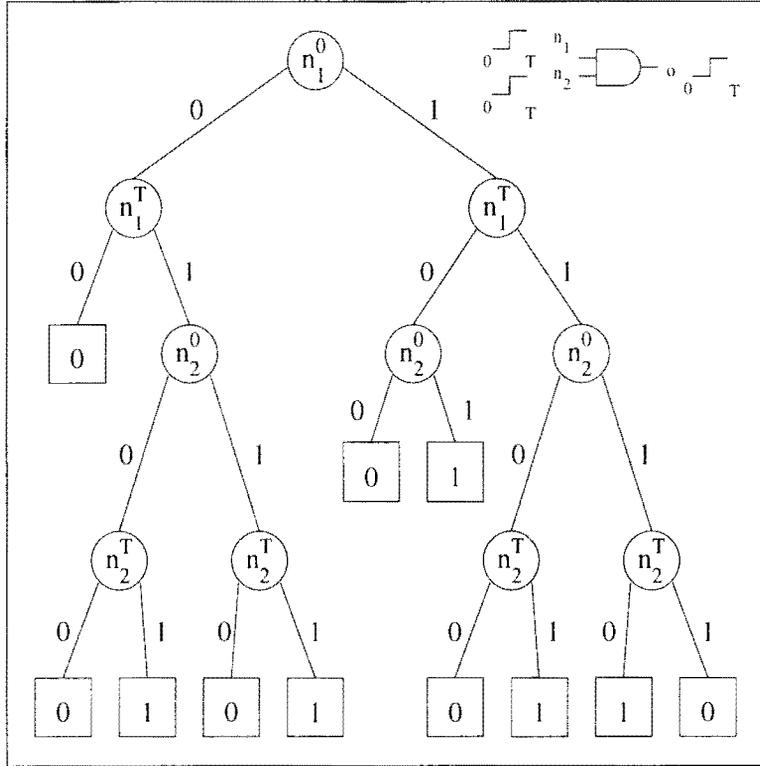


Figure 3.1: BDD of the output switching function of a 2-input AND gate.

To illustrate the dependency of the power consumption on the PIN transition probabilities, Fig. 3.1 shows the Binary Decision Diagram for the output switching

function (SW_o) of a 2-input AND gate. SW_o is 1 if and only if there is a transition from 1 to 0 or 0 to 1 at the output. The PINs are represented inside the bubbles such that superscripts represent the cycle number, and subscripts represent the node identity.

From Fig. 3.1, the output switching probability, t_o^{sw} is obtained in terms of PIN transition probabilities and is shown in Eq. 3.2.

$$t_o^{sw} = t_{n1}^{01} \cdot (t_{n2}^{01} + t_{n2}^{11}) + t_{n1}^{10} \cdot (t_{n2}^{10} + t_{n2}^{11}) + t_{n1}^{11} \cdot (t_{n2}^{01} + t_{n2}^{10}) \quad (3.2)$$

For a Boolean signal t^{01} and t^{10} are approximately the same and equal to $\frac{t^{sw}}{2}$, hence Eq. 3.2 becomes,

$$t_o^{sw} = \frac{t_{n1}^{sw} \cdot t_{n2}^{sw}}{2} + t_{n1}^{sw} \cdot t_{n2}^{11} + t_{n1}^{11} \cdot t_{n2}^{sw} \quad (3.3)$$

where subscripts of probabilities represent the node and superscripts represent the logical values at consecutive cycles (i.e. transitions). For the 2-input AND gate, output switching probability can be determined by evaluating the Eq. 3.3 when the transition probabilities of primary input nodes are known.

As in the 2-input AND gate example ($N=2$), the dependency of internal and primary output node activity on PIN activity can be up to Nth order, where N is the number of PINs. There is a trade-off between practicality and accuracy when the number of terms in the model equation is increased. The development of the CRAB Power Model will be explained with three models, and their relative accuracies will be compared in detail in Chapter 4. The required number of model coefficients for each model is summarized in Table 3.2.

The first order effect of PIN transitions on the average power dissipation of a circuit block is summarized in Eq. 3.4. This is the simplest approximation to the power dissipation that uses the bitwise activity effects. k_i^{00} , k_i^{11} , and k_i^{sw}

Power Model	Number of Model Coefficients
First-order model	3N
First-order + Second-order quadratic terms $((t_i^{sw})^2)$	4N
First-order + Second-order cross-terms $(t_i^{sw} \cdot t_j^{sw})$	$\frac{N^2+5N}{2}$
CRAB (First-order (t_i^{11}, t_i^{sw}) + Second-order cross-terms)	$\frac{N^2+3N}{2}$

Table 3.2: Power models and number of coefficients.

represent the slopes of (P_{ave}, t_i^{00}) , (P_{ave}, t_i^{11}) and (P_{ave}, t_i^{sw}) respectively. As shown in Table 3.2, the number of coefficients needed for this model is 3N.

$$P_{ave} = \sum_{i \in \{PIN\}} (k_i^{00} \cdot t_i^{00} + k_i^{11} \cdot t_i^{11} + k_i^{sw} \cdot t_i^{sw}) \quad (3.4)$$

Second-order effects of PIN activities on the power dissipation are caused by the re-convergent fan-out nodes as shown in Fig. 3.2. According to the figure, the output switching probability is a quadratic function of I2's switching probability. If the depth of any circuit is greater than one then quadratic terms of PIN switching probabilities may appear in the output switching probability function. A model based on this observation is shown in Eq. 3.5. The required number of power model coefficients is 4N, because of the N additional $(t_i^{sw})^2$ terms.

$$P_{ave} = \sum_{i \in \{PIN\}} (k_i^{00} \cdot t_i^{00} + k_i^{11} \cdot t_i^{11} + k_i^{sw} \cdot t_i^{sw}) + \sum_{i \in \{PIN\}} k_{qi}^{sw} \cdot (t_i^{sw})^2 \quad (3.5)$$

Pairwise switching probabilities of different PINs occur more frequently in an output switching probability than quadratic switching probabilities of the same PINs. For the circuit in Fig. 3.2, output switching probability is a function of pairwise cross-terms of two PIN switching probabilities such as $t_{I1}^{sw} \cdot t_{I2}^{sw}$, $t_{I1}^{sw} \cdot t_{I3}^{sw}$, $t_{I2}^{sw} \cdot t_{I3}^{sw}$ which are more in number than the single quadratic term $(t_{I2}^{sw})^2$. Hence, adding

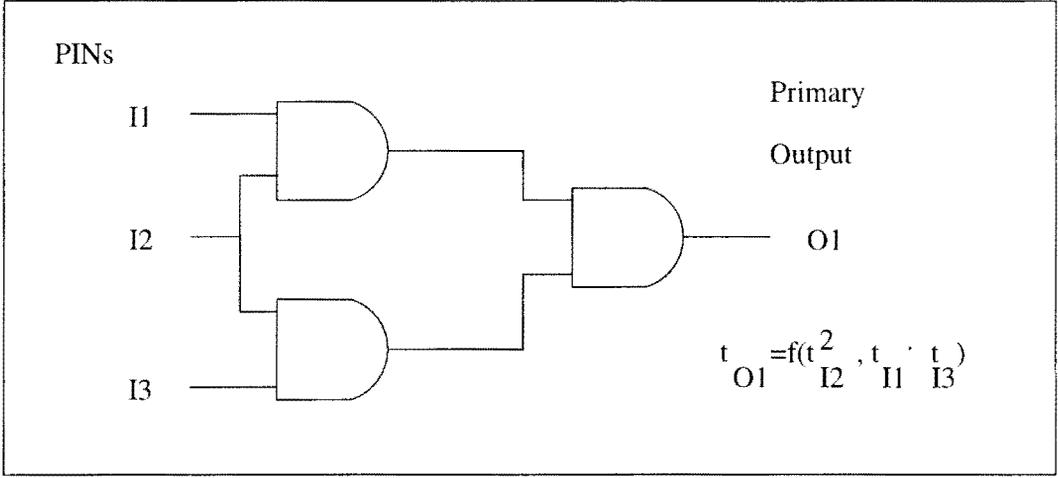


Figure 3.2: Second-order effect of a PIN on a primary output node.

the pairwise cross-terms of PIN switching probabilities is a natural way to increase the flexibility of the model. The third proposed model is

$$P_{ave} = \sum_{i \in \{PIN\}} (k_i^{00} \cdot t_i^{00} + k_i^{11} \cdot t_i^{11} + k_i^{sw} \cdot t_i^{sw}) + \sum_{i \in \{PIN\}} \sum_{\substack{j \in \{PIN\} \\ j \neq i}} k_{ij} \cdot t_i^{sw} \cdot t_j^{sw} \quad (3.6)$$

The total number of coefficients for describing a block has increased to $\frac{N^2+5N}{2}$. Since the typical number of PINs for a micro-architectural block is 32 bits or less, the number of model coefficients would be less than 600. Although this number seems large, the time cost is about a couple of hours for solving the model coefficients with a *linear least square* or other *matrix* solution algorithm. Furthermore, subsequent evaluation of the model equation for any PIN statistics will be reduced to seconds or less.

The final CRAB Power Model is slightly-modified version of the previous model. From Eq. 3.1, $k_i^{00} \cdot t_i^{00}$ are not explicitly included. The CRAB Power Model in words is first order in switching and 1 to 1 transition probabilities, and second order in

pairwise cross-terms of switching probabilities of the PINs. The final model is shown in Eq. 3.7.

$$P_{ave} = \sum_{i \in \{PIN\}} (k_i^{11} \cdot t_i^{11} + k_i^{sw} \cdot t_i^{sw}) + \sum_{i \in \{PIN\}} \sum_{\substack{j \in \{PIN\} \\ j \neq i}} k_{ij} \cdot t_i^{sw} \cdot t_j^{sw} \quad (3.7)$$

From Eq. 3.7 it is seen that, there are N coefficients for each t_i^{11} and t_i^{sw} . The additional second-order term coefficients are $N(N - 1)/2$ in number. The total number of coefficients extracted during the characterization phase is $N(N + 3)/2$.

To illustrate the CRAB power model, let us consider the *2-input AND* gate. Then the model for this gate contains five coefficients as in Eq. 3.8.

$$P_{ave} = k_0^{11} \cdot t_0^{11} + k_0^{sw} \cdot t_0^{sw} + k_1^{11} \cdot t_1^{11} + k_1^{sw} \cdot t_1^{sw} + k_{01} \cdot t_0^{sw} \cdot t_1^{sw} \quad (3.8)$$

Temporal correlation of each PIN is embodied in the $t_0^{11}, t_0^{sw}, t_1^{11}, t_1^{sw}$ terms and spatial correlation of the two PINs is included in the $k_{01} \cdot t_0^{sw} \cdot t_1^{sw}$ product. This equation resembles the previous Eq. 3.3 because the cross-term $t_0^{sw} \cdot t_1^{sw}$ is present in both. The other two products in Eq. 3.3 are approximated by weighted sum of first-order transition probabilities.

3.2 CRAB-PC: CRAB Power Characterization

The CRAB-PC process involves the extraction of power model coefficients for each micro-architectural block in an RTL library. To explain the CRAB-PC phase, closed-form Eq. 3.7 for the CRAB power model is expressed in matrix form as

$$T \cdot k = P \quad (3.9)$$

The matrix representation of the model is depicted in detail, in Fig. 3.3 where left superscripts of the matrix elements represent the run number, right superscripts stand for the transitions (11 or sw), and the subscripts represent the PIN number.

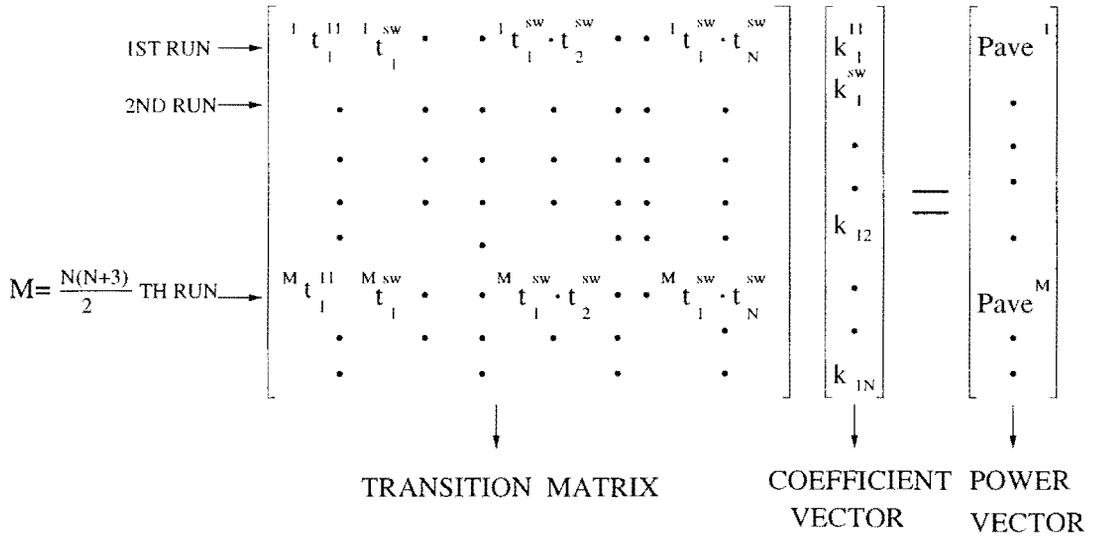


Figure 3.3: Matrix representation of the CRAB power model for the characterization phase.

The transition matrix (T) is formed from bit-level probabilities, and has $N(N+3)/2$ columns. The number of rows of T and the computed power vector (P) is set by the number of power characterization experiments. Each micro-architectural component is characterized only once but requires at least $\frac{N(N+3)}{2}$ simulations to complete. Therefore, in the experiments reported in Chapter 4, the row dimension of the transition matrix is $N(N+3)/2$ or larger.

As depicted in Fig. 3.4, CRAB-PC has three main functional parts, input vector generator (IVG), model coefficient extractor (MCE), and a lower-level power estimation tool. For each micro-architectural block in the library, a gate-level description is a required input to CRAB-PC phase. A power library for the gate-level technology used is also needed for the CRAB-PC phase. The RTL library

output contains model coefficients k_i^{11} , k_i^{sw} , k_{ij} of length $N(N+3)/2$ for each micro-architectural block, computed by the MCE. The micro-architectural blocks and the model coefficients are retrieved from the RTL Library during CRAB power analysis which is depicted in Fig. 3.5.

CRAB-PC

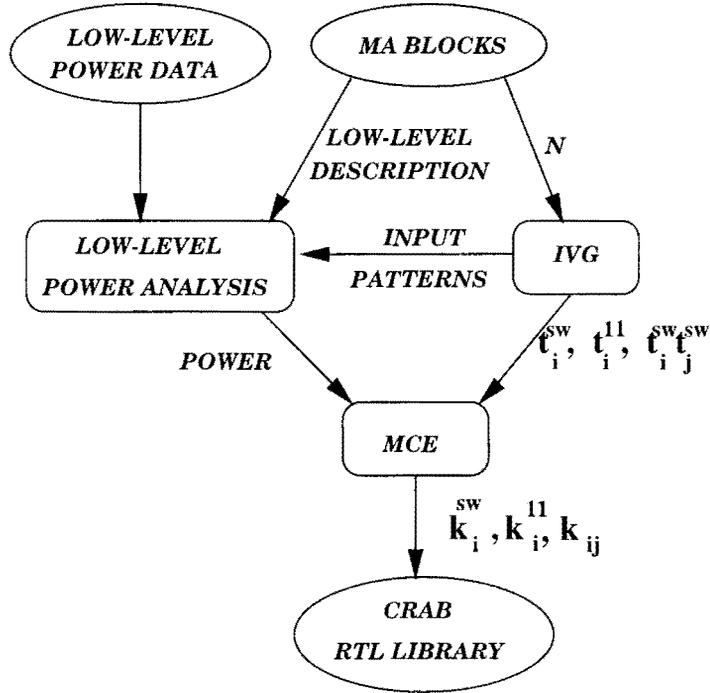


Figure 3.4: CRAB Power Characterization phase.

3.2.1 IVG: Input Vector Generator

The IVG part of the characterization flow generates primary input vectors with desired bit-level statistics for micro-architectural blocks for each characterization run. The number of input vector pairs used during characterization is selected at this point. This phase defines the run by run PIN transition probabilities, t_i^{00} , t_i^{11}

and t_i^{sw} used during $N(N + 3)/2$ simulations.

The inputs to IVG are the length of the stimulus to be generated and a control file including the PIN statistics. The length of the stimulus per run is selected based on statistical measures that will be clarified in Chapter 4. The choice of the number of runs or the number of stimuli depends on the number of model coefficients. Since the number of coefficients in a CRAB power model is $N(N + 3)/2$, the number of characterization runs required for well-characterization of a circuit block is at least $N(N + 3)/2$. Some supplementary code generates the control files for a block's characterization.

3.2.2 MCE: Model Coefficient Extractor

The MCE part of the characterization flow extracts power model coefficients k_i^{11} , k_i^{sw} , k_{ij} . This step follows IVG and power simulation. Extraction of the model coefficients is done by solving the linear system which is depicted in Fig. 3.3. The power vector is of length $N(N + 3)/2$ or more which is composed of *gate-level power estimates* obtained by simulation. The transition matrix is formed by post-processing the probability files that include t_i^{00} , t_i^{11} and t_i^{sw} . The transition matrix has $N(N + 3)/2$ columns and same number or more rows depending on the number of runs.

Singular Value Decomposition (SVD) is the preferred method to solve for under-determined, over-determined or square systems of linear equations. The main part of MCE is the SVD code which computes the model coefficients.

3.2.3 Power Simulator

A power simulator can be either a transistor-level (e.g. IRSIM) or a gate-level simulator (e.g. Quickpower) that provides power estimates during characterization process. The power estimates from the simulation are the ones in Fig. 3.3 and used in MCE.

The simulator takes the lower-level (transistor or gate-level) description of each micro-architectural block as an input. Another requirement for the simulator is a power library for the technology used in the lower-level descriptions.

The simulator will be also used for the verification of the power evaluation of a sub-block of RTL design in Chapter 4.

3.3 CRAB-PA: CRAB Power Analysis

Power analysis is the process which results in a power estimate for an RTL design. As shown in Fig. 3.5, high level synthesis, power evaluation of design sub-blocks and the simulator are the three main functional parts of CRAB-PA. IVG can be used at this phase if no other stimulus is available to the user.

3.3.1 HLS: High Level Synthesis

HLS is a part of power analysis flow which converts RTL design to sub-blocks. The composition of sub-blocks is also called as the *structural RTL representation* of the design. Synthesis of the design can be done by either a high-level synthesis tool or manually. Previously in Chapter 2, structural RTL representation of a chip has been depicted in Fig. 2.1. The sub-blocks of the design are mapped to RTL library components and the related model coefficients are retrieved for power evaluation

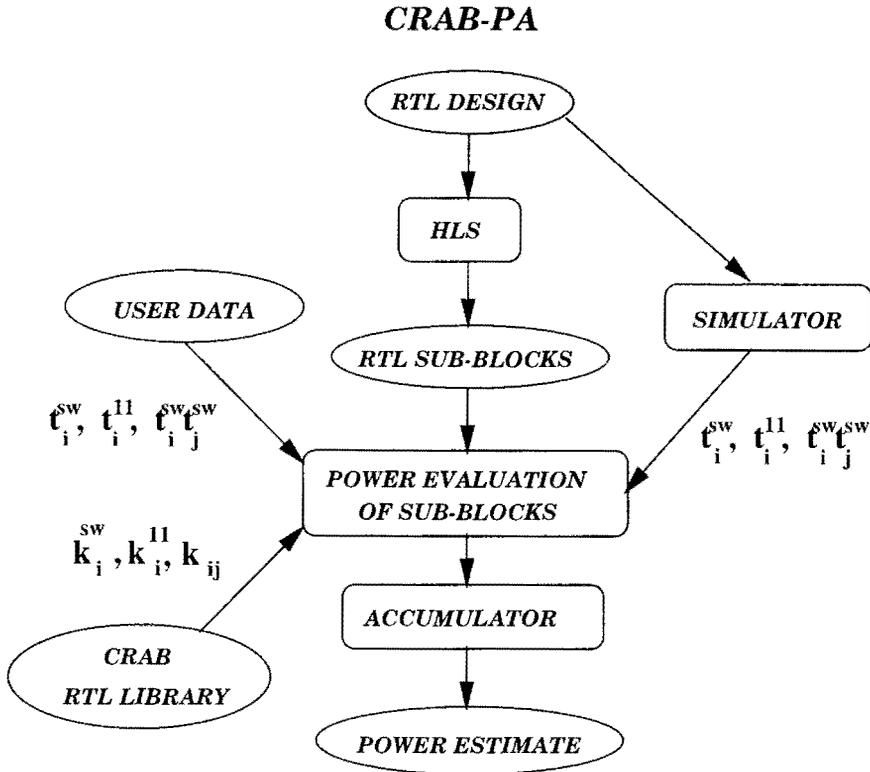


Figure 3.5: CRAB Power Analysis phase.

of each component. If there is no corresponding component in the library, then the sub-block can further be synthesized until it is described by the components in the RTL library. If the sub-block can be neither synthesized nor mapped to the library then a predictive model could possibly be used to estimate the average power dissipation.

The verification of the model accuracy for each micro-architectural block is the first aim in this thesis. The experiments presented in Chapter 4 do not include HLS, because they focus on the CRAB-PC of a micro-architectural block and CRAB power evaluation of the same block.

3.3.2 PE: Power Evaluation

The PE part of power analysis returns a power contribution for a sub-block in the RTL design as depicted in Fig. 2.2 (Chapter 2) and specifically in Fig. 3.5. Power evaluation of design sub-blocks is done by retrieving the model parameters from the RTL library and collecting the switching, and 1 to 1 transition probabilities of the PINs from the simulator or the user. If the user provides PIN activity of the RTL design, the power evaluation of internal RTL sub-blocks must propagate the primary input transition probabilities to sub-blocks outputs. The transfer coefficients from input statistics to output statistics can be solved by replacing the power vector with output statistics in Fig. 3.3, however this is not included in this thesis. User-based power analysis is potentially faster than simulation-based power analysis as far as the propagation functions can be defined for each primary output of a module. The RTL library may not always contain the modules that HLS selects. In that case, as discussed in Chapter 2, predictive models for RTL power estimation can be used.

3.3.3 Simulator

The simulator in CRAB-PA phase is used to simulate the synthesized RTL design to obtain the PIN bit-level statistics. If a test bench with the RTL description of the design is not available, IVG can be used to generate the input stimulus. If the user provides specific information about PIN activity, there is no need to invoke the simulator, because the only required data for the power evaluation of each micro-architectural block is PIN transition probabilities. For example, the user can define *the typical PIN activity as uniform white noise*, then based on

the knowledge that a uniform white noise signal has $t^{sw}=0.5$ and $t^{11}=0.25$ power can be evaluated immediately for the sub-blocks with these inputs. PIN activity could then be propagated to other sub-blocks' input nodes such that the power contributions of each block can be evaluated.

3.4 Limitations and Extensions

Although CRAB-RPE will be shown to return reliable, accurate results for some ISCAS benchmarks and common circuits, several extensions can be done for further improvement and practicality.

First, CRAB-RPE was verified for solely combinational blocks. It may further be improved for sequential and control parts of an RTL design. No experiments have been carried out in this regard, hence it is unknown if CRAB-RPM can give satisfactory results for these blocks.

Second, the number of terms in the CRAB-RPM is $N(N + 3)/2$ which may increase significantly for blocks that have more than 32 PINs. Although CRAB-PC is done once for each block, the large number of terms increases the time cost at this phase. Hence reduction of terms in the CRAB-RPM can be a way to improve practicality. Reduction of number of input vector pairs can be another way of decreasing the time cost in the CRAB-PC phase.

Third, usage of the CRAB-RPM for parameterizable blocks is a practical concern. Since the model is based on PIN bit-level statistics, it does not include number of PINs as a term. Hence, the model can be improved if it could handle parameterization of PIN numbers.

Fourth, all the experiments in Chapter 4 are conducted ignoring the HLS part

in Fig. 3.5. Lack of an HLS tool in the experimental environment was the first reason for this. The integration of an HLS tool to the CRAB-PA phase would be helpful to experiment with larger RTL designs.

3.5 Chapter Summary

This chapter introduced a novel RTL power estimation technique called CRAB-RPE (Complete-Range Activity-Based RTL Power Estimation) which has two main phases: CRAB-PC (Complete-Range Activity-Based Power Characterization) and CRAB-PA (Complete-Range Activity-Based Power Analysis).

CRAB-RPE is based on a new model called CRAB-RPM (Complete-Range Activity-Based RTL Power Model). CRAB-RPM does not make any assumption about PIN activity where previous models were based on certain PIN activity such as uniform white noise. In other words, CRAB-RPM is independent of either *numerical representation* (e.g. 2's complement) or *spatiotemporal correlations* of the data. CRAB-RPM is based on the first-order and second-order PIN transition probabilities. First-order terms are based on 1→1 and switching (1→0 or 0→1) probabilities of PINs. Second-order terms are based on the pairwise cross-terms of different PIN switching probabilities.

The coefficients of the first and second-order terms in the CRAB-RPM ($k_i^{11}, k_i^{sw}, k_{ij}$) are stored in the CRAB-PC phase. During the CRAB-PA phase, PIN statistics ($t_i^{11}, t_i^{sw}, t_i^{sw} \cdot t_j^{sw}$) are required from either the simulator or the user. The returned value from CRAB-PA phase is the average power dissipation of the RTL design with the stated PIN statistics.

Chapter 4

CRAB Model Evaluation

In previous chapters, the background for the RTL power estimation techniques was established and the novel CRAB model was introduced. Although power dissipation is pattern dependent, many researchers considered using only uniform white noise (UWN) as the PIN data or equivalently 0.5 PIN switching probabilities. Only the DBT approach [3, 19] is based on a temporally correlated 2's complement data representation which is different from uniform white noise. Other than the references given in [3], no source has been found on the bit-level behavior of data in realistic settings such as two's complement, sign magnitude or floating point. Even if the PIN signals of an RTL design are uncorrelated, the internal RTL design sub-blocks may have spatially and temporally correlated data streams at their inputs, since these blocks do not interface to PINs of the RTL design and their inputs may become correlated because of re-convergent fanout. These correlated data streams may have bit-level activities different from UWN such that higher and lower order bits may be highly active and the bits in the center region may be uniform white noise (which is completely different pattern from DBT data). Hence for different data representation and correlations, a complete-range activity-based power model is required. In this work, the sampling of the complete range of a PIN switching activity has been considered and the CRAB technique is implemented based on

this notion.

The proposed CRAB power model was built for several modules including the ISCAS combinational circuits and experiments were performed for the verification. Note that the coefficients of each power model is dependent on the cycle period, in other words, they have the units of *Watts*. For the experiments, the same cycle period is used for both CRAB-PC and CRAB-PE phases, so the period (or frequency) component of the coefficients is same for two phases and can be ignored. However, changing the units of coefficients from *Watts* to *Joules* is scaling by the elapsed time so that the energy is modeled instead of power.

In this chapter, the experimental setup and some application specifics are presented. The experimental results for models proposed in addition to the CRAB power model are introduced and four models are compared and discussed. Experimental results for each circuit using the CRAB power model are presented in detail. The power model accuracy is compared to lower level power simulators and discussed.

4.1 Experimental environment

In this section, the CRAB-PC and CRAB-PA phases (introduced in Chapter 3) are revisited specifically, the tools used, modifications and strategies for each are discussed in detail. Since the aim for the experiments is to verify the accuracy of CRAB power model with respect to lower-level power estimates for different modules, the high level synthesis (HLS) part is omitted in the CRAB-PA phase, hence only CRAB-PE is shown in Fig. 4.8 on page 55. The experimental flow for the CRAB-PC and CRAB-PE phases are in Figs. 4.1 and 4.8, respectively. The

circuits used for the verification of the model are shown in Table 4.1.

Circuit Name	Circuit Function	Total Gates/Transistors	Number of PINs	Number of PONs
NUR.ALU	ALU	254 Transistors	10	5
MAGCMP	Magnitude Comp.	22 Gates	8	3
ADD4	4 bit adder	15 Gates	8	4
C17	Arbitrary	6 NAND Gates	5	2
C432	Priority Decoder	160 (18 EXOR) Gates	36	7
C499	Error Correction	202 (104 EXOR) Gates	41	32
C1908	Error Correction	880 Gates	33	25
C6288	16-bit Multiplier	2406 Gates	32	32

Table 4.1: Circuits for the experiments.

4.1.1 CRAB-PC Implementation

In this section, the implementation of each part in the CRAB-PC phase is discussed in detail and is shown in Fig. 4.1.

Simulator Environment

Power Simulator in the original CRAB-PC phase (Fig. 3.4) was replaced by two low-level power simulators: 1. Quickpower, a gate-level power analysis tool from Mentor Graphics [32, 33]. 2. IRSIM, a transistor-level power analysis tool [35].

IRSIM is used only once for the characterization of the original (first-order) model (Chapter 3). It is an event-driven simulator and it combines the node capacitance and the output activity of each node in the circuit when calculating the average dynamic power.

The rest of the circuits were characterized by using Quickpower in QuickHDL environment. Quickpower is run with an event-driven simulator (QuickHDL in

CRAB-PC

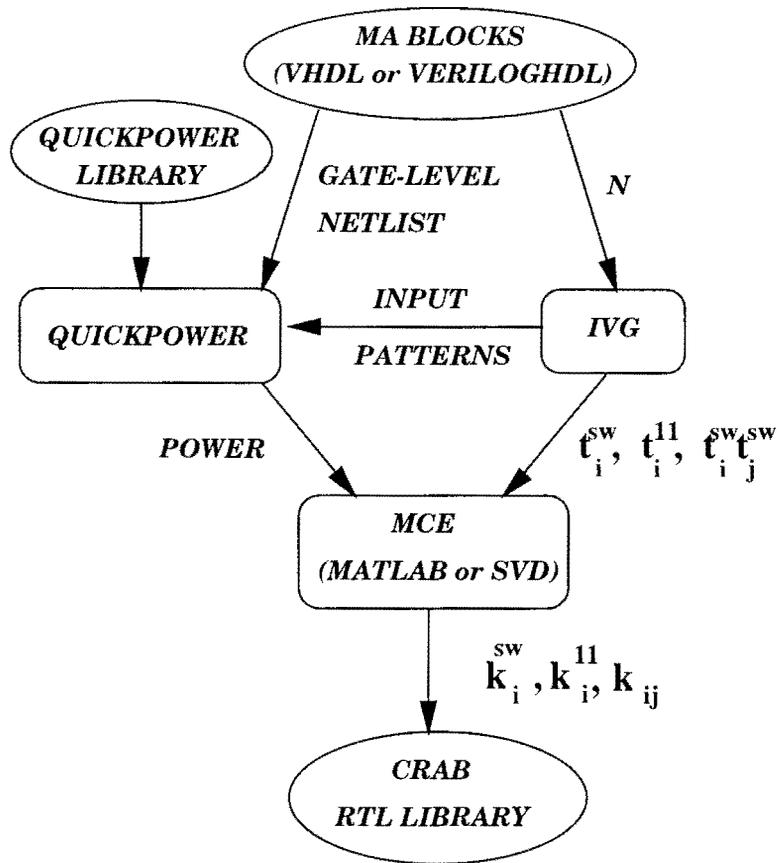


Figure 4.1: Implementation of CRAB-PC.

this case) to monitor switching activity. For power analysis, it uses a gate-level power library characterized with respect to varying input slew rates and output capacitances. For the experiments, a Quickpower library targeted to a 0.8μ CMOS technology was used. The contents of the library is as follows: 2 inverters, 2-3 input AND, 2 input NAND, 2-3 input OR, 2 input NOR, 2-input Multiplexer, a buffer, a tri-state buffer, a D Flip-Flop, and a latch. Quickpower also requires a gate-level description of each micro-architectural block either in VHDL or VerilogHDL

format. If only an RTL representation of the circuit is available, it has to be synthesized to the desired gate-level library. For this purpose, Mentor Graphics Autologic Synthesis Tool was used.

Micro-architectural Blocks

ISCAS benchmarks are the majority of the circuits that were used as examples of the micro-architectural blocks. For the CRAB technique, either the Quickpower library for the present ISCAS library package or RTL descriptions for ISCAS benchmarks were needed. Fortunately, RTL descriptions of ISCAS-89 combinational circuits in VHDL format are in the public domain. Minor modifications on the RTL descriptions were required on the RTL descriptions in order to synthesize them using the 0.8μ CMOS gate-library. First, “library selfext” declaration part was deleted. Second, the structural or gate-level declaration part was deleted. Finally, “fout” term in all Boolean-logic declarations was deleted. An illustration of the VHDL code for C17 before and after the modifications are in Figs. 4.2 and 4.3, respectively.

C17, C432 and C499 were modified as explained above, Autologic was used for synthesis. They were successfully synthesized to the 0.8μ CMOS gate-library. On the other hand, RTL descriptions of C1908 and C6288 could not be synthesized through Autologic, possibly because of the tool’s capability limits. Hence their gate-level netlists were used by mapping the ISCAS gate-library to the 0.8μ CMOS gate-library. However several gates (e.g. NAND5) in the ISCAS gate-library do not match with any gate in the 0.8μ CMOS. Hence those gates were redefined in terms of the gates in 0.8μ CMOS and replaced in the gate-level netlists of C1908 and C6288.

```

library IEEE;
use IEEE.std_logic_1164.all;
library selfext;
use work.gates_pkg.all;
use work.fflop_pkg.all;
ENTITY c17_i89 IS
PORT (
INP: in std_ulogic_vector(0 to 4);
OUTP : out std_ulogic_vector(0 to 1);
H : in std_ulogic
);
END c17_i89 ;
ARCHITECTURE structural OF c17_i89 IS
signal INTERP : std_ulogic_vector(0 to 3):=(others=>'0') ;
signal OUTPI : std_ulogic_vector(OUTP'range):=(others=>'0') ;
NAND0 : NANDG_N generic map (2,1 ns,1 ns)
port map (
inp(0) => INP(0),
inp(1) => INP(2),
out1 => INTERP(0));
.
.
.
ARCHITECTURE rtl OF c17_i89 IS
signal INTERP : std_ulogic_vector(0 to 3):=(others=>'0') ;
signal OUTPI : std_ulogic_vector(OUTP'range):=(others=>'0') ;
BEGIN
REGVECT : BLOCK (H='1' AND NOT H'STABLE)
BEGIN
END BLOCK ;
NAND6 : INTERP(0) <= NOT(fout,INP(0) AND INP(2)) after 1 ns;
.
.
.

```

Figure 4.2: c17.vhdl before modification.

```

library IEEE;
use IEEE.std_logic_1164.all;
ENTITY c17_i89 IS
PORT (
INP: in std_logic_vector(0 to 4);
OUTP : out std_logic_vector(0 to 1);
H : in std_logic
);
END c17_i89 ;
ARCHITECTURE rtl OF c17_i89 IS
signal INTERP : std_logic_vector(0 to 3):=(others=>'0') ;
signal OUTPI : std_logic_vector(OUTP'range):=(others=>'0') ;
BEGIN
REGVECT : BLOCK (H='1' AND NOT H'STABLE)
BEGIN
END BLOCK ;
NAND6 : INTERP(0) <= NOT(INP(0) AND INP(2)) after 1 ns;
NAND7 : INTERP(1) <= NOT(INP(2) AND INP(3)) after 1 ns;
NAND8 : INTERP(2) <= NOT(INP(1) AND INTERP(1)) after 1 ns;
NAND9 : INTERP(3) <= NOT(INTERP(1) AND INP(4)) after 1 ns;
NAND10 : OUTPI(0) <= NOT(INTERP(0) AND INTERP(2)) after 1 ns;
NAND11 : OUTPI(1) <= NOT(INTERP(2) AND INTERP(3)) after 1 ns;
BUFFER_OUT : OUTP <= OUTPI;
END rtl ;

```

Figure 4.3: c17.vhdl after modification.

Another micro-architectural block, NUR.ALU was designed by using MAGIC [38]. Since the transistors are the building units of this circuit, a transistor count is depicted in Table 4.1 as a metric for the size. Hence, IRSIM (a transistor-level power simulator) is used for the CRAB-PC phase. This circuit was used to evaluate the original model (first-order model) proposed in Chapter 3. The results of the experiments will be presented in the following sections.

The rest of the circuits (MAGCMP, ADD4) in Table 4.1 are in either VHDL or VerilogHDL form. The synthesis of these RTL circuits is done by using Autologic and targeting to the 0.8μ CMOS library.

IVG Implementation

There are two important issues that need to be considered in the IVG part of CRAB-PC. One is the generation of *complete-range* input statistics and the other is the selection of input vector length.

Complete-range summarizes the desire to characterize power dissipation for all possible input statistics. As discussed in previous chapters most models are based on a *uniform white noise* activity assumption where $t_i^{11}=0.25$ ($1\rightarrow 1$ probability) and $t_i^{sw}=0.5$ (switching probability). For a single PIN, that is a point in a three-dimensional space where t_i^{11} and t_i^{sw} are the x-y coordinates respectively and the z-coordinate is the average power of the circuit as illustrated in Fig. 4.4. The aim of the CRAB PIN statistics generation is to choose three points of t_i^{sw} from three regions to model the effect of the PIN's *low*, *high* and *uniform white noise* activity. These are depicted as LA, HA and UWN in Fig. 4.4. The average power values in Fig. 4.4 are from Quickpower results for one PIN of ADD4 benchmark. A similar figure applies to all other PINs of ADD4. For the complete-range the

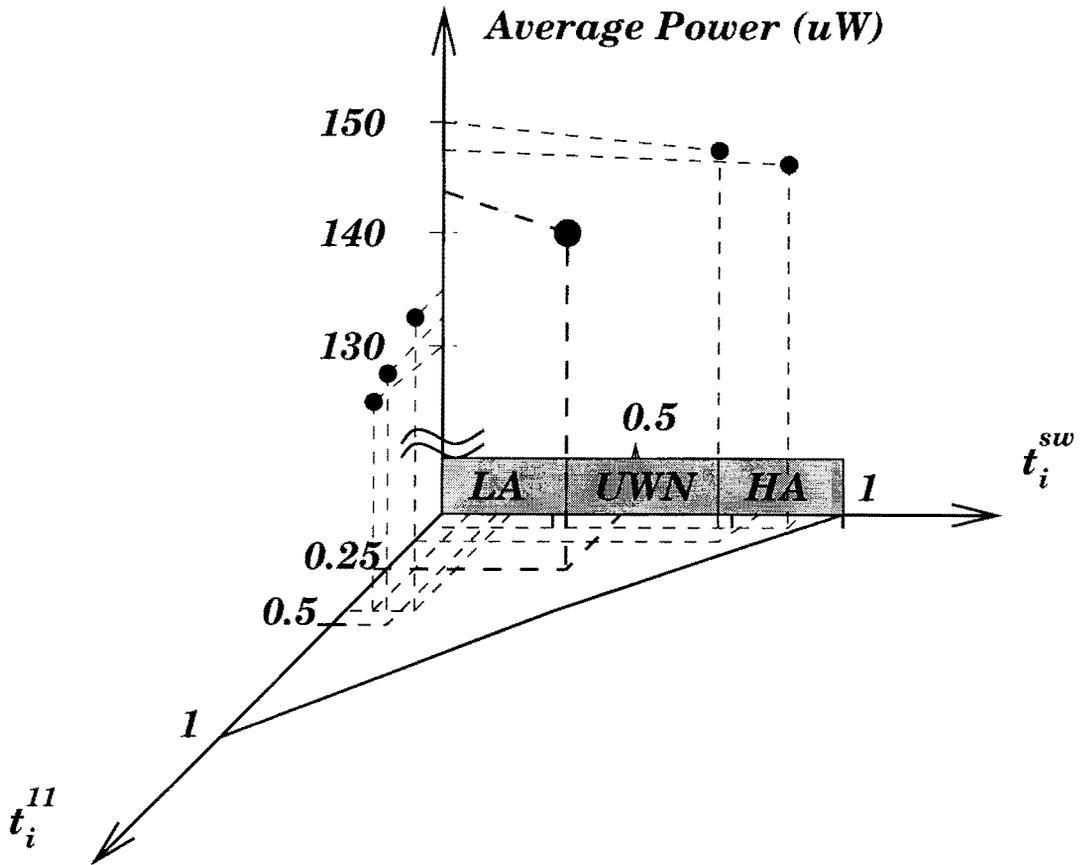


Figure 4.4: Illustration of average power dependency on t_i^{sw} and t_i^{11} .

average power is a point in a $2N+1$ dimensional space.

The PINSTAT algorithm for generating PIN statistics for the vector sets is in Fig. 4.5. The vector sets generated with these PIN statistics are used in the CRAB-PC phase. The aim of the PINSTAT algorithm is to generate $N(N+3)/2$ or more different PIN statistics that span the $2N$ -dimensional activity space $(t_0^{11}, t_0^{sw}, t_1^{11}, t_1^{sw}, \dots, t_{N-1}^{11}, t_{N-1}^{sw})$. NMB (Number of Multiple Bias) represents the number of different selections of 2 to $N-1$ PINs from N PINs. For example, if 2 PINs are to be biased, NMB pair combinations out of N PINs are selected, in other words the number of vector sets for the 2-PINs biased to LA (low activity) is NMB.

PINSTAT(N,LA,HA)

1. Set NMB (number of vector sets for multiple (2 to N-1) biased PINs)
2. Set NSB (number of vector sets for single biased PIN)
3. For both LA and HA
 - 3.1 Bias N PINs at the same time.
 - 3.2 For NM = 2 to N-1 PINs
 - 3.2.1 Bias NMB different combinations of NM PINs.
 - 3.3 Bias NSB number of single PINs.

Figure 4.5: PINSTAT Algorithm for the specification of PIN statistics.

NSB (Number of Single Bias) is the number of sets where different single PINs are biased to desired activity. For majority of the experiments, NSB is N which means each single PIN is biased to the desired activity.

According to the PINSTAT algorithm, two activity points, one from LA and one from HA regions are used for the selected PINs. If any PIN is not biased by an activity point, it will be assumed to be distributed by uniform white noise. Thus different combinations of the PINs are biased to activities in LA, HA or UWN range by PINSTAT. The exhaustive number of all possible PIN bias points

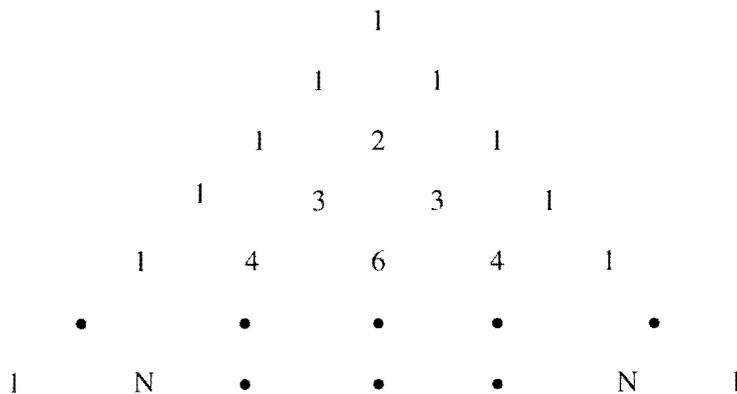


Figure 4.6: Pascal Triangle.

Number of PINs Biased with LA and HA	Number of pattern sets
1	2·NSB
2	2·NMB
3	2·NMB
⋮	⋮
N-1	2·NMB
N	2

Table 4.2: Number of pattern sets for each PIN bias.

is 2^N . This can be found by adding the number of single, pair, triple up to N combinations of N PINs in a Pascal Triangle Row (Fig. 4.6) as in Eq. 4.1 which is equivalently the sum of the binomial coefficients.

$$\begin{aligned}
 NE = & 1 + 2N + 2N(N-1) + 2N(N-1)(N-2) + \dots \\
 & + 2N(N-1)(N-2) \cdots (N - \lfloor \frac{N}{2} \rfloor)
 \end{aligned} \tag{4.1}$$

The summation does not include the 0th combination of N PINs since it has no practical meaning. First term (i.e. 1) in Eq. 4.1 is the number of all PINs biased at the same time with a desired activity. Second term ($2N$) is the sum of the number of combinations for both one PIN and N-1 PINs set to a transition activity different than all other PINs. The final term represents the number of combinations for $\lfloor \frac{N}{2} \rfloor$ PINs biased.

For N PINs, the number of vector sets that are generated with respect to PIN-STAT algorithm is shown in Table 4.2. For the reported experiments, single PIN combinations and all PINs are biased with desired activity, however the remaining multiple PIN biases are selected samples from the exhaustive combinations. Therefore, the number of combinations (NMB) is less than the exhaustive PIN

combinations. The second column of Table 4.2 is always $N(N+3)/2$ or more. For a very small N the exhaustive number of combinations and $N(N+3)/2$ are close to each other. An example of the PINSTAT algorithm is illustrated for $N=3$ in Fig. 4.7. The all possible combinations of the biased PINs are shown in this figure where $NMB=3$, $NSB=3$ and the total number of PIN statistics to be generated is 14. The MCE would require at least 9 of this set.

BIT POSITION			
2	1	0	
UWN	UWN	LA	LA = 5% switching prob. HA = 95% switching prob. UWN=50% switching prob. Single PIN Bias with LA and HA NSB=3
UWN	LA	UWN	
LA	UWN	UWN	
UWN	UWN	HA	
UWN	HA	UWN	
HA	UWN	UWN	
LA	UWN	LA	Double PIN Bias with LA and HA NMB=3
LA	LA	UWN	
UWN	LA	LA	
UWN	HA	HA	
HA	UWN	HA	
HA	HA	UWN	
LA	LA	LA	Triple PIN Bias with LA and HA
HA	HA	HA	

Figure 4.7: The PIN statistics generated by PINSTAT(3,5%,95%).

On the other hand, the number of input vectors in a vector set applied to a circuit module was chosen such that the standard deviation (σ) from the weighted power (wp) was within the reasonable bounds of 3% or less of the average value (see Table 4.3). The standard deviation is calculated by the following:

Stim. Stat.	wp(μ W)	$\sigma(500)$	$\sigma(750)$	$\sigma(1000)$	$\sigma(1250)$	$\sigma(2500)$
5_10	130.94	2.38	3.16	0.80	2.24	0.34
5_15	131.75	2.07	1.30	0.94	3.19	0.42
5_64	142.24	3.20	2.16	2.05	1.15	0.10
5_85	144.23	0.37	1.39	0.52	2.02	0.73

Table 4.3: In the first column, the number at the left of underscore represents the bit position and the number at the right of underscore shows the percentage of the PIN switching activity.

$$\sigma = \sqrt{(P_L - wp)^2} \quad (4.2)$$

where $wp = \frac{\sum_L L \cdot P_L}{\sum L}$, L represents the vector length, and P_L stands for the average power related to L number of input vectors. A statistical description of the vector set is also shown in the first column of Table 4.3. The maximum standard deviation in each row is not more than 2.5% of wp which implies that the number of vectors 500 to 2500 can be used for the experiments. Table 4.3 illustrates the strategy that was used for the selection of vector length (L). The results of previous work for the selection of input vector length (when the bits are uniform white noise) can be found in [23, 29]. In [29], the required vector length to guarantee a specified accuracy ($\epsilon=0.1, 1 - \delta=0.9$) for most of the ISCAS benchmarks is shown to be not more than 2000 except C6288.

MCE Implementation

MCE part in the CRAB-PC phase (Fig. 4.1) was implemented by using MATLAB's matrix or least-square solution facilities. SVD code from [30] was modified to replace MATLAB after majority of the experiments. SVD was preferred for the matrix solution algorithm because of its *under-determined*, *over-determined* and *square* matrix solution capability.

Depending on the implementation of the MCE, both the probability files from IVG and power values from Quickpower were post-processed. The main strategy for post-processing was to build the transition matrix shown in Chapter 3. Since the probability files from IVG contain only the first-order PIN probabilities, the second-order pairs were required to be computed and placed in the transition matrix. The set of power values were also gathered as the right-hand-side power vector in Chapter 3.

4.1.2 CRAB-PE Implementation

As noted before, the HLS part in the original CRAB-PA phase was omitted for these experiments. Therefore, only the implementation of CRAB-PE is shown in Fig. 4.8. This phase of the CRAB technique has considerably lower time cost compared to the CRAB-PC phase.

In this phase, the input vector sets generated by IVG are completely different than those created for the CRAB-PC phase. During the power evaluation phase, the CRAB power model is evaluated by using the model coefficients from CRAB-PC and the transition probabilities from IVG. To verify the model for these new vector sets, Quickpower was used to obtain the gate-level power values. The CRAB power estimates were compared with the power values from Quickpower by using a *relative error* measure. However, as it will be explained later, the *absolute error* measure will also be considered during the discussion of some of the results. These measures are based on the power estimates from CRAB-PE (P_{ave}) and Quickpower (QP) as defined in Eqs. 4.3 and 4.4.

$$\text{Absolute Error} = |P_{ave} - QP| \quad (4.3)$$

$$\text{Relative Error} = \frac{|P_{ave} - QP|}{QP} \quad (4.4)$$

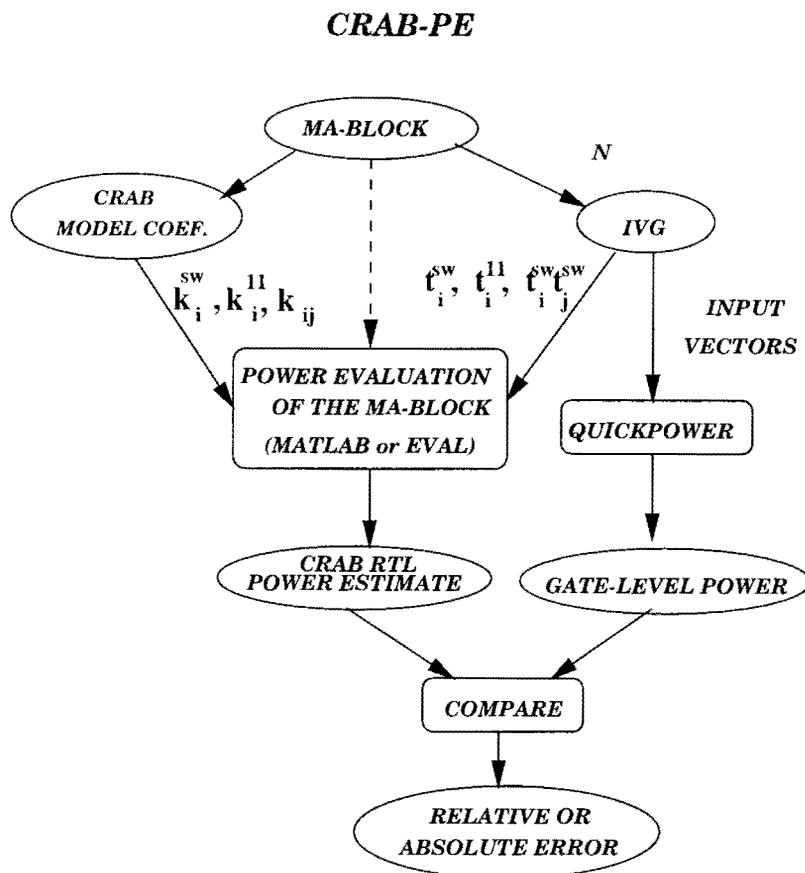


Figure 4.8: Implementation of CRAB Power Evaluation.

4.2 CRAB Experimental Results

In this section, the results of the three models proposed in addition to second-order CRAB model will be introduced and discussed. For the evaluation of the original (first-order) model, the circuits *MAGCMP*, *NUR.ALU*, *ADD₄* were used

(refer Table 4.1). *C17* was used to compare the results of four CRAB models presented in Chapter 3. The rest of the results for *C432*, *C499*, *C1908*, and *C6288* use the second-order CRAB power model.

4.2.1 NUR.ALU Results

NUR.ALU is a transistor (layout) level ALU circuit. The total number of data input bits (A and B) is eight and the number of control bits is two. The simulations were carried out using IRSIM. The *first-order original model* was evaluated using NUR.ALU.

NUR.ALU was first characterized with low PIN activity (LA=15%), high PIN activity (HA=85%) and the uniform white noise PIN activity. A thousand vectors were generated for each PIN's statistics. MCE used the direct matrix solution facility of MATLAB. The model coefficients extracted during CRAB-PC range from -8.90 ($k_{A(1)}^{00}$) to 17.85 ($k_{A(2)}^{sw}$). The relative error (compared to IRSIM power values) during the CRAB-PE phase for input vector sets different than those selected in CRAB-PC is depicted in Fig. 4.9.

The notation for the data activity which appears in the x-axis of the graph is as follows: The four values after B or A specify the percent of switching activities of an ascending order of bits where H stands for one hundred percent and U stands for uniform white noise. For example, B0HH0AHH00 means that B(0), B(3), A(2), A(3) are set to zero switching probability and the rest of the bits are set to one hundred percent activity. If there is only one number after B or A it means all the bits of A or B are biased with the same activity.

As it is seen in Fig. 4.9, the relative errors in the center of the graph jump to 20-40%. This set of experiments intentionally biased the input bits with activities that

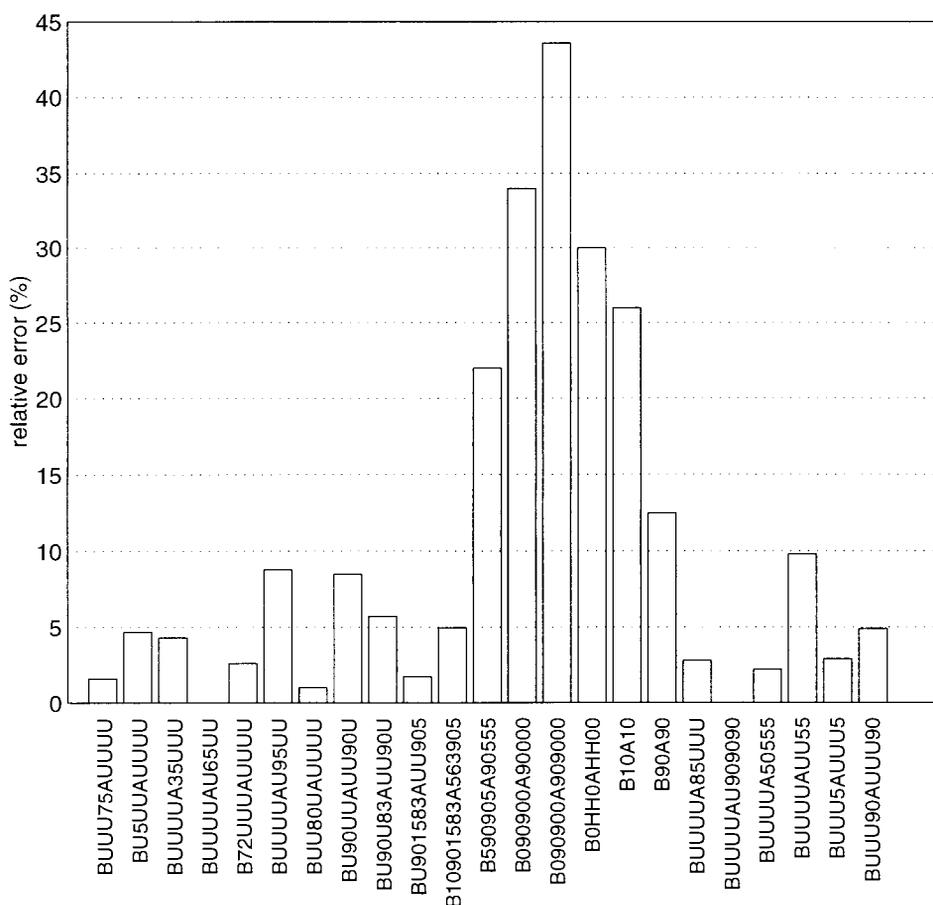


Figure 4.9: U stands for uniform white noise, H stands for 100% switching activity. Other values are percent switching activity.

amplified the model coefficients such that the relative errors would be increased. These test cases do not represent the typical PIN activity of an ALU, however they may occur inside other RTL structures. Hence the model for the ALU was validated at PIN activities that force differences of large values of coefficient and activity products. The other vector sequences applied to the circuit resulted in under 10% relative errors.

To demonstrate the capabilities of the first-order CRAB power model, it was

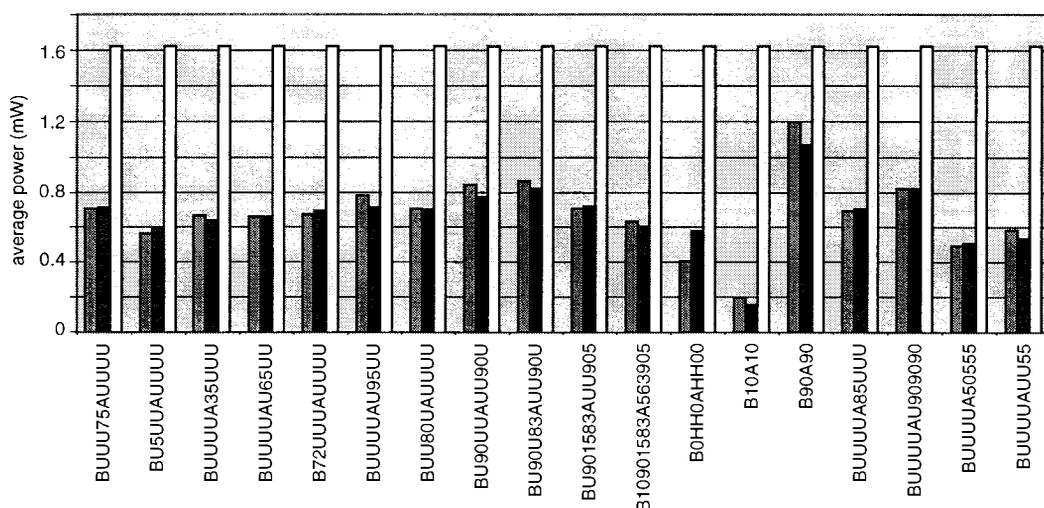


Figure 4.10: U stands for uniform white noise, H stands for 100% switching activity. The lightest of the triples shows CES estimates, the darkest shows IRSIM results and leftmost of the triple shows the first-order CRAB power estimates.

compared with two previous techniques (DBT and CES) discussed in Chapter 2. CES is a complexity-based predictive technique which does not take the data activity into account. DBT is an activity-sensitive, descriptive technique which models solely two's complement correlated data activity. The rightmost two bars in Fig. 4.9 shows that the CRAB model relative error for DBT-like input activity is under 5%. *DBT-like input activity* means the higher order (or sign) bits of the inputs A and B are biased with either very low activity (LA=5%) or very high activity (HA=90%) while the other lower bits are left as uniform white noise. The comparison of CRAB with CES results is shown in Fig. 4.10. The y-axis indicates the average power value in μW . The x-axis represents the different biased input activities. For example, the 13th triple-bars were obtained for all PINs biased with low activity (LA=10%). The lightest bars (or the rightmost of a triple) indicate the results obtained with CES, the center of the triple bars represent the IRSIM

power values and the leftmost of the triples are the estimates obtained with the CRAB model. Clearly, CRAB tracks the input activity effects on the power well, on the other hand, CES errors are as much as 800% of IRSIM values (for the 13th triples).

These experiments demonstrate that the first-order CRAB model reproduces the IRSIM power values for a wide range of activity. However, it was also observed that, for the LA bias of PINs the relative errors are as high as 25%. The insignificance of this error on power estimation is better understood by *absolute error* measure. As it is seen in Fig. 4.9, when all PIN activities are biased to LA, the IRSIM power values are very small thus missing 25% of a small amount is ignorable. In other words, the absolute error for all PIN statistics are in the same order, but for LA bias of PINs the denominator of the relative error (i.e. IRSIM power value) is very small. Hence the CRAB power estimates track the IRSIM results with reasonable absolute error bound.

4.2.2 MAGCMP Results

The properties of the MAGCMP are shown in Table 4.1 on page 43. This circuit was characterized by using Quickpower as the low-level power simulator. The vector length was 1000 as in NUR.ALU experiments and the CRAB algorithm was based on LA=15% and HA=85%. The square transition matrix was formed and the direct matrix solution facility of MATLAB was used to solve for the model coefficients. The extracted CRAB model coefficients for this circuit range from -129.71 ($k_{B(0)}^{sw}$) to 152.73 ($k_{B(2)}^{00}$).

The results for the CRAB-PE phase of this micro-architectural block are shown in Fig. 4.11. The relative errors, between 20% and 30%, were caused by: 1. When

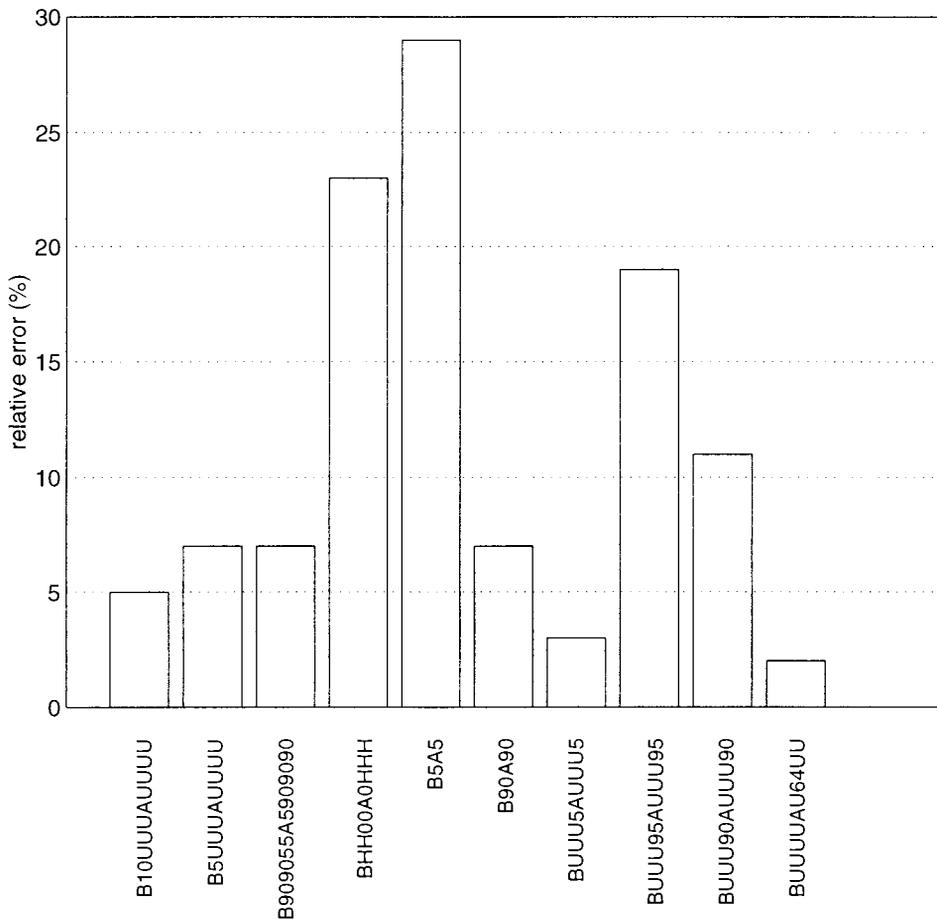


Figure 4.11: Relative error vs. PIN statistics for the CRAB-PE of MAGCMP.

all the PINs were biased with very low activity ($LA=5\%$) and **2**. When the PIN statistics were biased such that the extremum differences were amplified. For the remainder PIN statistics, the relative error is under 10% except for the case of biasing of higher order PINs with very high activity (B(3) and A(3) are set to $HA=95\%$).

MAGCMP experiment was another step towards building the CRAB model for any data-path element and any simulation environment. Although NUR.ALU and MAGCMP are represented at different levels (transistor and gate-level respec-

tively) and were characterized by using different power simulators (IRSIM and Quickpower respectively), they produced similar large relative errors in the same range of PIN statistics (e.g all PINs biased with LA).

4.2.3 ADD4 Results

The ADD4 block has eight PINs. It was characterized by using Quickpower and 51 sets of vectors of length 1000 with single PIN and all PINs and pair PINs (DBT) biases ranging from 5% to 95%. MATLAB's least square solution (LSS) facility was used to solve for the CRAB model coefficients. The first-order CRAB model coefficients range from -3.38 ($k_{A(2)}^{00}$) to 3.91 ($k_{B(3)}^{sw}$).

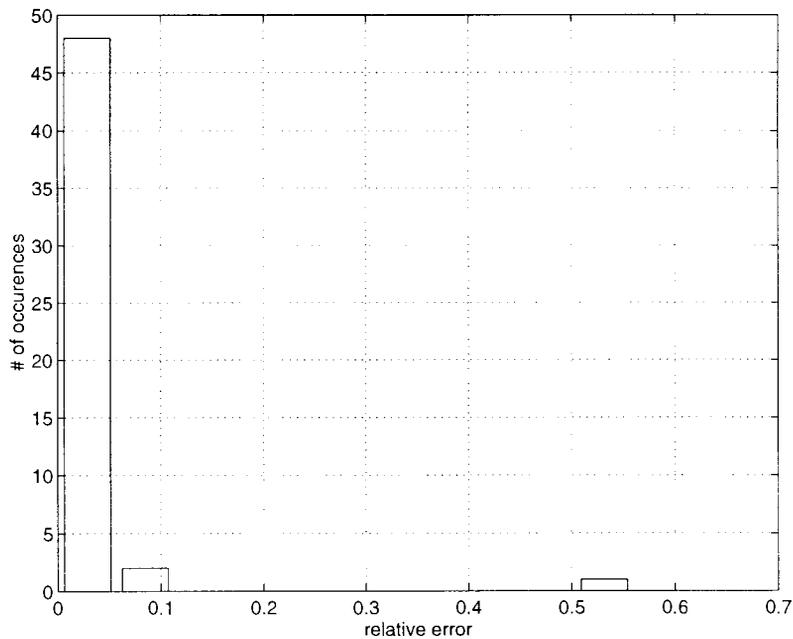


Figure 4.12: Relative error distribution for the LSS of ADD4 CRAB coefficients.

Fig. 4.12 shows the results of the LSS for the CRAB-PC phase. Since the number of vector sets used is 51, a histogram of the relative errors are shown.

Only one out of 51 solutions has 50% relative error which is the power obtained by all PINs biased with LA=5%. The rest of the solutions have under 10% relative error.

CRAB-PE phase was implemented with two different sets of PIN activity. For the first evaluation phase, only a single PIN or pair of PINs were biased with activities ranging from 5% to 85%. The total number of vector sets for these activities was 48. The histogram of the relative error of CRAB power model with respect to Quickpower values is depicted in Fig. 4.13. Relative errors for three of 48 runs are between 10% and 25% and 45 of the relative errors are under 10% . The first-order CRAB model appears to work well for single and pair PIN biases. Using these PIN statistics, the first-order model estimates the power for pairwise correlations as well as the single PIN temporal correlation.

The second evaluation phase used 36 new vector sets. This time the used vector sets that bias three PIN statistics (with LA and HA) in addition to the single and pair PIN biases. The relative error distribution of CRAB power model with respect to Quickpower values are depicted in Fig. 4.14 where 5 out of 36 relative errors have values between 10% and 18% and 31 of them resulted under 9%.

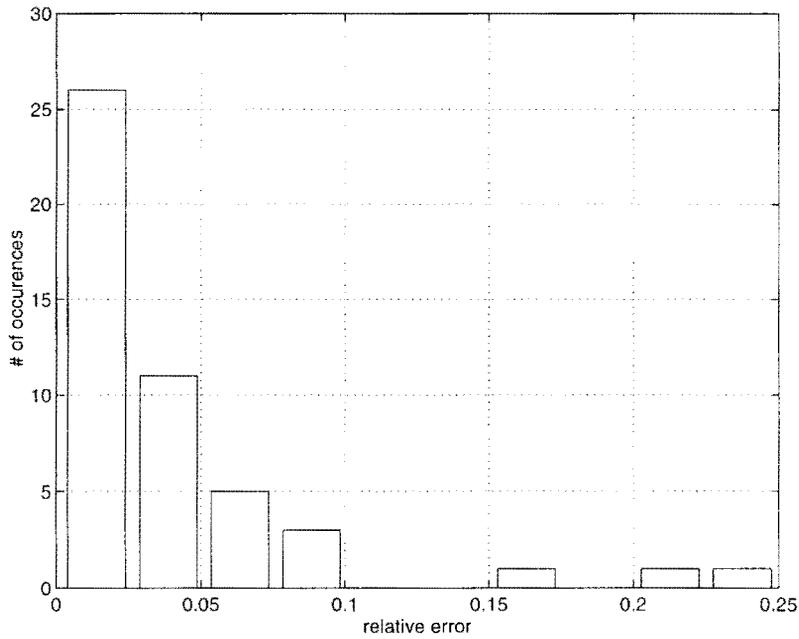


Figure 4.13: Relative error distribution of 48 CRAB-PE results for ADD4 with the original model.

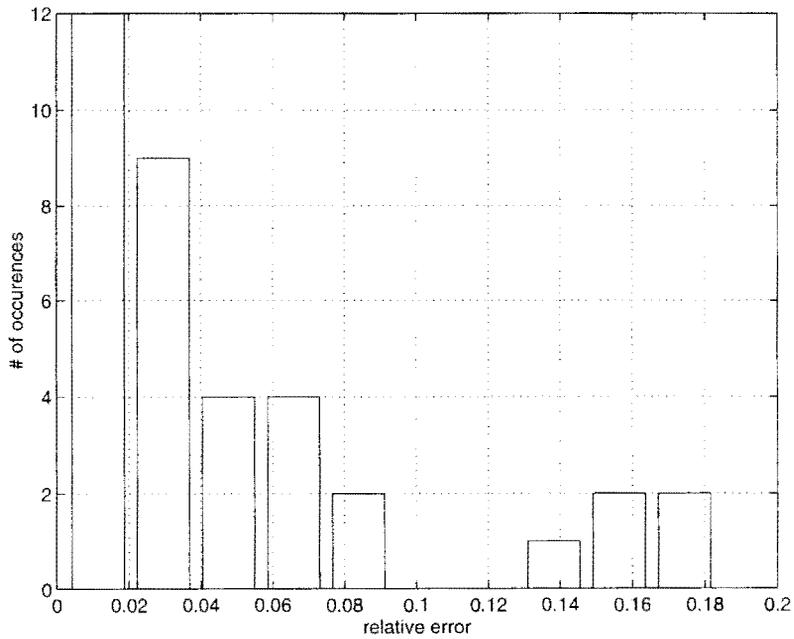


Figure 4.14: Relative error distribution of 36 CRAB-PE results for ADD4 with the original model.

Hence the first-order CRAB power model also provides reliable results for three biased PINs. Three biased PINs appear as the third-order product terms in the power equation, and these terms are error term for the first-order model. In summary, the first-order model predicts the second-order and third-order power contributions and return power estimations with small relative errors compared to Quickpower estimates.

4.2.4 Comparison of the Proposed Models with C17

The previous results showed that the first-order model provides power estimates with no more than 30% relative errors when all PINs are biased with very low activity. In order to improve the low activity range, the second-order terms were added to the original model as discussed in Chapter 3. In this section, the development of this CRAB model and a comparison of all four models will be presented for an ISCAS combinational circuit, C17.

To compare the proposed models directly, the same set of input vectors were used for CRAB-PC and CRAB-PE phases. However, the vector sets were completely different in each phase. For the characterization phase, 53 vector sets of length 1000 were generated in an attempt to cover the range of bit-level statistics (1,2,...N combinations of PINs are biased) at LA and HA. During the CRAB-PC phase, the coefficient vectors for the four different models were extracted by using the least square solution (LSS) facility in MATLAB during the CRAB-PC phase. The number and range of the four model coefficients are shown in Table 4.4. The histograms of the relative errors of the four models are shown in Figs. 4.15, 4.16, 4.17 and 4.18.

The CRAB relative errors decreased significantly from 120% to 8% of Quick-

Model	Number of Coefficients	Minimum Coefficient	Maximum Coefficient
First-order	15	$k_1^{00} = -0.465$	$k_0^{sw} = 0.66$
Quadratic Model	20	$k_1^{00} = -0.894$	$k_2^{sw} = 1.09$
First-order+Cross-terms	25	$k_4^{11} = -0.436$	$k_2^{sw} = 0.564$
CRAB second-order	20	$k_{3,4} = -2.78e-05$	$k_2^{sw} = 4.10e-5$

Table 4.4: The range and the number of four model coefficients.

power. It can be also observed that the inclusion of pairwise cross-terms had more effect on the results than the quadratic terms. This result was suggested in the earlier analysis in Chapter 3 when it was observed that re-convergent fanout for two PINs occurs more frequently than a single PIN. If the numbers of coefficients differ for two models then improvements in power estimate may be because the degree of freedom in the $2N+1$ dimensional space would be increased. Since the number of terms for each model is 20 and the same vector sets are used in CRAB-PC and CRAB-PE phases, the results of these models can be directly compared.

For the CRAB-PE phase, 41 sets of input vectors of length 1000 were generated. These vector sets were completely different than the characterization stimulus sets in terms of PIN statistics. The model coefficients extracted in the CRAB-PC phase were used to estimate the power for each model and the histogram of the relative errors are shown in Fig. 4.19, 4.20, 4.21, 4.22. As before, the y-axis shows the “number of occurrences” and the x-axis shows the “relative error ranging from 0 to 1”. As it is clear from the figures, the relative error for all PINs biased with LA decreased from 80% to 12% for first-order through pairwise CRAB models, respectively.

For both CRAB-PC and CRAB-PE phases, it was observed that the final proposed CRAB model provided significantly improved power estimates (for all

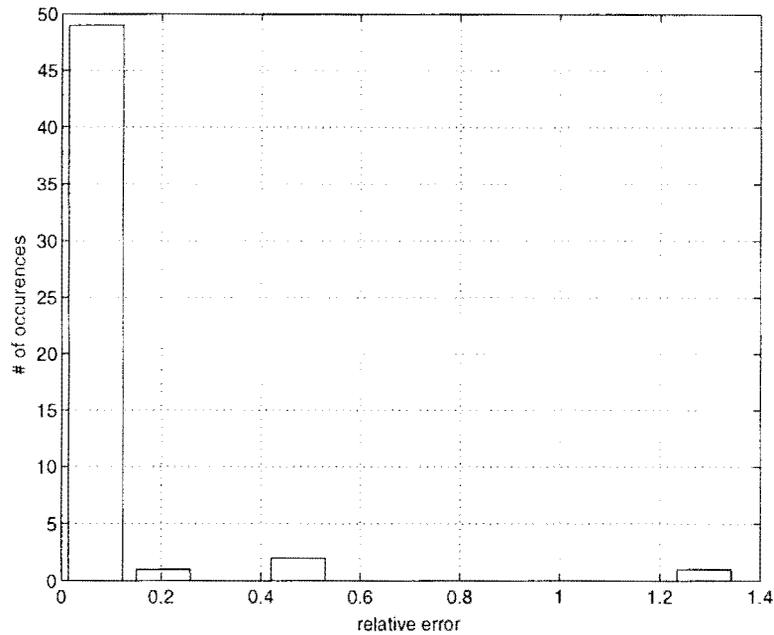


Figure 4.15: Relative error distribution of the LSS for C17 with the original model.

types of PIN statistics) when compared to the results of the first, second and third CRAB models. On the other hand, the fourth CRAB model requires a large number of vector sets than the previous two models. Because power model accuracy (compared to Quickpower) for all PIN activities was preferred to the CRAB-PC time cost, the fourth CRAB power model is used for the power estimation of remaining ISCAS circuits.

Many researchers have assumed uniform white noise to build their models, and they used random (uniform white noise) data to evaluate them. To examine the performance of the CRAB second-order model for uniform white noise data, 10 vector sets of length 2000 were generated for additional CRAB-PE phase. The results of this experiment are shown in Fig. 4.23. As seen, all the tests resulted in relative errors in comparison to Quickpower of less than 3.5%.

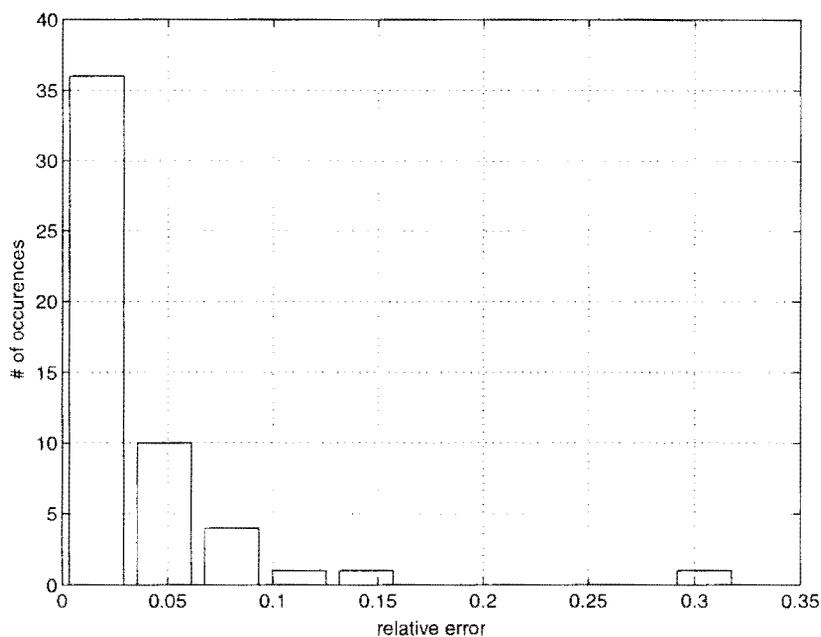


Figure 4.16: Relative error distribution of the LSS for C17 with the quadratic model.

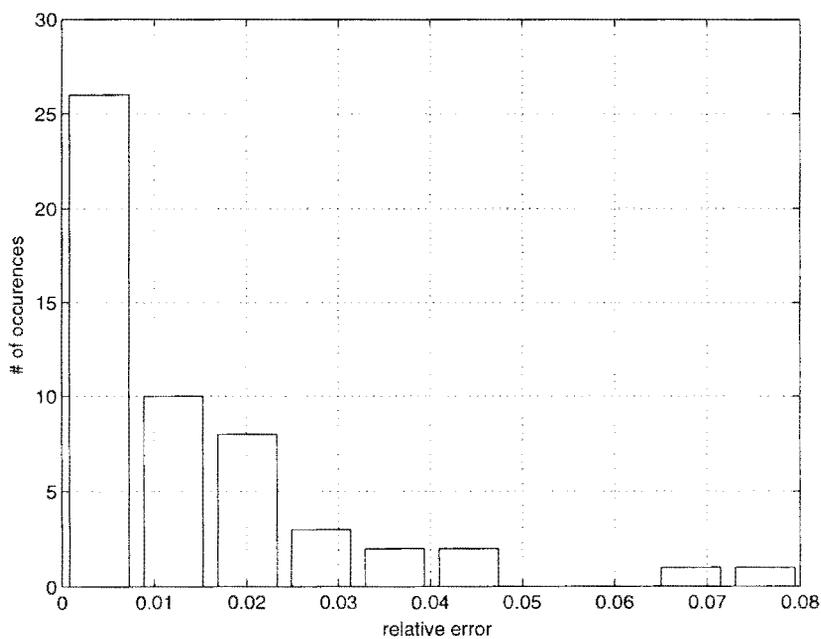


Figure 4.17: Relative error distribution of the LSS for C17 with the third model.

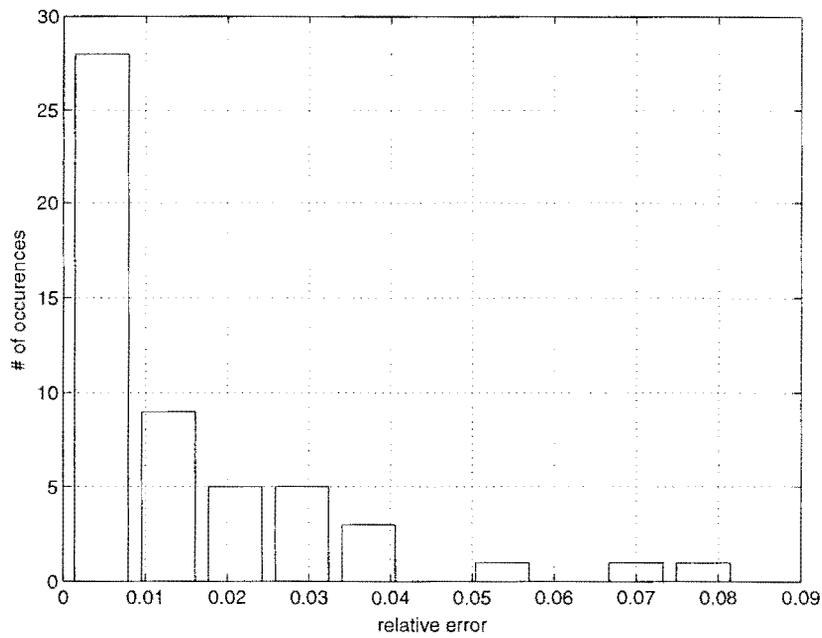


Figure 4.18: Relative error distribution of the LSS for C17 with the CRAB model.

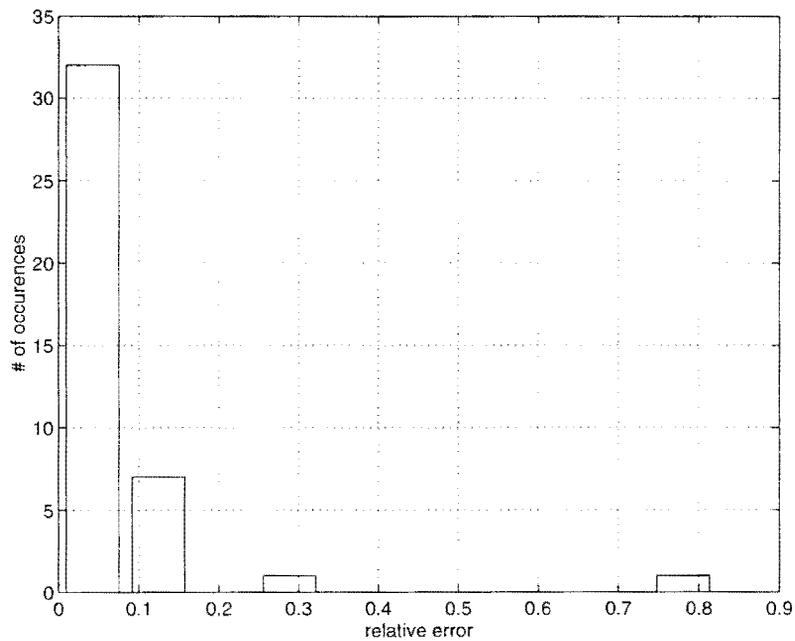


Figure 4.19: Relative error distribution of 41 CRAB-PE results for C17 with the original model.

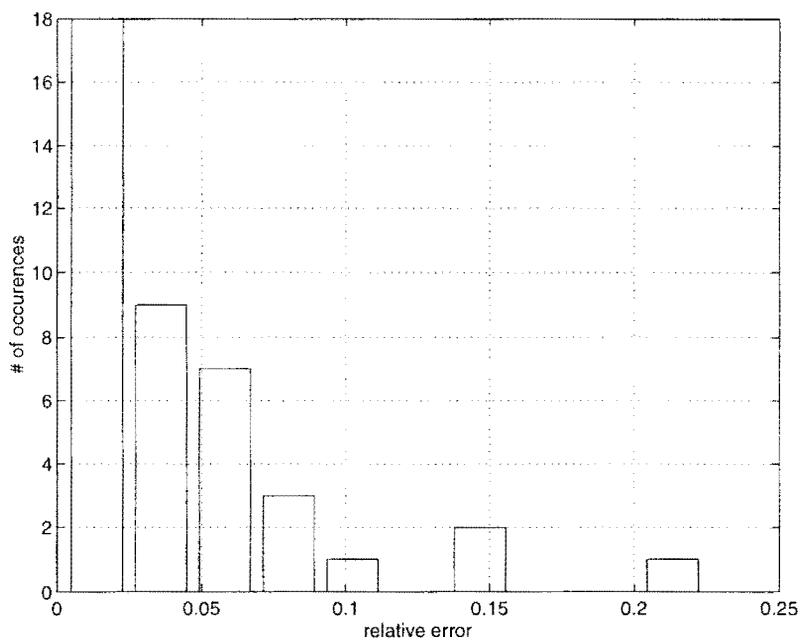


Figure 4.20: Relative error distribution of 41 CRAB-PE results for C17 with the quadratic model.

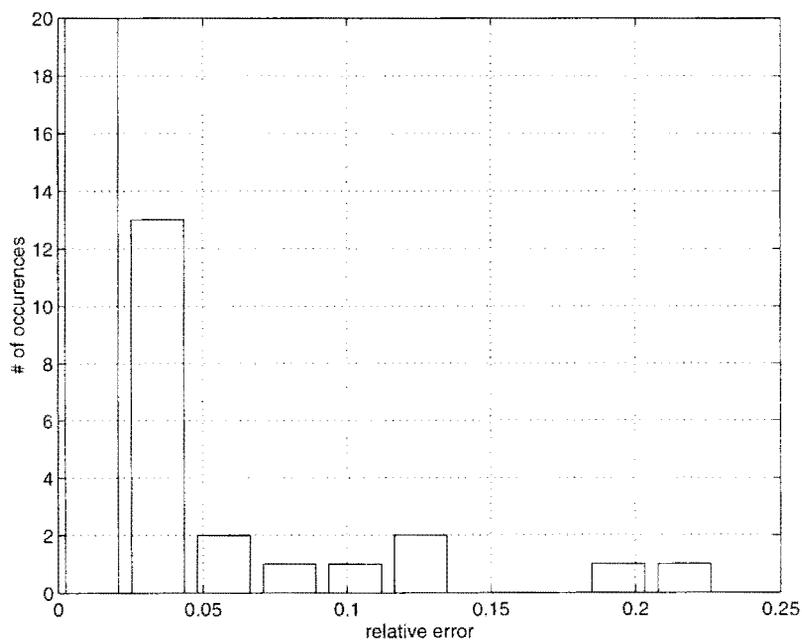


Figure 4.21: Relative error distribution of 41 CRAB-PE results for C17 with the third model.

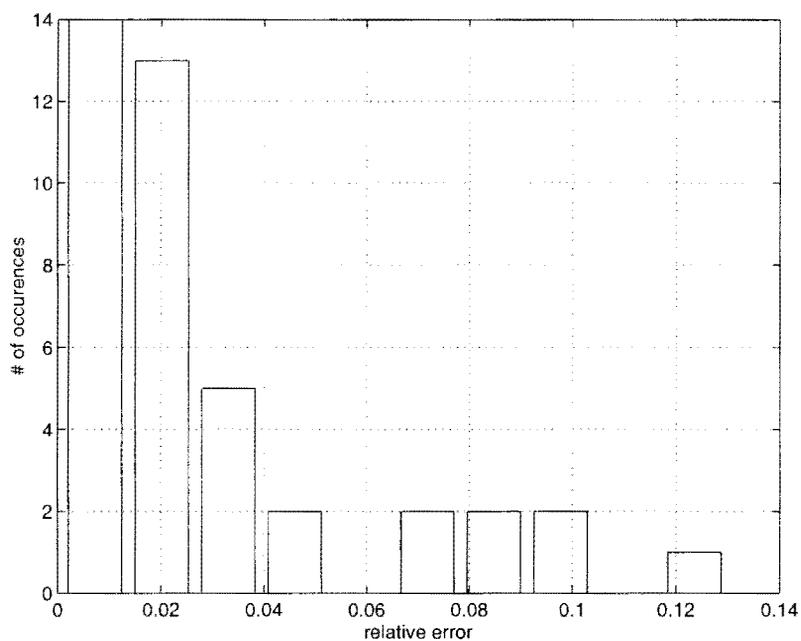


Figure 4.22: Relative error distribution of 41 CRAB-PE results for C17 with the CRAB model.

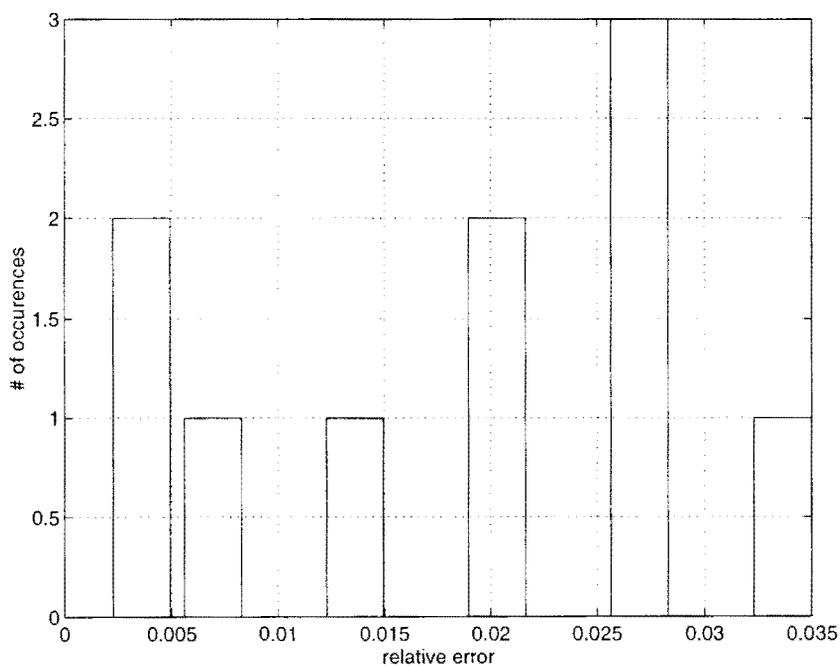


Figure 4.23: Relative error distribution of 10 CRAB-PE results using random (uniform white noise) data for C17.

4.2.5 C432 Results

Since C432 has 36 PINs, at least 702 vector sets (i.e. PIN statistics) are required for the CRAB-PC phase. For the experiments reported in this thesis, 754 vector sets of length 2000 were used. The *complete-range* PIN statistics for both the CRAB-PC and CRAB-PE phases were generated by `PINSTAT(36,5%,95%)`. The model coefficients for this circuit were obtained using MATLAB's LSS capability. The coefficients of the CRAB model range from $-4.99\text{e-}04$ ($k_{14,23}$) to $5.33\text{e-}04$ ($k_{22,23}$). The verification of the LSS for this circuit is shown in Fig 4.24. Clearly, the relative error for the LSS is under 1% for over 700 vector sets and results for 30 more have less than 7% relative error. Hence 95% of the same vector sets applied in the CRAB-PC phase provide power estimates under 1%.

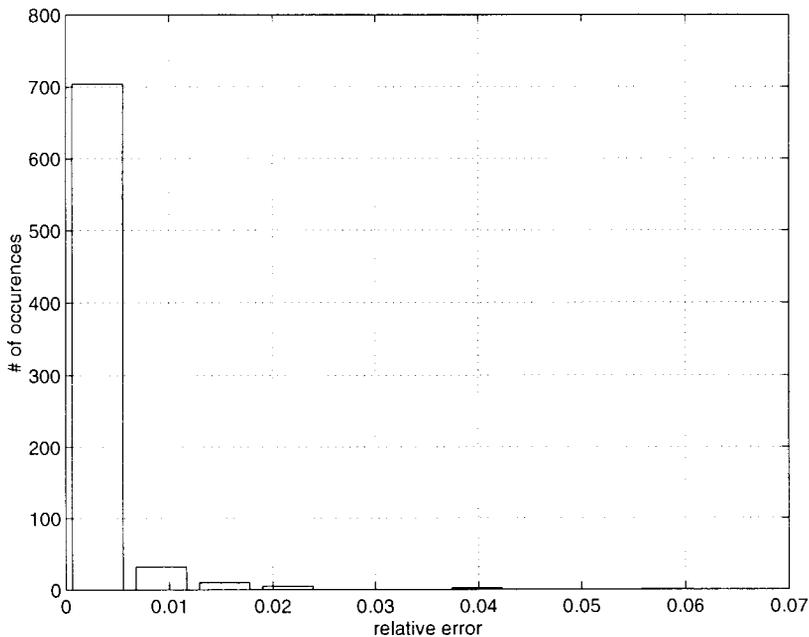


Figure 4.24: Relative error distribution of the LSS for C432.

For the CRAB-PE phase, three different groups of vector sets were generated.

The first group consisted of 704 different PIN statistics (1,2,3,4,5 and 36 PINs biased to LA=5%, HA=95%). The results of the evaluation of the first group is depicted in Fig. 4.25. About 99% of the vector sets resulted with relative errors under 10%. Relative errors of less than 15% were the result of evaluations with 3 vector sets where 5 PINs were biased with 6% LA and a vector set with 4 PINs were biased with 10% LA.

The second group of vector sets is composed of 680 different PIN statistics from PINSTAT(36, 5%, 95%). The results of this showed that approximately 80% of the tests have relative errors under 20%. The results exhibit larger relative errors (from 20% to 90%) for experiments that used more LA PIN biases (as depicted in Fig. 4.26). However, in terms of *absolute error* the errors are comparable even smaller than the results with under 10% relative error.

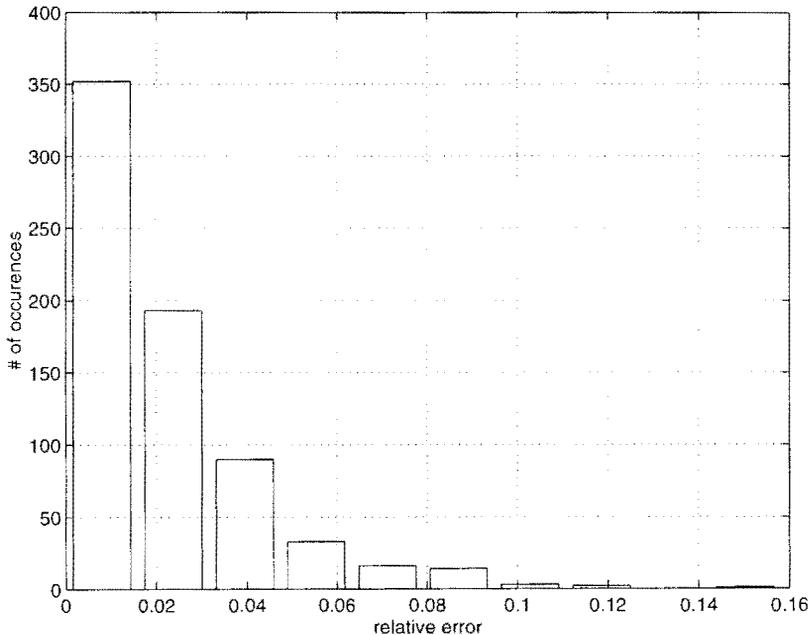


Figure 4.25: Relative error distribution of the 704 CRAB-PE results for C432.

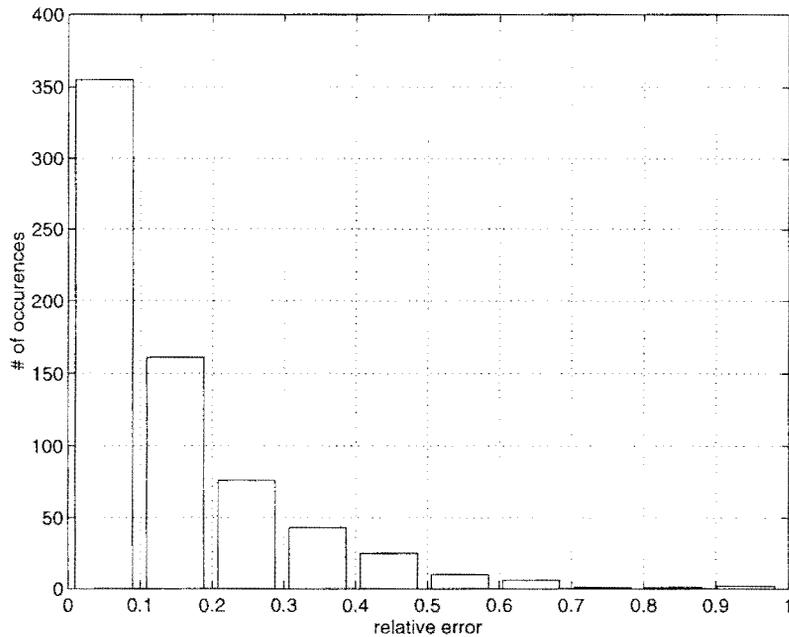


Figure 4.26: Relative error distribution of the 680 CRAB-PE results for C432.

The third group of vector sets is composed of uniform white noise data. For 29 test cases, the CRAB model returned power estimates under 2% within Quickpower (refer Fig. 4.27). *The aim of testing the model for uniform white noise is to verify the CRAB model superiority to previous models proposed in the literature.* CRAB model provides very accurate results within low level estimates for uniform white noise as well as a wide-range biased input data.

Similar to previous circuits, C432 was the first circuit with a high number of PINs (36) and required at least 702 vector sets for the CRAB-PC phase. The PIN LA and HA were selected at 5% and 95% aggressively from the end points of the t_i^{sw} axis. The experimental results show that the LA bias on four or more PINs had relative errors greater than 10%. Since the CRAB power model contains only second-order terms, higher-order correlations' effects appear as error terms in the

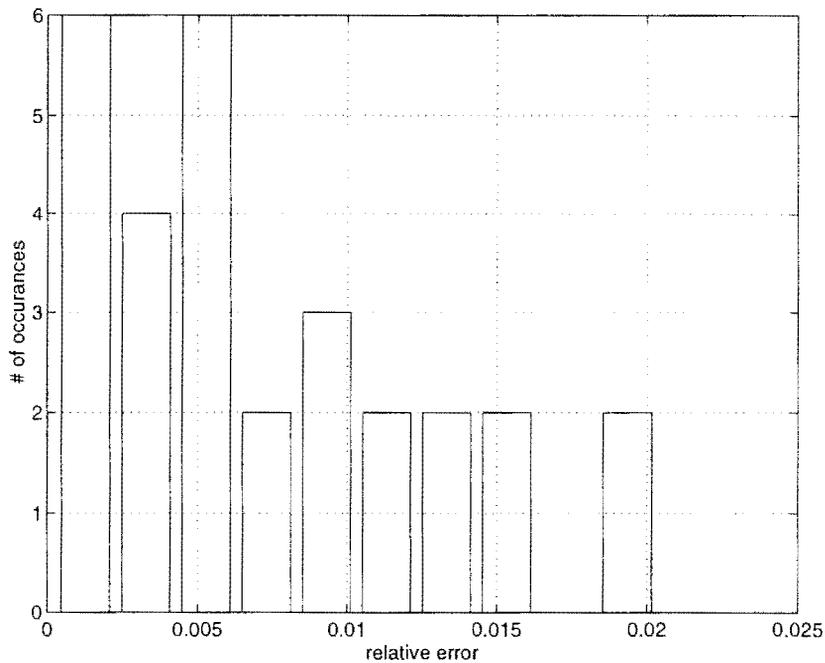


Figure 4.27: Relative error distribution of the 29 CRAB-PE results for C432 using uniform white noise data.

model. The higher relative errors are likely caused by unmodeled contributions of these higher-order terms. On the other hand, the absolute power contribution of this circuit with 5 or more LA PINs is only 20% of the power contribution of the same circuit with a single LA PIN (when the rest of the PINs are random). Therefore, the absolute errors of power estimates for many PINs set at low activity and all PINs set to uniform white noise are of the same order.

4.2.6 C499 Results

CRAB-PC phase requires at least 902 vector sets for the C499 circuit's 41 PINs. For the CRAB-PC phase, 972 vector sets of length 2000 were generated with `PINSTAT(41,%5,%95)`. The model coefficients of length 902 were solved by MATLAB as in previous cases. The coefficients ranged from -0.0011 ($k_{14,15}$) to

$7.3781e-04$ ($k_{5,27}$). For the same 972 vector sets of length 2000 were applied to this circuit, the relative error histogram is depicted in Fig. 4.28. The LSS for this circuit exhibited almost the same behavior as C432. This time the number of vector sets that have under 1% relative error is above 99% of all cases tested.

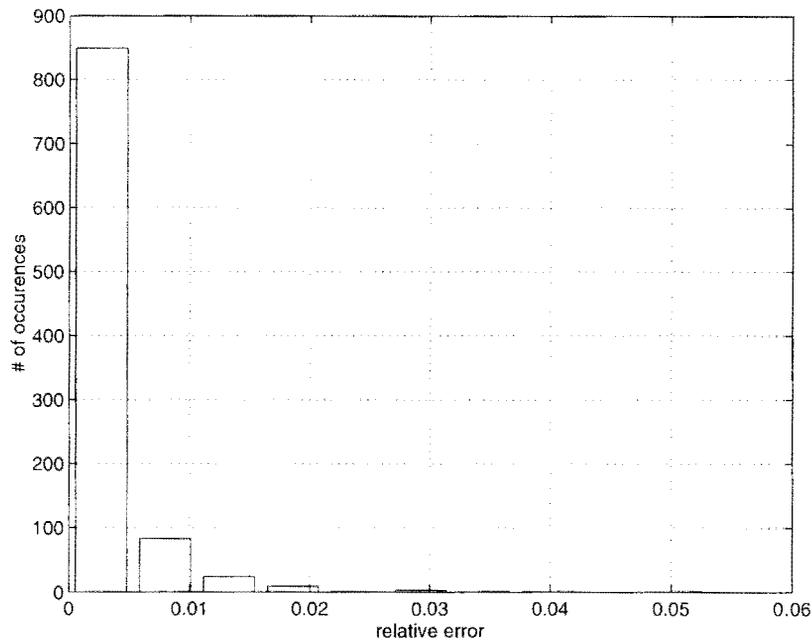


Figure 4.28: Relative error distribution of the LSS for C499.

For the CRAB-PE phase, new vector sets 827 in number with lengths 2000 were used to evaluate the model accuracy. The PIN statistics are from PIN-STAT(41, 5%, 95%). The CRAB model comparison for these experiments is shown in Fig. 4.29. It is a similar distribution compared to the one for C432. The tail of the relative error distribution shifted up to 80% for the low activity of multiple PINs (greater than 30). However, about 85% of the CRAB power estimates are under 20%. The same discussion for the high relative errors of C432 holds true for this circuit. In summary, if the absolute error displays a stable behavior for the

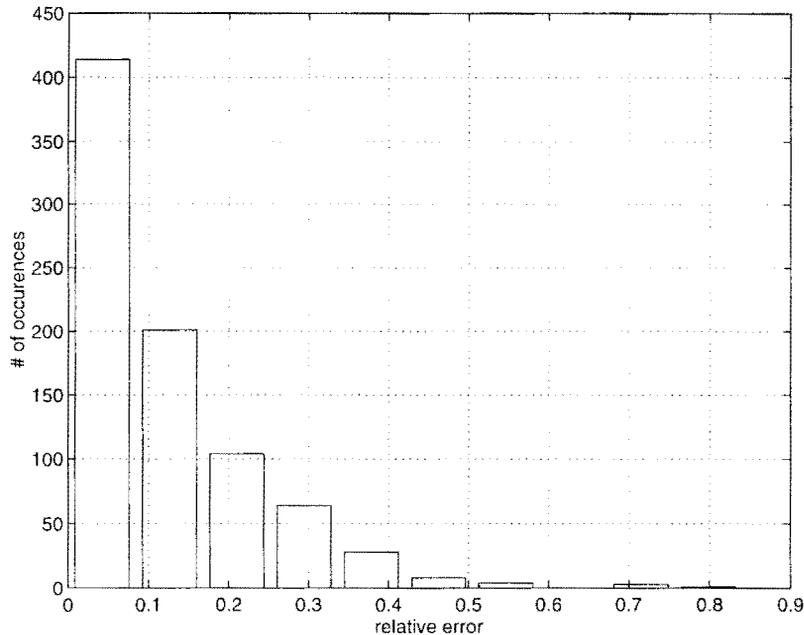


Figure 4.29: Relative error distribution of the 827 CRAB-PE results for C499.

complete-range PIN statistics (which is true for all cases), then the high relative errors have little effect on the model accuracy. In effect, the absolute power contribution of a micro-architectural block to the overall design’s power dissipation is not significant.

4.2.7 C1908 Results

The ISCAS C1908 needs 594 CRAB model coefficients to be stored in the CRAB-PC phase. For the experiments, 672 vector sets were generated for the Quickpower analysis. As in the previous cases (except for the values of LA and HA), the PIN statistics of the vector sets are from PINSTAT(33, 10%, %90). And as before, LSS was used to extract the coefficients. For the previous three ISCAS benchmarks, (C17, C432, C499) the PIN switching activities were biased to 5%

and 95%, whereas, for C1908, they were biased to 10% and 90% that is the two end points on the t_i^{sw} axis (recall Fig 4.4) were selected such that they are closer to UWN. The aim of this experiment is to show that the characterization and evaluation based on a narrower range can yield acceptable (in some cases better) results for PIN statistics biased to LA. This effect is explained by noting that modeling power dissipation by using $N(N+3)/2$ number of points in a narrower range would give better estimates than a wider range using the same number of points.

The C1908 model coefficients varied from $-1.074e-05$ ($k_{2,9}$) to $3.095e-05$ (k_{16}^{sw}). The histogram of the relative error after the least squares solution is shown in Fig. 4.30. Since the low and high activity was set to 10% and 90% respectively (higher and

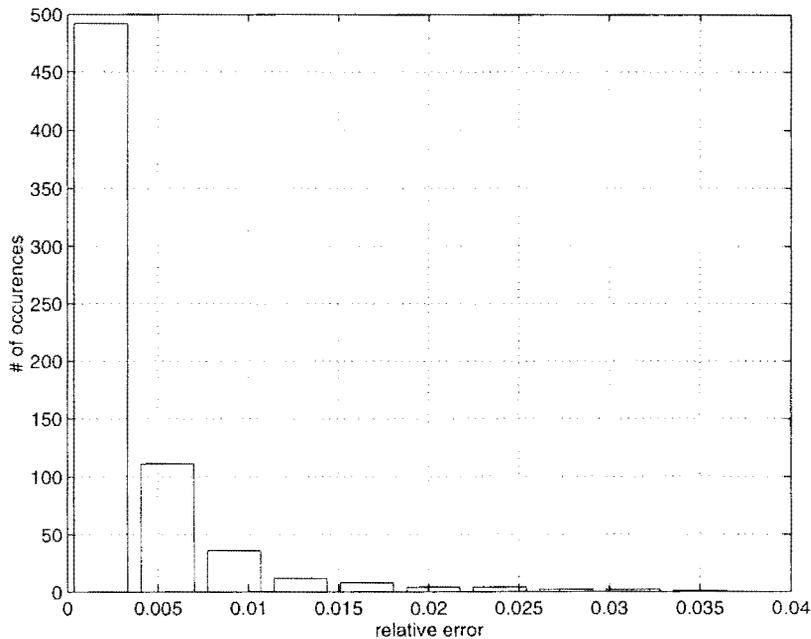


Figure 4.30: Relative error distribution of the LSS for C1908.

lower than 5% and 95% respectively), the relative error for all vector sets is under

3.5%.

For subsequent evaluation of the model, new vector sets, 512 in number, with varying PIN statistics from $\text{PINSTAT}(33,10\%,90\%)$ were generated by IVG. The histogram of the relative error was obtained from MATLAB after the CRAB model was evaluated. As seen in Fig. 4.31, 96% of test cases have relative error under 15% which is an improvement from other benchmarks. Again the larger relative errors between 15% and 45% were caused by the multiple low PIN activity (e.g. when sets of 19 PINs were biased with 10%).

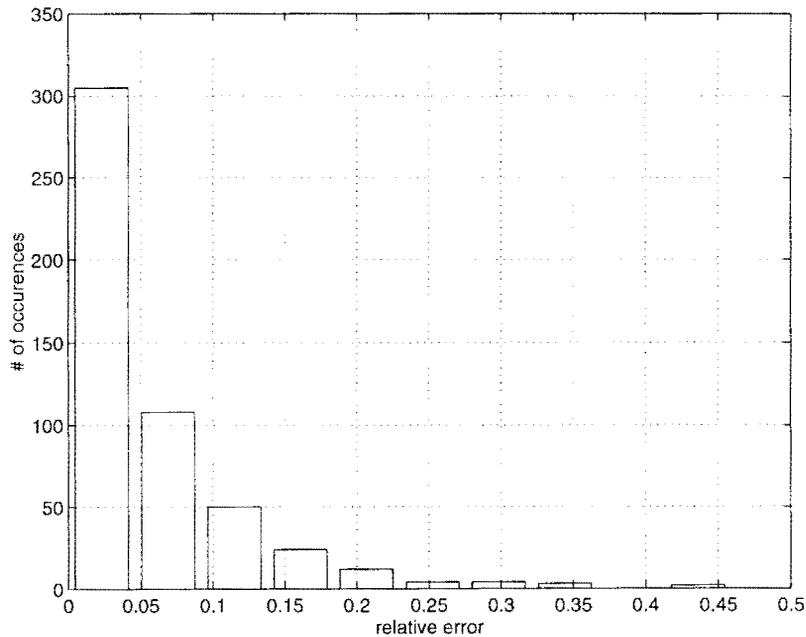


Figure 4.31: Relative error distribution of the 512 CRAB-PE results for C1908.

For this circuit, LA and HA values were different than those of C17, C432, C499. In fact the two end points in the t_i^{sw} axis (Fig 4.4) was selected closer which improved the estimates from the model as expected. This trade-off suggests a possible optimization technique: Activity range vs range of power estimate.

4.2.8 C6288 Results

C6288 is a 16 bit multiplier, unlike other circuits C6288 exhibited more glitches per vector during Quickpower analysis (CRAB-PC phase). The required number of vector sets for the characterization is 560. For this, PINSTAT(32,10%,90%) was used. To reduce the computation time the vector length was 500. The CRAB model coefficients were extracted by MATLAB as before. They were found to be between -0.6420 ($k_{11,21}$) and 0.5616 ($k_{14,18}$). The relative error distribution for these vector sets is depicted in Fig. 4.32.

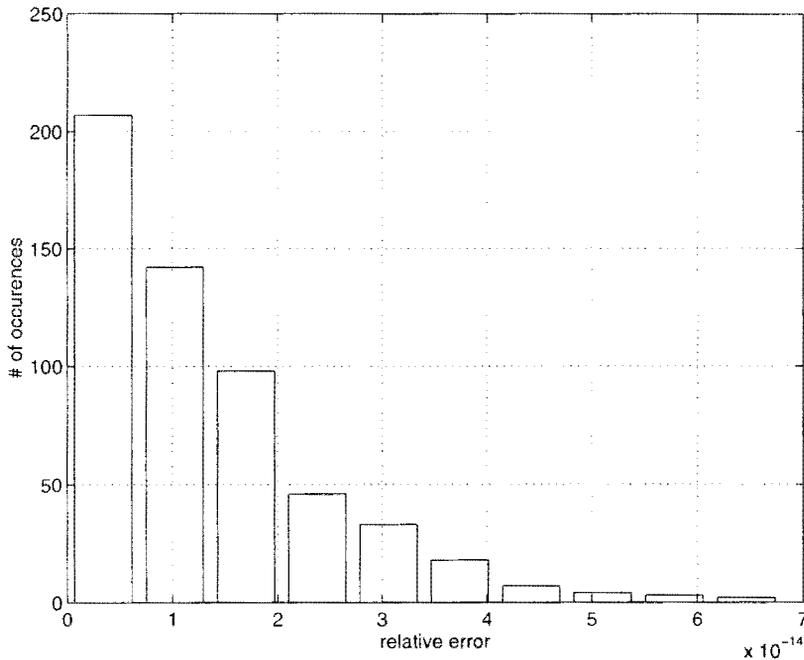


Figure 4.32: Relative error distribution of the LSS for C6288.

Since the Quickpower analysis run-time for 500 vectors was around 30 to 40 minutes in a Sparc 10, 121 number of runs was performed for the CRAB-PE phase. The 121 vector sets with different PIN activities were generated as in the CRAB-PC phase. The evaluation of the model for these activities exhibited a relative

error distribution depicted in Fig. 4.33. The errors went beyond 100% for some vector sets. And the majority (about 70%) of the errors are under 25%.

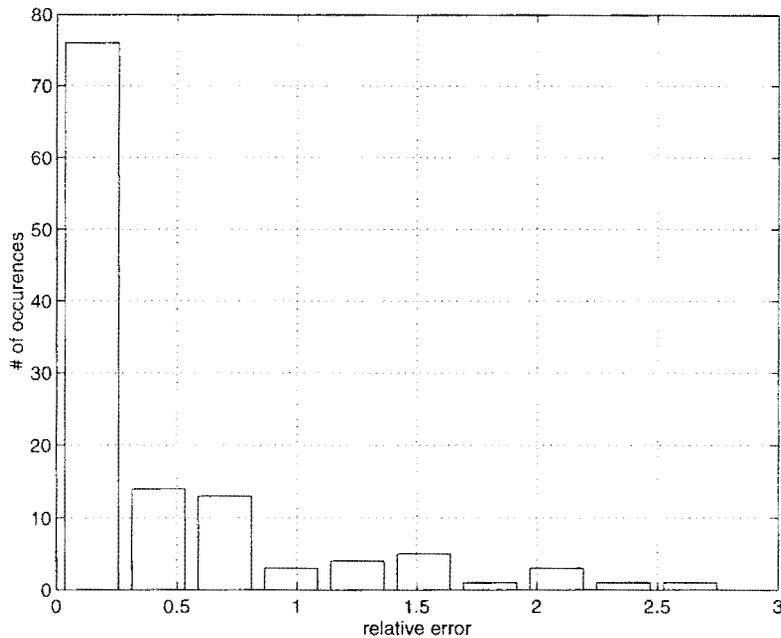


Figure 4.33: Relative error distribution of the 121 CRAB-PE results for C6288.

There may be several possible explanations for the very large relative errors:

1. Selection of the vector length: 500 vectors was chosen to decrease the runtime cost. However, this length is not enough for this circuit to get reliable average power values from Quickpower. Because in [29] for the specified accuracy ($\epsilon=0.1$, $1 - \delta=0.9$), the required number of vectors for the C6288 is 19,000.

2. Modeling the glitches: The number of glitches at internal or primary output nodes change with respect to different PIN activity values. Because of the large number of glitches in the multiplier, high order spatial correlations become significant.

With this circuit, it was observed that the CRAB-PC efficiency directly depends

on both the low-level power estimator and the machine used. However, the time cost can be decreased further by processing the vector sets concurrently. For example, 560 vector sets can be divided into 10 groups and applied in parallel such that the time cost for obtaining power values from Quickpower is reduced as a factor of 10.

In [29], it is shown that there is a trade-off between the specified accuracy $(\epsilon, 1 - \delta)$ and the number of vectors. Hence the use of 500 vectors (which is significantly less than 19,000) resulted with higher error (ϵ) and lower confidence level $(1-\delta)$.

4.3 Chapter Summary

The four CRAB power models proposed in Chapter 3 were evaluated with ISCAS benchmarks and several micro-architectural blocks. For circuits with small numbers of PINs ranging from 5 to 10, three of the proposed models provided power estimates with relative errors under 10% for the majority of all cases. The deviations from this range were caused when a high number of PIN combinations were set to low activity (e.g. all PINs biased to LA). The relative errors for low activity regions were decreased by including the pairwise second-order cross-terms in the original model.

The final model was tested with larger circuits with PINs ranging from 32 to 41. The worst-case relative error was observed to be 90% when a large number of PIN combinations (e.g. 5 PINs or more for ADD4) were biased to very low switching probability (e.g 5%). This problem was investigated by narrowing the [LA, HA]. The narrow range CRAB-PC provided better results. This is explained by the

observation that average power points (see Fig. 4.4), the least squares solution returns a better fit to the power distribution for a narrower switching activity range. Hence the CRAB-PC can be improved further by dividing the switching probability space to two or more regions. For example, one set of model coefficients could be stored for the LA region (e.g. 1% to 50%) and the other set could be kept for the HA region (e.g. 50% to 99%) so that the relative errors for the entire range are decreased.

Absolute error in addition to *relative error* of the CRAB power estimate was also compared to explain the practicality of the model for high relative errors.

The CRAB power estimates were compared with the CES results. The superiority of the CRAB model to CES is verified with various PIN activities. Additionally, the CRAB models are evaluated for *DBT* and *uniform white noise* data streams. The relative error for both PIN activities are shown to be under 5% of the Quickpower values.

The above observations are promising in the sense that, the CRAB model provides reliable results for the *complete-range PIN activity* which has not been considered in previous research.

Chapter 5

Future Work

In this thesis, the CRAB power model is presented and evaluated. Some of the limitations and extensions for the proposed model will be discussed in this chapter.

The inclusion of second-order cross-terms improved the accuracy with respect to low-level power estimates, however the required number of coefficients increases quadratically in the number of primary input bits. The limitations in the computational resources may increase the costs of the CRAB-PC phase. Although the time cost may be of limited concern because it is a one time process for each micro-architectural block. However, the reduction of this time cost is a possible modification to the model presented. Two possible reduction methods may be decreasing the number of model coefficients or reducing the number of total vector pairs for the CRAB-PC phase. The reduction of model coefficients may be possible by using the statistical techniques suggested by Pedram [24]. A reduction of the number of input vector pairs can be achieved by systematically selecting fewer vectors for each run or using less than $N(N+3)/2$ PIN statistics for the complete-range characterization. These two extensions are worthwhile to work on. Beyond CRAB-PC performance improvements, the model predictions may be improved further by dividing the complete-activity range to two or more regions and completing CRAB-PC in each of these regions. However this would increase the

total number of coefficients. Using a simpler model (e.g. first-order CRAB model) would be a way to optimize the number of model coefficients versus the number of activity range divisions.

During the CRAB-PA phase, in the case of no behavioral simulation, the transfer of RTL design PIN statistics to internal sub-block's input statistics is required. For this, the extraction of the transfer coefficients (from PIN statistics to PON statistics) can be completed during the MCE step of the CRAB-PC phase. In other words, the transfer coefficients from input statistics to output statistics of a micro-architectural block can be solved by using the same linear system in Fig. 3.3. The only difference would be to change the right-hand side vector such that the output statistics would replace the power values. This task is a possible extension to the CRAB-RPE technique.

The CRAB RTL power model was developed for average power estimation. Recently, estimation of peak power has emerged as another concern for the designers. To observe the peak power, cycle-accurate power estimation techniques have been proposed [24]. The CRAB model readily predicts peak power during the CRAB-PC phase.

The application and modification of the model to a wider range of micro-architectural blocks is another area of research. In this thesis, the CRAB-RPE was proposed for combinational circuit blocks. However, it is possible that it can also be applied to sequential circuits, memories, control blocks and other finite state machines (FSM).

After modifying the CRAB power model to other micro-architectural blocks (i.e. sequential circuits, control blocks etc.), power estimation at the higher levels such as the behavioral level would be possible.

Chapter 6

Conclusion

In this thesis, the background for the RTL power estimation is established and a novel RTL power estimation descriptive technique called CRAB-RPE is presented. CRAB-RPE has two phases (CRAB-PC and CRAB-PA) which are built upon the CRAB power model. CRAB-PC is the characterization phase where the model coefficients are extracted and CRAB-PA is the analysis phase where the RTL design is synthesized to micro-architectural blocks in the RTL library and power contributions of each block are evaluated.

The CRAB model originated from earlier gate-level switching activity estimation techniques for spatially and temporally correlated data. In many of these techniques, gate output switching activity is shown to be first-order, second-order and higher-order function of gate input transition probabilities. From this knowledge, the following observation is made in this thesis. The average power of the whole design is also a weighted sum of first-order, second-order and higher-order PIN transition probabilities. Based on this observation, the CRAB RTL power model is introduced as a linear function of first and second-order PIN transition probabilities and the higher-order terms for transition probabilities are a known error term.

The name CRAB (Complete-Range Activity-Based) comes from the fact that

the activity range from very low values to very high values is spanned by sampling with three activity points for every PIN. Hence the effects of full activity range for each PIN is characterized and evaluated for each micro-architectural block. The wide-range of CRAB PIN statistics also model data activity regions that previous RTL power models are based on. Specifically, they are uniform white noise and temporally correlated two's complement data. The CRAB technique also accounts for the spatial correlations.

The CRAB power model has been evaluated for different circuits including the ISCAS combinational benchmarks for various PIN activities. The model relative error results are less than 5% for biased single and pair PIN statistics or random or DBT-like (higher order bits are very low or high active and the data bits are uniform white noise) data. Additionally, the model was tested aggressively by biasing a wide-range of combinations of PIN statistics using the PINSTAT algorithm. In those cases, the model predicted power values within acceptable absolute errors. The largest relative errors are caused by high number of low activity PINs which would contribute little to the total design's power. Improvement in the model accuracy was observed by narrowing the modeled activity range. The largest deviations in the CRAB-PE tests were obtained with C6288 (multiplier) because of unreliable CRAB-PC phase average power values.

In conclusion, the CRAB technique has made a significant contribution to existing RTL power estimation techniques by considering the effects of PIN transition probabilities on the power dissipation. Development of new methodologies for high level power estimation remains an open area for researchers to focus their efforts on.

Bibliography

- [1] Sasan Iman, *Estimate Power, Iterate Design Flow at the Highest Levels*, Computer Design, April 1997, pp. 29-32.
- [2] M. Pedram, *Power Minimization in IC Design: Principles and Applications*, ACM Transactions on Design Automation of Electronic Systems, vol. 1, no. 1, January 1996, pp. 3-56.
- [3] Paul Landman, *Low Power Architectural Design Methodologies*, Ph.D. Thesis, UC Berkley, August 1994.
- [4] J.M.Rabaey, M.Pedram, *Low Power Design Methodologies*, Kluwer Academic Publishers, 1996.
- [5] N. H. E. Weste, K. Eshraghian, *Principles of CMOS VLSI Design*, chapter 4, Addison Wesley, 1994.
- [6] H. J. M. Veendrick, *Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits*, IEEE Journal of Solid State Circuits, August 1984, pp. 468-473.
- [7] M. Nemani, F. Najm, *Towards a High-Level Power Estimation Capability*, IEEE Trans. on CAD of ICs and Systems, vol.15, June 1996, pp. 588-598.
- [8] F. Najm, *Transition Density: A new measure of Activity in Digital Circuits*, IEEE Trans. CAD of ICs and Systems, vol.12, February 1993, pp. 310-323.
- [9] M. Xakellis, F.Najm, *Statistical Estimation of the Switching Activity in Digital Circuits*, Proc. 31st Design Automation Conference, San Diego, CA, June 1994, pp. 728-733.

- [10] F. N. Najm, *Feedback, Correlation, and Delay Concerns in the Power Estimation of VLSI Circuits*, Proc. 32nd Design Automation Conference, 1995, pp. 612-617.
- [11] P. Landman, *High Level Power Estimation*, IEEE International Symposium on Low-Power Electronics and Design, Monterey, CA, 1996.
- [12] K. D. Muller-Glaser, K. Kirsch, K. Neusinger, *Estimating Essential Design Characteristics to support Project Planning for ASIC Design Management*, Proc. International Conference on Computer Aided Design, Los Alamitos, CA, November 1991, pp. 148-151.
- [13] D. Liu, C. Svensson, *Power Consumption Estimation in CMOS VLSI Chips*, IEEE Journal of Solid-State Circuits, vol. 29, no. 6, June 1994, pp. 663-670.
- [14] M. Nemani, F. Najm, *High-level power estimation and the area complexity of boolean functions*, IEEE International Symposium on Low Power Electronics and Design, Monterey, CA, August 12-14, 1996, pp. 329-334.
- [15] D. Marculescu, R. Marculescu, M. Pedram, *Information Theoretic Measures for Power Analysis*, IEEE Trans. on CAD, vol. 15, no. 6, 1996, pp. 599-610.
- [16] S. R. Powell, P. M. Chau, *A Model for Estimating Power Dissipation in a Class of DSP VLSI Chips*, IEEE Trans. on Circuits and Systems, vol. 38, no. 6, June 1991, pp. 646-650.
- [17] T. Sato, Y. Ootaguro, M. Nagamatsu, H. Tago, *Evaluation of Architecture-Level Power Estimation for CMOS RISC Processors*, IEEE International Symposium on Low Power Electronics and Design, San Jose, CA, 1995.*
- [18] P. E. Landman, J. M. Rabaey, *Black-Box Capacitance Models for Architectural Power Analysis*, Proc. of the 1994 International Workshop On Low-power Design, Napa Valley, CA, April 1994.*

- [19] P. E. Landman, J. M. Rabaey, *Architectural Power Analysis: The Dual Bit Type Method*, IEEE Trans. on VLSI Systems, vol. 3, no. 2, June 1995, pp. 173-187.
- [20] P. E. Landman, J. M. Rabaey, *Activity-Sensitive Architectural Power Analysis*, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, no. 6, June 1996, pp. 571-587.
- [21] S. Gupta, F. Najm, *Power Macromodeling for High Level Power Estimation*, Proc. 34th Design Automation Conference, Anaheim, CA, June 1997, pp. 215-218.
- [22] S. Ramprasad, N. R. Shanbhag, I. N. Hajj, *Analytical Estimation of Transition Activity From Word-Level Signal Statistics*, Proc. 34th Design Automation Conference, Anaheim, CA, June 1997.*
- [23] C. Ding, C. Hsieh, Q. Wu, M. Pedram, *Stratified Random Sampling for Power Estimation*, Proc. International Conference on Computer Aided Design, November 1996, pp. 577-582.
- [24] Q. Qiu, Q. Wu, M. Pedram, C. Ding, *Cycle-Accurate Macro-Models for RT-Level Power Analysis*, IEEE International Symposium on Low Power Electronics and Design, Monterey, CA, 1997, pp. 125-130.
- [25] R. Marculescu, D. Marculescu, M. Pedram, *Switching Activity Analysis Considering Spatiotemporal Correlations*, Proc. International Conf. Computer-Aided Design, 1994.*
- [26] P. Schneider, U. Schlichtmann, B. Wurth, *Fast Power Estimation of Large Circuits*, IEEE Design and Test of Computers, Spring 1996, pp. 70-78.
- [27] H. Mehta, M. Borah, R. M. Owens, M. J. Irwin, *Accurate Estimation of Combinational Circuit Activity*, Proc. 32nd Design Automation Conference, 1995.*

- [28] Q. Qiu, Q. Wu, M. Pedram, C. Ding, *Cycle-Accurate Macro-Models for RT-Level Power Analysis*, IEEE International Symposium on Low Power Electronics and Design, 1997.*
- [29] A. M. Hill, S. Kang, *Determining Accuracy Bounds for Simulation-Based Switching Activity Estimation*, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, no. 6, June 1996, pp. 611-618.
- [30] William H. Press et. al., *Numerical Recipes in C*, New York, NY, USA, Cambridge University Press, 1992.
- [31] MENTOR GRAPHICS, *Autologic Synthesis II Manual*, Online.
- [32] MENTOR GRAPHICS, *Language-Based Tutorial*, QuickPower User's and Reference Manual.
- [33] D. V. Nguyen, *An ASIC power analysis system for digital CMOS design*, M.S. Thesis, Portland State University, 1997.
- [34] D. A. Preece, *Distributions of the Final Digits in Data*, The Statistician, vol. 30, no. 1, March 1981, pp. 31-60.
- [35] A. Salz, M. Horowitz, *IRSIM: An Incremental MOS Switch-Level Simulator*, Proc. 26th Design Automation Conference, 1989, pp. 173-178.
- [36] C. Deng, *Power Analysis for CMOS/BiCMOS Circuits*, Proc. of the 1994 International Workshop on Low Power Design, April 1994, pp. 3-8.
- [37] B. J. George, D. Gossain, S. C. Tyler, M. G. Wloka, G. Yeap, *Power Analysis and Characterization for Semi-custom design*, Proc. of the 1994 International Workshop on Low Power Design, April 1994, pp. 215-218.
- [38] John Ousterhout, *MAGIC: Tutorial#1-#4*, Department of Electrical Engineering and Computer Sciences, University of California Berkeley, CA.

* Page numbers were not available from CD-ROM.