Power Macromodeling for High Level Power Estimation[†]

Subodh Gupta and Farid N. Najm
ECE Dept. and Coordinated Science Lab.
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Abstract – A modeling approach is presented that captures the dependence of the power dissipation of a combinational logic circuit on its input/output signal switching activity. The resulting *power macromodel*, consisting of a single three dimensional table, can be used to estimate the power consumed in the circuit for any given input/output signal statistics. Given a low-level (typically gate-level) description of the circuit, we describe a characterization process by which such a table model can be automatically built. In contrast to other proposed techniques, this can be done for *any* given logic circuit without any user intervention, and applies to all possible input/output signal statistics; it does not require one to construct specialized analytical equations for the power dissipation. The three dimensions of our table-based model are the *average input signal probability*, *average input transition density*, and *average output zero-delay transition density*. This approach has been implemented and models have been built for many benchmark circuits. Over a wide range of input signal statistics, we show that this model gives very good accuracy, with an RMS error of under about 6%.

I. Introduction

With the advent of portable and high-density microelectronic devices, the power dissipation of very large scale integrated (VLSI) circuits is becoming a critical concern. Modern microprocessors are hot, and their power consumption can exceed 30 or 50 Watts. This fact is evident from the recent introduction of a 50 W 300 MHz implementation of the DEC Alpha architecture [1]. Due to limited battery life, reliability issues, and packaging/cooling costs, the power consumption can be a more critical design concern than speed and area in some applications. Hence, to avoid problems associated with excessive power consumption, there is a need for CAD tools to help in estimating the power consumption of VLSI designs.

A number of CAD techniques have been proposed for gate-level power estimation (see [2] for a survey). However, by the time the design has been specified down to the gate level, it may be too late or too expensive to go back and fix high power problems. Hence in order to avoid costly redesign steps, power estimation tools are required that can estimate the power consumption at a high level of abstraction, such as when the circuit is represented only by Boolean equations. This would provide the designer with more flexibility to explore design trade-offs early in the design process, reducing the design cost and time.

In response to this need, a number of high-level power estimation techniques have been recently proposed (see [3] for a survey). Two styles of techniques have been proposed, which we refer to as top-down and bottom-up. In the top-down techniques [4, 5], a combinational circuit is specified only as a Boolean function, with no information on the circuit structure, number of gates/nodes, etc. These methods are still in their infancy, and they currently provide estimates only of the switching activity, and not of the total power. Top-down methods would be useful when one is designing a logic block that was not previously designed, so that its internal details are unknown.

In contrast, bottom-up methods [6, 7, 10, 11] are useful when one is reusing a previously-designed logic block, so that all the internal structural details of the circuit are known. In this case, one develops a *power macromodel* for this block which can be used during high-level power estimation (of the overall system in which this block is used), in order to estimate the power dissipation of this block without performing a more expensive gate-level power estimation on it.

The method in [6] uses the power factor approximation technique, which treats all the circuit input bits as digital "white noise" and due to this assumption can give errors of up to 80% in comparison to gate-level tools. Although [7] gives more accurate result, its main disadvantage is that it treats different modules differently, requiring specialized analytical expressions for the power to be provided by the user. Thus, depending upon the functionality of the module, a different type of macromodel (analytical equation) may have to be used.

The method in [10] characterizes the power dissipation of circuits based on input transitions rather than input

[†] This work was supported by Rockwell, by Intel Corp., and by the National Science Foundation, MIP-9623237.

DAC 97, Anaheim, California
(c) 1997 ACM 0-89791-920-3/97/06 ..$3.50

statistics. Since the number of possible input transitions for an n-input combinational circuit is 2^{2n}, they present a clustering algorithm to compress the input transitions into clusters of input transitions that have the same power values (approximately). They use heuristics to implement the clustering algorithm, but it is not clear how efficient the method would be on large circuits.

In [11], the authors present a technique to estimate switching activity and power consumption at the RTL for data path and control circuits, in the presence of glitching activity. To construct a power macromodel, they use both analytical equations and look-up tables. The method is quite good and uses 9 or more variables in the power macromodel. Our independent work has shown that it is possible to construct a look-up table power macromodel with much fewer variables (only 3 can be enough).

In this paper, we propose a power macromodeling approach that (1) takes into account the effect of the circuit input switching activity and does not treat the circuit inputs as white noise, and (2) is based on a single fixed macromodel template which does not depend on the type of model being analyzed. Our model is table-based. Specifically, we construct a three dimensional look-up table whose axes are the *average input signal probability* (P_{in}), *average input transition density* (D_{in}) and *average output zero delay transition density* (D_{out}). For a logic node, the transition density is defined as the average number of logic transitions per unit time [8]. The zero delay transition density refers to the case when the circuit gates are considered to have zero delay, so that only truly required logic transitions (and no hazards or glitches) are observed. From a high-level view, it is reasonable to assume that fast functional simulation will be applied to measure signal switching statistics, so that only the zero delay output density (and not the general delay output density) will be computed. The main advantage of our approach is that all types of circuits are treated in the same way, i.e., we do not use different model equation types for different modules. As a result, the method is very easy to use, and requires no user intervention. Indeed, we will present an automatic characterization procedure by which the macromodel can be built for a given circuit.

The paper is organized as follows. In section II we will discuss the macromodeling problem in more detail. In section III we will describe the characterization procedure for the models. In section IV we will evaluate the accuracy of the macromodels and in section V we will give some conclusions.

II. Power Macromodeling

What should a power macromodel look like? Which features are desirable and which are too expensive and infeasible? To begin with, it is clear that a macromodel should be simple to evaluate, otherwise there would be no advantage in using it and one might as well perform the analysis at the gate level. Furthermore, it must apply over the whole range of possible input signal statistics. Finally, it should consist of a fixed template, in which certain parameter values can determined by a well-defined and automatic process of *characterization*, without user intervention. We present a macromodel that has all these properties.

Since the power depends on the circuit input switching activity, it is clear that a power macromodel should take the input activity into account. The question is, exactly what information about the inputs should be taken into account and included in the macromodel. When the circuit being modeled is small (one or a few gates), then a simple modeling strategy is to create a table that gives the power for every possible input vector pair. In this case, there is no loss of accuracy. However, this strategy cannot be applied to large circuits. A circuit with 32 inputs will have 2^{64} possible input vector pairs, which would be prohibitively expensive to store in a table.

This leads to a trade-off between the amount of detail that one includes about the inputs and the accuracy resulting from the model. One possibility is to consider the signal probability $P(x_i)$ and transition density $D(x_i)$ at every input node x_i, and to build a model that depends only on these two variables. Notice that any information about correlations between the input nodes is lost when this is done. Thus, for instance, one could consider building a table which gives the power for every given assignment of input $P(x_i)$ and $D(x_i)$ values. Even in this case, however, such a table-based model would be too expensive, because a circuit with 32 inputs would require a 64-dimensional table.

Given the above observations, we have considered what aggregate compact descriptions of the $P(x_i)$ and $D(x_i)$ values would be sufficient to model the circuit power. For instance, one could consider building a two-dimensional table whose axes would be the average input $P(x_i)$, which we will denote by P_{in}, and the average input $D(x_i)$, to be denoted D_{in}. In this case, two different input assignments of $P(x_i)$ and $D(x_i)$ values, which may lead to different power values, may have the same P_{in} and D_{in} averages, and the table would predict the same power for both assignments, obviously with some error.

Table I.

Details for a Number of ISCAS85 Circuits.

Circuit	Function	#inputs	#outputs	#gates
c432	Interrupt control	36	7	160
c880	ALU	60	26	383
c1908	Error correction	33	25	880
c2670	ALU and control	233	140	1193
c3540	ALU	50	22	1669
c5315	ALU	178	123	2307
c6288	Multiplication	32	32	2406
c7552	ALU	207	108	3512

We have studied how big this error can be, as follows. Given a gate-level circuit and for a certain fixed P_{in} and D_{in}, we generate a large number (80 or more) of P and D assignments at the circuit inputs that each have averages equal to the specified P_{in} and D_{in}. We then perform an accurate power estimation for each assignment using a Monte Carlo gate-level (with a general delay model) simulation technique [9]. The average of the resulting power values is a good candidate value to store in the table. For each of the estimated power values, any deviation from this average value is considered to be an "error" relative to this table. The root-mean-square (RMS) and maximum errors for ISCAS85 circuits (see Table I for details of these circuits) are reported in Table II, for $P_{in} = 0.4$ and $D_{in} = 0.4$. A density of 0.4 means that the node makes an average of 4 transitions in 10 consecutive clock cycles. The largest RMS error is about 17% and the largest maximum error is -40%.

Table II.

RMS and Maximum Error in the 2-dimensional Table Approach, when Total Power is Estimated.

Circuit	P_{in}	D_{in}	RMS.Error	Max.Error
c432	0.4	0.4	1.61%	34.88%
c880	0.4	0.4	1.77%	40.46%
c1908	0.4	0.4	1.74%	16.80%
c2670	0.4	0.4	2.43%	-31.61%
c3540	0.4	0.4	2.96%	35.77%
c5315	0.4	0.4	1.76%	20.94%
c6288	0.4	0.4	16.6%	-40.04%
c7552	0.4	0.4	3.37%	19.02%

The power estimator (simulator) used to generate this table uses a scalable-delay timing model that depends on fanout and gate output capacitance. Thus, it captures the glitching power accurately (multiple transitions per cycle due to unequal delay from the inputs to an internal node). The glitching power is hard to account for in a high-level model. This is why such a high RMS error is seen for c6288, in which some internal nodes make up to 20 transitions per cycle. The errors improve considerably if the power estimates are based on a zero-delay timing model, in which the glitches are excluded, as shown in Table III. The largest RMS error is now 1% and the largest maximum error is 27%.

In any case, with such a high RMS error, the total power estimation using Table II is too inaccurate. The simple 2-dimensional table approach is too simplistic. Another parameter is needed by which we can accurately model the variation of the power due to various input P and D assignments. We have found that if one more dimension is added to the table, reasonably good accuracy can be obtained. The third axis is the average output transition density over all the circuit output nodes, measured from a zero-delay (functional) simulation of the circuit, and which we will denote by D_{out}. The stipulation that D_{out} corresponds to zero-delay is not optional, but rather required for the following reason. We envision that during high-level, say RTL, power estimation, one would perform an initial step of estimating the signal statistics at the visible RTL nodes from a high-level functional simulation. These (zero-delay) statistics would then be applied to the power macromodel in order to estimate the power. Thus, the power model will be given by:

$$P_z = f(P_{in}, D_{in}, D_{out}) \tag{1}$$

In order to study the accuracy in this 3-d approach, and to perform a direct comparison with Tables II and III, we will show the errors in the estimation for the same $P_{in} = 0.4$ and $D_{in} = 0.4$ specifications as before. The value of D_{out} will naturally be different in different runs. For each circuit, we selected the largest subset of cases that has the same value (approximately) and examined the errors based on the results in that subset. It is clear from Table IV that the errors are much less now, and the RMS error in c6288 is now reduced to an acceptable 6%. The largest error of -33% is somewhat undesirable, but we will see later on that the spread of the error values over a wide range of input/output statistics is quite acceptable. For comparison with Table III, the errors in the zero-delay power are given in Table V. The RMS error is now below 0.77% and the maximum error is under about 12%.

Table III.

RMS and Maximum Error in the 2-dimensional Table Approach, when Zero-delay Power is Estimated.

Circuit	P_{in}	D_{in}	RMS.Error	Max.Error
c432	0.4	0.4	0.59%	16.02%
c880	0.4	0.4	0.85%	27.5%
c1908	0.4	0.4	0.46%	-7.28%
c2670	0.4	0.4	0.92%	-18.82%
c3540	0.4	0.4	0.83%	-19.07%
c5315	0.4	0.4	0.47%	10.88%
c6288	0.4	0.4	0.72%	-16.82%
c7552	0.4	0.4	1.01%	-15.54%

Based on the above and other data, we conclude that the 3-dimensional table approach is superior, without requiring much more computational or memory cost. When the macromodel is *used*, i.e., during high level power estimation, we assume that a functional RTL simulation is performed in order to measure the switching activity and signal probability at every *visible RTL node*. These are then averaged to get P_{in}, D_{in}, and D_{out}, which are used to look up the power value in the table for each combinational circuit block. In the next section, we will describe an automatic procedure by which a full 3-d look-up table-based macromodel can be built.

III. Characterization

We assume that the combinational circuit is embedded in a larger sequential circuit, so that its input nodes are the outputs of latches or flip-flops and that they make at most one transition per clock cycle. We assume that the sequential design is a single clock system and ignore clock skew, so that the combinational circuit inputs $x_1, x_2, \ldots, x_n$ switch simultaneously.

At this point it is helpful to give some definitions. The signal probability $P(x_i)$ at an input node x_i is defined as the average fraction of clock cycles in which the final value of x_i is a logic high. The transition density $D(x_i)$ at an input node x_i is defined as the average fraction of cycles in which the node makes a logic transition (its final value is different from its initial value). For brevity, in this section we will write P_i and D_i to represent $P(x_i)$ and $D(x_i)$. Both P_i and D_i are real numbers between 0 and 1.

Since the input signals x_i make at most a single transition per cycle, there is a special relationship between probability and density, given by:

$$\frac{D_i}{2} \leq P_i \leq 1 - \frac{D_i}{2} \tag{2}$$

The derivation of this property is rather simple, but it depends on a number of other definitions and facts that are not relevant to this paper, so it will not be included. Equation (2) can be rewritten as:

$$D_i \leq 1 - 2\,|P_i - 0.5| \tag{3}$$

so that for a given $P(x)$, $D(x)$ is restricted to the shaded region shown in Fig. 1.

Table IV.
RMS and Maximum Error in the 3-dimensional Table Approach, when Total Power is Estimated.

Circuit	P_{in}	D_{in}	D_{out}	RMS.Error	Max.Error
c432	0.4	0.4	0.44	0.97%	16.48%
c880	0.4	0.4	0.32	1.58%	27.87%
c1908	0.4	0.4	0.44	1.18%	12.71%
c2670	0.4	0.4	0.37	1.78%	-18.82%
c3540	0.4	0.4	0.44	1.94%	-20.33%
c5315	0.4	0.4	0.42	1.76%	17.16%
c6288	0.4	0.4	0.44	6.05%	-33.54%
c7552	0.4	0.4	0.42	2.97%	-15.67%

Table V.
RMS and Maximum Error in the 3-dimensional Table Approach, when Zero-delay Power is Estimated.

Circuit	P_{in}	D_{in}	D_{out}	RMS.Error	Max.Error
c432	0.4	0.4	0.44	0.33%	4.90%
c880	0.4	0.4	0.32	0.55%	9.87%
c1908	0.4	0.4	0.44	0.19%	-3.23%
c2670	0.4	0.4	0.37	0.65%	-9.70%
c3540	0.4	0.4	0.44	0.47%	-12.37%
c5315	0.4	0.4	0.42	0.45%	6.32%
c6288	0.4	0.4	0.44	0.45%	-10.18%
c7552	0.4	0.4	0.42	0.77%	-8.82%

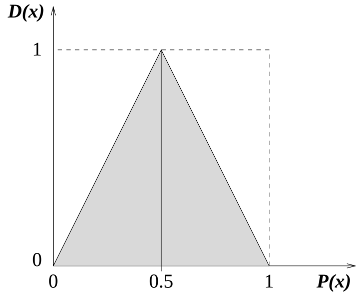

Figure 1. Relationship between density and probability for discrete-time signals.

We also recall the definitions of the average input probability, denoted P_{in}, and average input density, denoted D_{in}, as follows:

$$P_{in} = \frac{1}{n}\sum_{i=1}^{n} P_i \qquad \text{and} \qquad D_{in} = \frac{1}{n}\sum_{i=1}^{n} D_i \tag{4}$$

where n is the number of input nodes. It is clear from (2) that similar bounds hold for P_{in} and D_{in}:

$$\frac{D_{in}}{2} \leq P_{in} \leq 1 - \frac{D_{in}}{2} \tag{5}$$

from which we also have:

$$D_{in} \leq 1 - 2\,|P_{in} - 0.5| \tag{6}$$

Thus, the 3 dimensional table with axes P_{in}, D_{in}, and D_{out} will not be completely full, and the choices of P_{in} and D_{in} during characterization will have to satisfy the above constraints (5). We subdivide the probability and density axes between 0 and 1 into intervals of size 0.1, so that we form a 10 × 10 grid in the (P_{in}, D_{in}) plane. This choice is rather an arbitrary one, which we have found works well. Only about half of these points are valid, namely those that fall inside the shaded triangle in Fig. 1. Each valid grid point will correspond to a column of cells in the table along the D_{out} axis, as shown in Fig. 2.

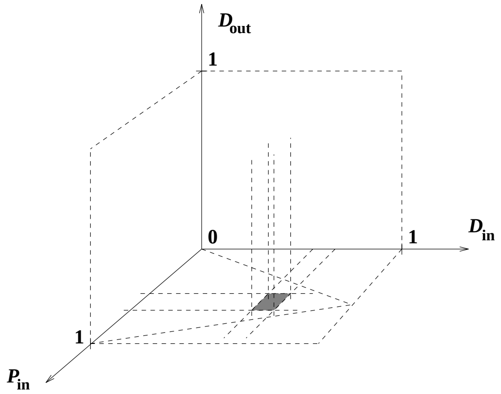

Figure 2. Three dimensional power macromodel.

For each valid grid point in the (P_{in}, D_{in}) plane, we randomly generate a large number of P and D assignments at the circuit inputs, all of which have average P and D values equal to the specific P_{in} and D_{in} at this grid point, and all of which satisfy the constraint (2). We will refer to such assignments to the circuit inputs as the "P vector" and the "D vector". For a given pair of P and D vectors, the circuit power is computed using Monte Carlo power estimation [9], and the value of D_{out} is computed as the average of the individual (zero-delay) density values at the circuit outputs, also found during the Monte Carlo analysis. The value of D_{out} is rounded to the nearest grid point on the D_{out} axis, and the power value obtained is associated with the resulting cell location $(P_{in}, D_{in}, D_{out})$ in the table. Eventually, a number of power values may be associated with a single cell in the table. At the end of the characterization, every cell is filled with the average of the power values associated with it. Some cells may have no power values associated with them, in which case their contents are left at zero. When it comes time to use the table, interpolation and extrapolation can be used to find the power for a $(P_{in}, D_{in}, D_{out})$ combination which does not exist in the table. In the next section, we will show a number of results that demonstrate the accuracy of this approach over a wide range of input statistics, in which interpolation and extrapolation were used whenever required.

The above characterization process is straightforward, except for the generation of the P and D vectors for a given P_{in} and D_{in}, which is explained below.

First, we randomly generate an input D vector such that the average of its components is equal to D_{in}, as follows. Based on a uniform distribution between 0 and 1, we use a random number generator to make an initial guess D_i^0 for $i = 1, \ldots, n$. The average of these D_i^0 values will probably be different from D_{in}, so we will scale them in a special way to make their average equal to D_{in} while keeping each of them between 0 and 1. To see how this works, define the following sum:

$$S_D = \sum_{i=1}^{n} D_i^0 \tag{7}$$

and notice that both S_D and nD_{in} are bounded by 0 and n. If $S_D > nD_{in}$, then we can find a value $0 < \lambda < 1$ such that:

$$\lambda S_D = nD_{in} \tag{8}$$

and the desired D_i values can be easily obtained as:

$$D_i = \lambda D_i^0 \tag{9}$$

so that they remain between 0 and 1, and their average is equal to D_{in}. If $S_D < nD_{in}$, then we can find a value $0 < \lambda < 1$ such that:

$$\lambda n + (1-\lambda) S_D = nD_{in} \tag{10}$$

and the desired D_i values can be easily obtained as:

$$D_i = \lambda + (1-\lambda) D_i^0 \tag{11}$$

so that their average is clearly equal to D_{in}. To see that the D_i values remain between 0 and 1, the above can be written as $D_i = D_i^0 + \lambda(1 - D_i^0)$, which is clearly between 0 and 1.

Given the D vector, we then generate the P vector so that its components have the specified average value P_{in}, and so that (2) is satisfied. The process is similar, but slightly more involved, than the generation of the D vector given above. Based on a uniform distribution between 0 and 1, we use a random number generator to make an initial guess P_i^0 for $i = 1, \ldots, n$. The average of these P_i^0 values will probably be different from P_{in}, so we will scale them in a special way to make their average equal to P_{in} and so they satisfy (2). To do this, we start by randomly choosing the values of n parameters, $\beta_1, \ldots, \beta_n$, such that:

$$\frac{D_i}{2} \le \beta_i P_i^0 \le 1 - \frac{D_i}{2} \tag{12}$$

The choice of each β_i is done using a random number generator based on a uniform distribution between the two bounds $D_i/2P_i^0$ and $(1 - D_i/2)/P_i^0$. We then compute the sum:

$$S_P = \sum_{i=1}^{n} \beta_i P_i^0 \tag{13}$$

Notice that both S_P and nP_{in} are bounded between $nD_{in}/2$ and $n - nD_{in}/2$, due to (12) and (5). If $S_P > nP_{in}$, then we can find a value $0 < \lambda < 1$ such that:

$$\lambda \frac{nD_{in}}{2} + (1-\lambda) S_P = nP_{in} \tag{14}$$

and if we set:

$$P_i = \lambda \frac{D_i}{2} + (1-\lambda) \beta_i P_i^0 \tag{15}$$

then it is clear that (2) is satisfied (since $\beta_i P_i^0 < 1 - D_i/2$ and because $0 < \lambda < 1$) and that the average of the P_i values is equal to P_{in}, due to (14).

Similarly, if $S_P < nP_{in}$, then we can find a value $0 < \lambda < 1$ such that:

$$\lambda \left(n - \frac{nD_{in}}{2}\right) + (1-\lambda) S_P = nP_{in} \tag{16}$$

and we can set:

$$P_i = \lambda \left(1 - \frac{D_i}{2}\right) + (1-\lambda) \beta_i P_i^0 \tag{17}$$

IV. Model Accuracy Evaluation

We have implemented this approach and built the power macromodels (3-dimensional look-up tables) for a number of combinational circuits. In order to study the accuracy over a wide range of signal statistics, we randomly generated P and D vectors at the circuit inputs without specifying P_{in} and D_{in} up-front. Since equation (5) must be enforced, any case that violated this constraint was rejected. Approximately 200 valid P and D vector assignments were generated this way, for which the power was estimated from gate-level Monte Carlo simulation. For each vector pair, the averages P_{in} and D_{in} were computed; the Monte Carlo simulation also provides accurate estimation of D_{out}. The power values predicted by the look-up table were compared to those from simulation, and the RMS and maximum errors were computed.

The results are summarized in Table VI. Since P_{in} and D_{in} were not specified up-front, one should not make a direct comparison between this and Tables II–V. Over a wide range of statistics, it is seen that the RMS error is very good, under about 6%. The largest maximum error is at 33% for c6288, but it can be seen from the scatter plot in Fig. 3 that this and the second largest maximum error value of 31% occur only in two (of about 1,600) cases, while all other cases have much better accuracy. An enlarged view of the lower section of that plot is given in Fig. 4. Both these plots report the normalized power values, so that the results for all the circuits can be examined on the same plot. For completeness, the accuracy of the macromodels when zero-delay power is estimated is shown in Table VII and in the scatter plot in Fig. 5. Over a wide range of signal statistics, the RMS error is below 0.83% and the maximum error is under 13%. The scatter plot also shows excellent agreement.

Finally, we should comment on the time required to do the characterization. Since the characterization needs to be performed only once, one can afford to spend some time on it. Nevertheless, at the time of this writing, the total time required to build the macromodel is not as small as one would like - it can take a few hours (SUN Sparc ELC) to build the macromodel for a circuit with a hundred or more input nodes. We are currently addressing this issue.

V. Conclusion

Since gate-level power estimation can be time consuming and because power estimation from a high level of abstraction is desirable so as to reduce design time and cost, we have proposed a power macromodeling approach. Our macromodel consists of a 3-dimensional look-up table with axes for average input signal probability, average input transition density, and average output (zero-delay) transition density. A novel and significant aspect of this approach is that we use the same model template for all types of combinational circuits, and no specialized analytical expressions are required. Another important fact is that this model works for all possible signal switching statistics.

We have shown why it is advantageous to use a 3-d rather than 2-d table, and described an automatic procedure for building the 3-d macromodel, without the need for user intervention. Once the model for a combinational block has been built, it can be used to estimate power during high-level power estimation, based on signal statistics that are computed from a high-level functional simulation. Over a wide range of input/output signal statistics, we have shown that this model gives very good accuracy, with an RMS error of under about 6%. Except for two out of about 1,600 cases, the largest error observed was under 20%. If one ignores the glitching activity, then the RMS error becomes under 1% and the largest maximum error (in all cases) under 13%.

Table VI.

Accuracy of the 3-d Look-up Tables,
When Total Power is Estimated.

Circuit	RMS.Error	Max.Error
c432	1.31%	-16.46%
c880	1.65%	31.4%
c1908	0.64%	-10.4%
c2670	1.825%	-17.03%
c3540	0.85%	19.21%
c5315	1.84%	-18.36%
c6288	6.06%	33.54%
c7552	2.60%	17.69%

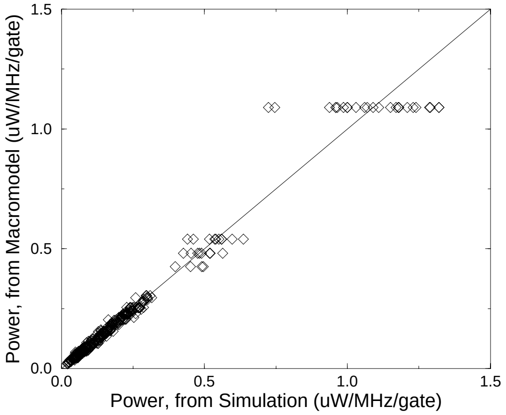

Figure 3. Agreement between the 3d table and accurate power estimation, when total power is estimated.

References

[1] W. Bowhill *et al.*, "A 300 MHz 64b quad-issue CMOS RISC microprocessor," in *ISSCC'95 Digest of Technical Papers*, pp. 182-183, Feb. 1995.

[2] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Transactions on VLSI Systems*, pp. 446-455, Dec. 1994.

[3] P. Landman, *High-level power estimation*, "International Symposium on Low Power Electronics and Design," pp. 29–35, Monterey, CA, August 12–14, 1996.

[4] M. Nemani and F. Najm, "Towards a High-Level Power Estimation Capability," *IEEE Transactions on CAD* , vol. 15 pp. 588-598, June 1996.

[5] D. Marculescu, R. Marculescu and M. Pedram, "Information Theoretic Measures of Energy Consumption at Register Transfer Level," *ACM/IEEE International Symposium on Low Power Design*, pp. 87-92, April 1995.

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

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

[8] F. Najm, "Transition Density: A New Measure of Activity in Digital Circuits," *IEEE Trans. on CAD*, vol. 12, pp. 310-323, Feb. 1993.

[9] M. Xakellis and F. Najm, "Statistical Estimation of the Switching Activity in Digital Circuits," *31st ACM/IEEE Design Automation Conference*, pp. 728-733, June 1994.

[10] H. Mehta, R. M. Owens and M. J. Irwin, "Energy Characterization based on Clustering," *33rd ACM/IEEE Design Automation Conference*, pp. 702-707, June 1996.

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

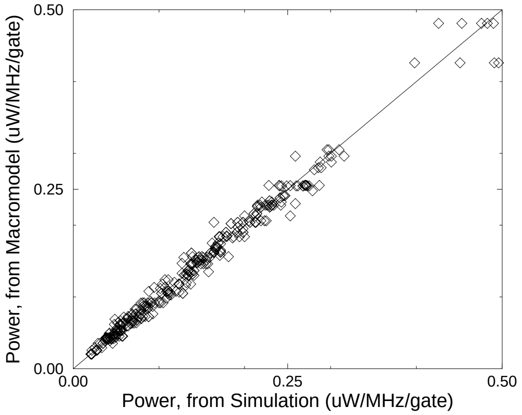

Figure 4. Agreement between the 3d table and accurate power estimation, when total power is estimated.

Table VII.

Accuracy of the 3-d Look-up Tables,
When Zero-delay Power is Estimated.

Circuit	RMS.Error	Max.Error
c432	0.33%	-7.16%
c880	0.73%	12.54%
c1908	0.24%	4.14%
c2670	0.52%	-10.13%
c3540	0.36%	-11.54%
c5315	0.55%	8.33%
c6288	0.25%	12.6%
c7552	0.83%	10.23%

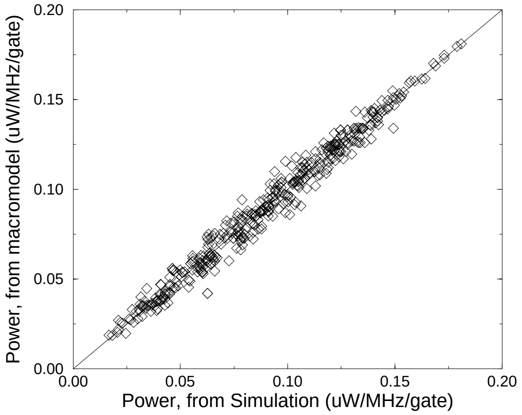

Figure 5. Agreement between the 3d table and accurate power estimation, when zero-delay power is estimated.