

Novel Modeling Techniques for RTL Power Estimation

Michael Eiermann
Institute for Integrated Circuits
Technical University of Munich
Arcisstr. 21, 80290 Muenchen, Germany
Phone +49 89 28923084
m.eiermann@ei.tum.de

Walter Stechele
Institute for Integrated Circuits
Technical University of Munich
Arcisstr. 21, 80290 Muenchen, Germany
Phone +49 89 28923862
w.stechele@ei.tum.de

ABSTRACT

In this work, we propose efficient macromodeling techniques for RTL power estimation, based only on word and bit level switching information of the module inputs. We present practicable combinations of these two properties for the construction of power macromodels. It is demonstrated, that our developed models reduce the estimation error compared to the Hamming-distance model at least by 64%. The total average errors (compared to PowerMill) achieved over a wide range of test modules and input stimuli are less than 4.6%. This is comparable to complex models, which however, have to make use of several more signal properties.

Categories and Subject Descriptors

I.6.5 [Simulation and Modeling]: Model Development - modeling methodologies.

General Terms

Design, Experimentation, Verification.

Keywords

Power estimation, power modeling, RTL macromodels, low power.

1. INTRODUCTION

In recent years, power consumption has become a key parameter in the design of integrated circuits (ICs). This is mainly due to the ever increasing integration density, which enables the functionality and the performance of ICs to improve dramatically. Higher complexity and higher performance inevitably lead to an increase of power consumption, if standard design methodologies are applied. Instead, in order to enhance the run-time of battery-operated portable applications, ICs have to be optimized with respect to power consumption. This also helps to ensure reliable operation and to reduce the cost for packaging and cooling.

In order to manage the rising complexity of today's chips, the design process has to be started on a very high level of abstraction. At those early design phases power optimization opportunities are sig-

nificantly larger than in later steps. Such optimization tasks have to be validated with respect to the yield for power reduction. For this purpose, power estimation tools are needed, but unfortunately standard tools only exist for gate level and lower levels. Estimating power at gate or transistor level is very time consuming. Therefore, a lot of techniques for high level power estimation have been proposed in the past years, most of them for the register transfer level (see [8][9][11] for a survey).

Beside characterization-free information-theoretic approaches (based only on the input-output functionality of a module) e.g. [10], the main strategy on RT level, targets on building power models for the used modules. This means, for every submodule type of an RTL design, the template power model parameters have to be investigated by performing a number of simulation experiments at lower levels of abstraction. Once the model is characterized, power estimation can be carried out by weighting the model parameters with the actual signal properties generated from running a behavioral simulation. A wide range of different approaches for power modeling can be found in literature [8][9][11].

The model's power properties are either stored into a multi-dimensional look-up table (table based) e.g. [7] or they can be expressed through an equation (equation-based) by using regression methods e.g. [3]. Further, the techniques are distinguished according to their application. In some case cumulative (average) power estimation is insufficient and power has to be modeled and estimated on a cycle-by-cycle basis [6][12]. The major difference between the approaches, however, can be seen in the kind and number of signal properties used for characterization and estimation. Nearly all models are activity-sensitive, which means power is expressed as a function of input (and output) switching activity. In order to improve accuracy, some models consider input signal probability [4], while other methods additionally use spatial correlation of the input signal [7]. Clearly, the price paid for this improvement is a higher effort for characterization and estimation.

Our approaches take into account only the input switching property, however in a specific way. We do not only consider the number of switching inputs (Hamming-distance of two consecutive input vectors), but we also regard the individual inputs, which take part in the switching. Experimental results demonstrate, that using these novel models, the estimation accuracy will be in the same range as models which, also consider other signal properties.

The remainder of the paper is structured as follows. In Section 2, our modeling approaches are described in detail. The model characterization and validation process are presented in Section 3 and 4, respectively. Results are given and discussed in Section 5. Finally, concluding remarks are provided.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'02, August 12-14, 2002, Monterey, California, USA.
Copyright 2002 ACM 1-58113-475-4/02/0008...\$5.00.

2. THE PROPOSED NOVEL MODELING APPROACH

First, we state out some assumptions about the conditions on RT level. In general, combinational modules are surrounded by registers. Thus, the input signals are treated as ideal, which means there is only one transition per bit and clock cycle, of one of the valid types {L-L, L-H, H-L, H-H}. All input transitions of a module occur at the same time and have the same rise and fall times.

2.1 Statement of the Problem

The power consumption of a module can be exactly modeled by assigning an energy value to every possible input transition and storing them in a look-up table (LUT). If V_{t-1} and V_t represent the input signal vectors in cycle $t-1$ and t , respectively, the energy E_t and power P_t in cycle t can be expressed by

$$E_t = E[V_{t-1}, V_t] \text{ and } P_t = E_t/T, \quad (1)$$

respectively, where $E[\]$ denotes the energy LUT and T the clock period. The relationship between the cycle power P_t and the average power P_{avg} for a sequence of M cycles is given by

$$P_{avg} = 1/M \cdot \sum_{t=1}^M P_t. \quad (2)$$

The number of LUT entries N for a n -bit input vector is $N = 2^{2n}$. A 16-bit input module e.g. would have 4.3×10^9 entries. Due to the effort of generating and storing the energy data, this can only be done for very small modules. Therefore, the general approach must be to reduced the size of the LUT by developing models based on more abstract signal properties.

2.2 Basic Power Dependencies

When we consider RTL modules, the inputs can be divided into control and data inputs. Concerning power dissipation, the two types of inputs behave very differently. In general, for *control inputs* the *signal state* is the decisive value for power consumption, while for *data inputs* it is their *switching activity*. Due to the usually small number of control inputs for typical RTL data path modules, we decided to use separate model parameter sets for each valid state of the control signals. Therefore, in the remainder of this paper we confine our investigations only on data inputs.

In order to further reduce the LUT entries, abstractions are conceived in the following way: Instead of considering the real signal changes, we distinguish only between *whether or not* an input bit transition takes place. For a n -bit data word, this leads to a switching word in cycle t

$$SW_t = (sb_{1,t}, sb_{2,t}, \dots, sb_{n-1,t}, sb_{n,t}), \quad (3)$$

where each $sb_{i,t}$ represents the switching of bit i in cycle t . Possible values for $sb_{i,t}$ are '1' or '0', *switching or not*. Using only this information, the number of LUT entries are reduced from 2^{2n} to 2^n , which is still too high.

Note, all these abstractions do not only reduce the modeling complexity, but they also decrease the accuracy. Therefore, there will be a trade-off between effort and accuracy!

Further abstractions of the switching vector leads to the following two alternative approaches:

2.2.1 Relating the power to the word level switching

A technique to reduce the complexity, used e.g. in [4] (beside other properties), relates the energy to the number of simultaneously switching input bits (Hamming-distance of two consecutive input vectors). The entries of the energy LUT E_w are reduced to the maximum number of switching bits $n + 1$. The energy of cycle t is easily expressed by

$$E_t = E_w[sw_t], \text{ with } sw_t = \sum_{i=1}^n sb_{i,t}. \quad (4)$$

where sw_t is the total number of word level switching bits in cycle t . In principle, the word level switching can be taken as a useful measure for average switching energy. The individual values, however, can be quite different from that, which can be seen by the comparison of the average, the standard deviation and the total range of switching energy in Figure 1. This is due to the abstraction,

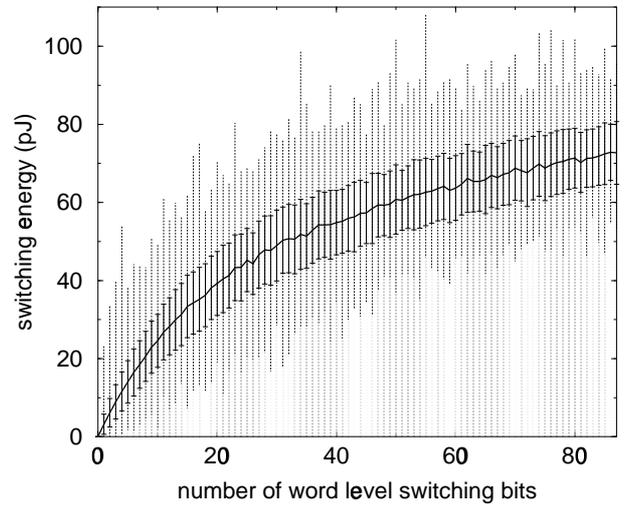


Figure 1. Average (bold), deviation (error bars) and total range (dotted) of the switching energy dependent on the word level switching for an 8x11 bit vector adder

where only the mean of the energy per number of switching bits is stored in the energy LUT, while the behavior of the individual bits involved in switching remains unconsidered. The dependence of the energy on the individual input bits is exemplified in Figure 2 for

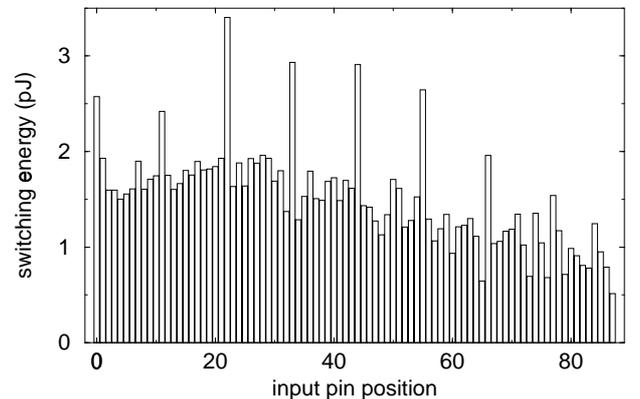


Figure 2. Energy contributions of the individual input bits for an 8x11 bit vector adder

the same module. Both diagrams have been created by performing PowerMill simulations, at least 200 cycles per word level switching number.

2.2.2 Relating the power to the bit level switching

The other alternative is to relate the energy to the switching of the single input bits (e.g. *bitwise data model* in [9]). The total energy of cycle t is given by

$$E_t = \sum_{i=1}^n sb_{i,t} \cdot E_b[i], \quad (5)$$

where, E_b and $sb_{i,t}$ denote the bit-level energy LUT and the bit-level switching of bit i in cycle t , respectively. The entries of the E_b -table are determined from a number of lower level power experiments, on which the least mean squares fitting method is applied (the values of Figure 2 have been created in this way). According to our investigations, the estimation accuracy tends to be very sensitive to the switching properties used during that characterization process. In particular, the estimation error will be acceptable, if the actual average word level switching is similar to that, applied during characterization. In Figure 3, the characterization is optimized for medium numbers of word level switching. The estimations for those switching properties are quite good, however for lower and higher values, there will be an under estimation and over estimation, respectively. This is due to the assumption in (5), that treats the single energy contributions as independent of the total number of switching pins, which is not accurate.

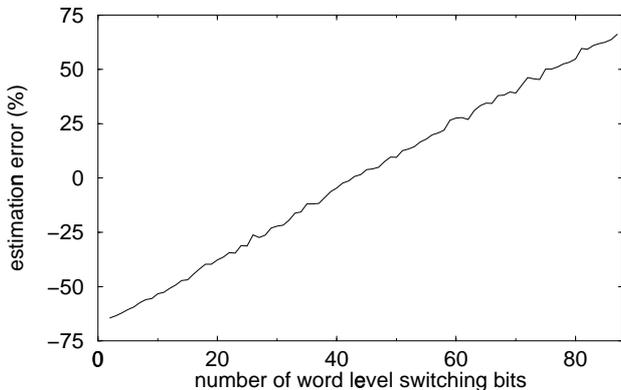


Figure 3. Estimation error only due to the difference in the average word level switching between the characterization and the estimation for an 8x11 bit vector adder

2.3 Our Modeling Approaches

As a consequence, we propose the combination of both alternatives to overcome the particular deficiencies. Thus, we constructed several models that utilize *bit* as well as *word level* switching properties. Due to the limited space, we cannot discuss all investigated combinations, but focus on the very promising techniques.

2.3.1 Subword model

The aim of this approach is to improve accuracy of the energy model relating to word level switching by subdividing all input bits into subwords or groups. For every subword the number of switching bits is evaluated separately. Therefore, each possible configuration of subword switching numbers requires an energy entry in a multi-dimensional LUT. The estimation is only based on a table look-up

according to the determined subword switching configuration for each cycle, e.g. the energy for cycle t is

$$E_t = E_{sub}[ssub_{1,t}, ssub_{2,t}, \dots, ssub_{g-1,t}, ssub_{g,t}], \quad (6)$$

where E_{sub} , $ssub_{i,t}$ and g are the subword energy LUT, the number of switching bits of subword i in cycle t , and the number of subwords, respectively.

Using the switching activity of two input buses instead of using the whole input switching has also been proposed in [5], where they additionally used the input signal probability. According to their published and our results, this model approach works well for small, regular modules (in [5] arithmetic modules with two input buses have been used). However, for modules with more than 50 input bits and more than two input buses, the estimation error increases. Further, we investigated the dependency of the accuracy on the subdividing strategies for the input pins. Among the subdivision criteria:

- A) by the input-output delay (from the static timing analysis),
- B) by the bit position on the input buses (LSB .. MSB),
- C) by the logic input buses,

we found the last one as the best by experiments.

2.3.2 Enhanced model relating to single bit switching

Similar to Section 2.2.2, this model relates the energy to the bit level switching. However, to improve the model's accuracy, an adjusting factor is used, which depends on the word level switching property. The energy consumption of cycle t is expressed by

$$E_t = c_{sgl}[sw_t] \cdot \sum_{i=1}^n sb_{i,t} \cdot E_{sgl}[i], \quad (7)$$

where, c_{sgl} and E_{sgl} are the LUTs for the adjusting factors and the bit level energy, while sw_t and $sb_{i,t}$ denote the word and bit level switching properties for cycle t , respectively. Each energy LUT entry is determined during the characterization process by the average of a number of single bit switching experiments for the corresponding bit, where the states of the remaining bits differ. Single bit switching has been chosen, because this allows the strongest distinction between the different bit switching energy contributions (cf. Figure 4 vs. Figure 2). The LUT for the adjusting factors for each word level switching number is simply determined by replacing E_t in (7) by the true energy $E_{act,t}$ and solve the equation for

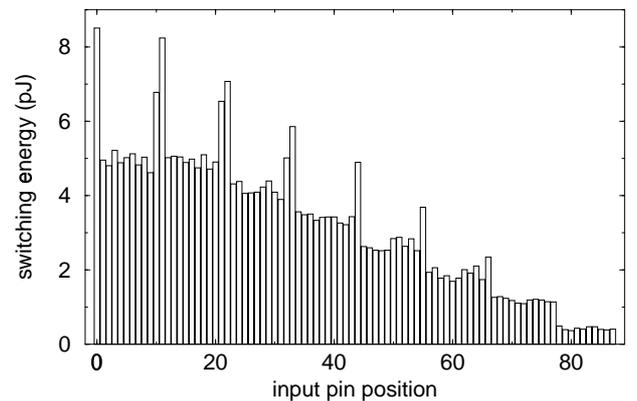


Figure 4. Single bit switching energies dependent on the input bit position for an 8x11 bit vector adder

the corresponding c_{sgl} entry. The mean for a number of experiments is taken as the coefficient (see Figure 5). However, it has

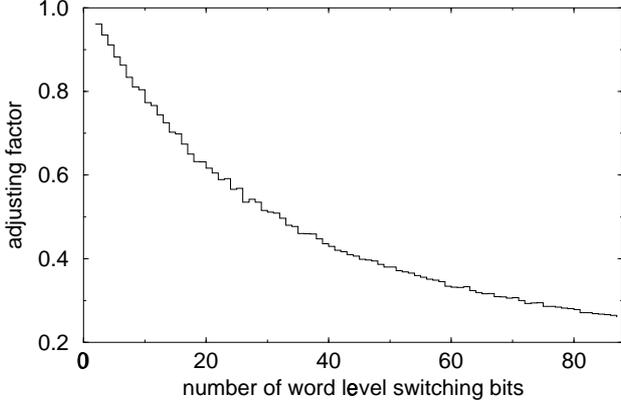


Figure 5. Adjusting factors for an 8x11 bit vector adder

been found out, that the bit-level energy contributions have to be adjusted differently, depending on their energy quantity. Considering these observations, the following equation based on higher order expression of the energy coefficients improves the model.

$$E_t = \sum_{o=1}^k \left(c_{sgl,o}[sw_t] \cdot \sum_{i=1}^n (sb_{i,t} \cdot E_{sgl}[i])^o \right). \quad (8)$$

The adjusting factors in LUT $c_{sgl,o}$ for the k orders can be found by applying the standard least mean squares method. The accuracy improvements due to the use of higher order energy coefficients are shown in Table 1. A large number of test modules and test sequences have been used to calculate these average errors. For the test setup, see Section 4. A value of $k = 2$ or 3 has been found to be satisfactory and leads to improvements of at least 10%.

Table 1: Improvements through the introduction of the higher order equation

equation orders	1st	2nd	3rd	4th
average errors	5.26	4.73	4.50	4.80
improvements to 1st order		10%	14%	9%
maximum errors	8.91	7.12	8.25	8.46

2.3.3 Enhanced model relating to bit pair switching

A slight drawback of the previous model can be seen in the fact, that no interdependencies between switching bits are considered. This can be obtained by relating the energy to input bit pairs as opposed to single bits. We modify the term in (8) to the following formulation

$$E_t = \sum_{o=1}^k \left(c_{pair,o}[sw_t] \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^n (sp_{ij,t} \cdot E_{pair}[i,j])^o \right), \quad (9)$$

where $sp_{ij,t}$ becomes ‘1’ only if bit i and bit j switch at the same time. $E_{pair}[i,j]$ represents the energy coefficient for the same switching pair. These energy coefficients and the adjusting factors in LUT $c_{pair,o}$ are determined similarly to the process described in Section 2.3.2. The improvements resultant of using higher order equations are about 10% (cf. Table 2).

Table 2: Improvements through the introduction of the higher order equation

equation orders	1st	2nd	3rd	4th
average errors	4.41	3.89	4.03	3.96
improvements to 1st order		12%	8%	10%
maximum errors	8.81	5.30	6.87	6.89

2.3.4 Enhanced regression model

To determine the energy coefficients the model is based on, we use the well known linear regression method [2], however in an enhanced manner for the additional consideration of the word level switching properties.

For each number of simultaneously switching bits we perform a separate least mean squares fitting. The energy equation for cycle t is given by

$$E_t = \sum_{i=1}^n sb_{i,t} \cdot E_{reg,2D}[sw_t, i], \quad (10)$$

with $E_{reg,2D}[sw_t, i]$ as the 2-dimensional LUT for the energy coefficient depending on the word level switching number sw_t and the switching bit i .

Since for every value of word level switching bits a number of low level power experiments has to be performed, the characterization effort increases. This can be reduced by the following process:

First, the energy coefficients $E_{reg,lin}[\]$ are determined with standard regression methods (LMS fitting). This is done with experiments, where pseudo random patterns are applied, that cover equally the whole range of switching properties, bit level as well as word level. In a second step, for each word level switching number sw_t the adjusting factor $c_{reg,o}$ is calculated as described in Section 2.3.2. The energy equation

$$E_t = \sum_{o=1}^k \left(c_{reg,o}[sw_t] \cdot \sum_{i=1}^n (sb_{i,t} \cdot E_{reg,lin}[i])^o \right) \quad (11)$$

has the same structure as (8). Note, the energy coefficients in $E[\]$ are different to the approach in Section 2.3.2, because they are determined differently. Also, the adjusting factors differ.

3. MODEL CHARACTERIZATION

As mentioned above, the model coefficients are determined once within the characterization process, in which the modules are stimulated by well defined characterization patterns. Lower level power estimators (gate or transistor level) are used to ascertain the energy for each initiated input transition. For that, we use Synopsys PowerMill, because of its capability for cycle based current estimation.

The characterization pattern generation represents a critical task. For this reason, we constructed a special sequence synthesizer written in C. Its main properties are listed below:

- the number of experiments per coefficient to be characterized can be chosen according to its relevance.
- in order to cut down simulation time, all transitions are arranged in one continuous stream.
- for the models based on LMS methods, the solvability of the system of equation is proved in advance.

- for every number of (sub)word level switching bits, the single inputs are covered equally.
- the order of the pattern is chosen pseudo randomly, in order to prevent similar signal probabilities for the same number of (sub)word level switching bits.

The effort for the characterization is shown in Table 3. It has been found that about 10 experiments are sufficient for most coefficients. For the models applying adjusting factors, which are calculated using the energy LUT (e.g. single bit switching energy), some more experiments per entry for these underlying energy LUTs were performed (10..100).

Table 3: Effort for the characterization

section	model type	number of coefficients
2.3.1	subword model	$(n/g + 1)^g$
2.3.2	relating to single bit sw.	$k(n - 2) + n + 2$
2.3.3	relating to bit pair sw.	$k(n - 2) + n(n - 1)/2 + n + 2$
2.3.4	enhanced regression using equation (10)	$n(n - 1) + 2$
	using equation (11)	$k(n - 2) + n + 2$
n : input number; g : subword number; k : order of equation		

4. MODEL VALIDATION

To assess the estimation accuracy of the proposed models, comparisons to PowerMill simulations have been performed for a large number of module types and input sequences. Thus, 21 testmodules have been taken partly from real designs and partly from the Synopsys DesignWare (DW). The properties of the modules are summarized below:

- gate equivalents: 41 .. 3136
- number of data pins: 16 .. 88
- number of data buses: 2 .. 9
- number of control inputs: 0 .. 2

In order to test the models in an extreme way, we synthesized a number of different input test streams for every module, each containing 1000 input patterns. These input sequences completely differ from those used for the characterization. We also made use of the logic inputs buses to which different switching activities were applied. The average switching activities were either the same for all bits of a bus, or they were linearly distributed, ranging from 50(95)% at the LSB to 25(5)% at the MSB (to have realistic test conditions). We also chose very extreme cases where only one, only two or all but one buses were switching. Four main types of test streams have been used. The individual streams of each type were distinguished by their switching properties, which are given below:

- switching activities of logic input buses are distributed for LSB..MSB: 50%..25% or 95%..5% (2 streams)
- only one or all but one logic input buses are switching; others remain nearly stable; bus activities: 25%, 50% or 75% equal for all bits of a bus or distributed for LSB..MSB: 50%..25% (8..56 streams)
- only two logic input buses are switching; bus activities distributed for LSB..MSB: 50%..25% (1..36 streams)

- average switching activities are equal for all bits 10%, 20%, .. , 80% or 90% (9 streams)
(LSB/MSB denotes least/most significant bit of a logic bus)

The number of streams of each type differ according to the number of the modules' input buses and the possible configurations resultant from that.

For every stream-module-model combination, we calculated a separate average relative error ϵ_{avg} compared to PowerMill using the following formula:

$$\epsilon_{avg} = \frac{P_{avg} - P_{PowerMill}}{P_{PowerMill}}, \quad (12)$$

where ϵ_{avg} is the average power for the stream calculated based on the our model equations in Section 2.3.

In order to prevent the compensation of positive and negative errors, the absolute values of the relative error ϵ_{avg} of each stream is taken to calculate the mean error of all streams S for every module-model combination,

$$\epsilon_{mean,abs} = 1/S \cdot \sum_{i=1}^S |\epsilon_{avg,i}|. \quad (13)$$

The power estimation effort are shown in Table 4, where the necessary integer and floating point operations are presented. It can be seen, that the enhanced model relating on bit pairs switching has the highest computational effort, while the other are approximately equal.

Table 4: Effort for the power estimation

section	operations per cycle:		only once for a sequence:
	compare	increment	floating point add and mult
2.3.1	n	$(s + 1)$	$(n/g + 1)^g$
2.3.2	n	$2s$	$n(n - 1) + 2$
2.3.3	n	$s(s + 1)/2$	$(n - 2)(n - 1)n/2 + n + 2$
2.3.4	n	$2s$	$n(n - 1) + 2$
n : input number; g : subword number; s : switching number			

5. RESULTS AND DISCUSSION

The functionality, the number of input pins and input buses as well as the size (in gate equivalents) of the test modules are given in Table 5 on the left. In the right columns, results are presented for the two basic models from Section 2.2.1-2 (for reference) and for our proposed hybrid models from Section 2.3.1-4. The mean estimation errors for each module-model combination correspond to the models in the following manner:

- r1* based only on word level switching (for reference)
- r2* based only on bit level switching (for reference)
- 1a** subword model (2 subwords)
- 1b** subword model (3 subwords)
- 2** relating to single input bit switching (3rd order equation)
- 3** relating to input pair switching (2nd order equation)
- 4** enhanced regression model

From the table, it can be seen, that using any of our novel models 2..4, the average estimation error has been reduced at least

Table 5: Design properties of test modules and estimation results for different models

description of the test modules			estimation errors $\epsilon_{mean,abs}$ in % for models						
type	input# (bus#)	size	r1	r2	1a	1b	2	3	4
DesignWare ripple-carry adder	16 (2)	41	5.0	15.6	4.4	3.4	5.0	4.3	5.1
	24 (2)	62	5.8	16.3	3.6	3.6	4.7	4.2	4.5
	32 (2)	83	3.5	15.6	2.0	4.3	3.6	3.2	3.6
DesignWare carry-look-ahead adder	16 (2)	54	5.7	12.6	3.8	4.5	5.3	3.9	3.9
	24 (2)	83	5.2	12.7	3.6	3.6	4.6	3.8	3.8
	32 (2)	119	4.0	15.0	1.8	3.6	3.6	3.1	3.4
DesignWare carry-save multiplier	16 (2)	436	5.6	16.1	3.8	4.5	4.8	4.5	4.4
	24 (2)	1035	4.1	15.9	3.6	3.9	3.8	3.9	3.1
	32 (2)	1681	5.3	15.3	6.3	7.2	3.1	3.3	2.7
DesignWare wallace-tree multiplier	16 (2)	512	9.1	18.0	3.3	3.6	4.2	3.0	4.0
	24 (2)	1056	8.5	22	2.2	3.9	6.1	3.4	6.1
	32 (2)	1709	9.8	24	2.4	4.7	8.3	5.2	7.9
DW duplex-comp.	64 (4)	173	7.3	8.2	4.8	4.5	2.5	2.1	2.9
9,32 bit accu.	41 (2)	173	35	9.2	16.3	21	2.1	4.7	2.7
median of 3 words	48 (3)	413	14.0	10.6	12.7	13.4	3.7	3.4	5.4
median (fast impl.)	48 (3)	972	12.2	10.0	10.7	11.4	3.0	3.5	4.6
2x2 mux, cmp, inc	58 (4)	202	7.4	9.2	7.6	2.9	5.8	4.2	5.3
2x2 sub_add, cmp	43 (4)	391	18.6	25	20	9.9	6.6	5.2	6.8
2x2 sub_abs, add	48 (4)	479	3.3	22	4.0	3.6	3.0	2.7	2.9
8 word vectoradd.	88 (8)	502	15.0	28	9.7	7.4	6.0	5.3	5.4
min/med/max of 9	81 (9)	3126	19.3	25	21	20	5.0	4.9	8.5
21 modules average			12.9	16.4	10.3	6.7	4.5	3.9	4.6
21 modules maximum			74	28	76	21	8.3	5.3	8.5

by 64% compared to the well known Hamming distance model (word level switching from Section 2.2.1) in column *r1*. This reduction has been achieved only by considering the combination of both, bit and word level input switching properties. Further more, it can be observed, that the improvements are small for regular modules (DW with two input buses), however, for complex modules (lower part of the table) the refinements are immense. Also the maximum errors of all modules for our models 2..4 are very low. Compared to the second reference *r2* the improvements are quite higher, but this is, as mentioned in Section 2.2.2, because the accuracy of this model based only on bit level switching is very sensitive to the characterization patterns (a median number of average word level switching bits has been taken).

The subword models 1a,b (cf. Section 2.3.1) only cause slight enhancements and have high characterization effort. The results can be improved by dividing the inputs into more than three subwords. However, according to the exponential growth of the LUT entries with the number of subwords (see Table 3), the grid of these LUT has to be widened.

If we take into account the effort for characterization and estimation, both models (enhanced model relating to single input bit switching and enhanced regression model) achieve the best trade off, for all test modules. The estimation error compared to *r1* can be reduced by up to 70% using the model 4 (relating to input pair switching) at the cost of enhanced estimation effort.

In all, these three novel approaches based on input switching information only, cause estimation errors less than 4.6% on average over a wide range of test modules and input stimuli. These results are comparable to recently published enhanced models [1][3][5][7], which achieve average estimation errors of about 3..15%, but they have to make use of several more signal properties (e.g. output switching, input probabilities, etc.). A combination of our efficient modeling techniques with those additional properties can increase the estimation accuracy, but each property leads to an additional dimension in the LUTs, which would result in a multiple effort, particularly for the characterization.

6. CONCLUSIONS

It has been shown, that using both, word and bit level switching information of module inputs for macromodeling without other signal properties, the estimation error compared to the Hamming distance model can be reduced by up to 70%. Total errors less than 4.6% on average for a large number of test modules and input stimuli have been achieved. This is comparable to complex models based on several more signal properties.

7. REFERENCES

- [1] M. Anton, I. Colonescu, E. Macii, M. Poncino, "Fast Characterization of RTL Power Macromodels," in IEEE Proc. of ICECS, pp. 1591-1594, 2001.
- [2] L. Benini, A. Bogliolo, M. Favalli, G. De Micheli, "Regression models for behavioral power estimation," in Proc. of PATMOS, pp. 179-187, 1996.
- [3] A. Bogliolo, L. Benini, G. de Micheli, "Regression-Based RTL Power Modeling," ACM Trans. on Design Automation of Electronic Systems, vol. 5, no. 3, pp. 337-372, July 2000.
- [4] G. Jochens, L. Kruse, W. Nebel, "A New Parameterizable Power Macro-Model for Datapath Components," in Proc. of European Design & Test Conference, Date, pp. 29-36, 1999.
- [5] G. Jochens, L. Kruse, E. Schmidt, A. Stammermann, W. Nebel, "Power Macro-Modelling for Firm-Macro," in Proc. of PATMOS Workshop, Germany, pp. 24-35, Sep. 2000.
- [6] S. Gupta, F.N. Najm, "Energy-per-cycle estimation at RTL," in Proc. ISLPED, Monterey, CA, pp.121-126, 1999.
- [7] S. Gupta, F.N. Najm, "Power Modeling for High-Level Power Estimation," in IEEE Trans. on VLSI, vol. 8, no. 1, pp. 18-29, February 2000.
- [8] P. Landman, "High-Level Power Estimation," in IEEE Proc. of ISLPED, Monterey, CA, pp. 29-35, June 1996.
- [9] E. Macii, M. Pedram, F. Somenzi, "High-Level Power Modeling, Estimation, and Optimization," in IEEE Transactions on CAD, vol. 17, no. 11, pp. 1061-1079, Aug. 1998.
- [10] D. Marculescu, R. Marculescu, M. Pedram, "Information theoretic measures for power analysis," in Trans. on CAD, vol. 15, no. 6, pp. 599-610, 1996.
- [11] A. Raghunathan, N.K. Jha, S. Dey, High-Level Power Analysis and Optimization, Kluwer Academic Publishers, Boston/Dordrecht/London, 1998.
- [12] Q. Wu, Q. Qiu, M. Pedram, C.S. Ding, "Cycle-Accurate Macro-Models for RT-Level Power Analysis," in Trans. on VLSI 1998, vol.6, no.4, pp. 520-528, 1998.