Accurate Power Estimation Using Circuit Partitioning

Ana T. Freitas
atf@inesc.pt
IST-INESC
Lisbon, Portugal

Arlindo L. Oliveira
aml@inesc.pt
IST-INESC/CEL
Lisbon, Portugal

Horácio C. Neto
hcn@inesc.pt
IST-INESC
Lisbon, Portugal

Abstract

Recently, a probabilistic approach that uses a simple but powerful formalism for exact power estimation taking into account word level input correlations has been proposed.

This paper describes a circuit partitioning technique that can be used to speed up this method, that is otherwise limited to small circuits. By using partitioning techniques, the method does not require the computation of global BDD representations for node functions, thereby extending considerably its range of applicability. Moreover, the partitioning maintains the full set of correlations and, therefore, does not induce any loss of accuracy.

By applying the approach proposed in this paper, it is possible to compute exactly the power dissipation of combinational modules using input statistics that would require extremely large traces if simulation based methods were used.

1 Introduction and related work

This work addresses the problem of estimation of the power dissipated in combinational modules under general input correlation statistics.

The basic model used by most power estimation techniques for a functional module is of the form:

$$P = C_{eff} V_{DD}^2 f \quad (1)$$

where V_{DD} is the voltage of the power supply of the module and f is the clock frequency at which it operates. The factor C_{eff} is the average switched capacitance of the functional unit.

Existing approaches for power estimation can be divided into two categories: simulation based methods and probabilistic methods.

Simulation based methods can model very accurately the spatial and temporal correlations present at the input and internal signals, but require the existence of a trace that accurately matches the conditions of operation of the system. These methods can be very inefficient if the trace is very long.

In order to speed up this computation, a number of methods have been proposed which aim at reproducing accurately the input correlations with traces that are much smaller than the initial trace. One such possibility is random sampling where only a subset of the transitions present in the trace is used. More sophisticated methods include sequence compaction [5] and sequence synthesis.

Probabilistic methods do not require the existence of a trace and can be used to characterize circuits in a manner that is independent of the inputs presented. However, the most efficient approaches presented to date allow only for very simplified models of temporal and spatial correlations, and, therefore, obtain estimates of limited accuracy. The most common simplification considers the primary inputs uncorrelated in time and space [4], but this represents, in general, a very crude approximation.

Several methods that aim at modeling spatial and temporal correlations more accurately have been presented. One such method [6] models the pairwise spatial correlations of the primary inputs and propagates them through the circuit. This process only involves local calculations, and therefore can be applied to larger circuits, but does not control the total error introduced in successive computations.

Another approximation that has the potential to model accurately spatial and temporal correlations is based on the use of *algebraic decision diagrams* (ADDs) to characterize the dissipation incurred by each module under each particular input transition [1]. The main problem of this method is that ADD representations can easily become unmanageably large. The circuit partitioning technique introduced in this work can also be applied to ADD based approaches, with the objective of reducing the overall computational complexity.

A method that is able to compute the power dissipated taking into account the temporal and spatial correlations at the inputs has been presented recently [3]. However, as presented there, it requires the computation of the BDD representation of all internal nodes, specified as a function of the primary inputs of the circuit. For large circuits, and for some special families of circuits like, for instance, multipliers, it is impossible to compute a global BDD representation of such a large size. Therefore, the applicability of this method was restricted to relatively small circuits.

In this work, we propose a circuit partitioning technique that effectively replaces the one global BDD representation by an appropriate number of partial BDDs of

limited size. This technique provides a natural extension of the methodology proposed in [3] while keeping the full set of correlations required to maintain the accuracy of the estimation. Therefore, its extends significantly the range of applicability of the former method by making it applicable to larger circuits, without any loss of accuracy.

The remainder of this paper is organized as follows. Section 2 describes the method proposed in [3] which represents the base of the current work. Section 3 contains the main contribution of this work and shows how the computation described in the previous section can be performed much more efficiently using circuit partitioning. Section 4 presents the experimental results obtained and section 5 presents the preliminary conclusions and proposes future work.

2 Input correlation modeling and power estimation

2.1 Input correlation modeling

Consider a combinational circuit with n input nodes and m internal nodes. The variables that correspond to the input nodes are $\{x_1, \ldots x_n\}$, while the circuit internal nodes are represented by variables $\{x_{n+1}, \ldots x_{n+m}\}$. A subset of the internal nodes are also output nodes of the circuit. For each node i, we will use two variables, one representing the value $b$efore the transition (x_i^b) and the other one representing the value $a$fter the transition (x_i^a).

The input correlation modeling proposed in [3] is based on the definition of a set of functions F_i and associated probabilities P_i. F_i, defined over the variables $X = \{x_1^b, \ldots x_n^b, x_1^a, \ldots x_n^a\}$ represents all transitions that take place with probability P_i (a real number between 0 and 1, and satisfying $\sum_i P_i = 1$).

By specifying a set of pairs (F_i, P_i) complex temporal correlations between input words are accurately modeled. Each F_i is specified using the following formalism: for each possible transition, a description of the input word before and after a transition is given; individual bits in the word can be represented by:

- `0` : The bit takes the value 0
- `1` : The bit takes the value 1
- `-` : The bit takes any value.
- `.` : The bit keeps its value after the transition.
- `#` : The bit changes value after the transition.

As an example, a 4 bit Gray code and a 4 bit binary code are modeled by the specifications of the F_i's shown in table 1.

The functions F_i are manipulated using a standard *binary decision diagram* (BDD) package [2].

2.2 Computation of dissipated power

The computation of the power dissipated in the circuit using the inputs statistics specified as above can be viewed as a generalization of the work by Ghosh [4]. His approach is based on the computation, for each node, of a function (represented by a BDD) that describes the conditions under which that node switches. With this function it is possible to evaluate the dissipated power by computing the number of transitions in internal nodes. However this computation is straightforward only if one assumes that all transitions at the inputs have equal probability. If the input transitions have different probabilities one needs to compute a sum over all possible paths in the BDD.

Gray			Binary		
`...#`	`...#`	0.500	`...0`	`...1`	0.500
`..#.`	`..#.`	0.250	`..01`	`..10`	0.250
`.#..`	`.#..`	0.125	`.011`	`.100`	0.125
`#...`	`#...`	0.125	`#111`	`#000`	0.125

Table 1: Specification of input word statistics for Gray and Binary distribution codes.

Our approach avoids the need to explicitly sum over all the paths in the BDD by using the functions F_i and the associated probabilities P_i, defined in the previous section. To use this information in an effective way, a single function that characterizes the circuit is computed. This function, the Transition Consistency Function (TCF) depends on the input and internal signals at time t and time $t+1$ and represents all possible transitions of the circuit under analysis.

We are interested in the computation of the dissipated power when the input variables change from $\{x_1^b, x_2^b \ldots x_n^b\}$ to $\{x_1^a, x_2^a \ldots x_n^a\}$. For simplicity, we will consider only the zero delay model.

Since there exists a total of $2(n+m)$ variables, the TCF specifies a probability distribution defined over a space of dimension $2^{2(n+m)}$. This function is defined over the space $X \cup Y$, where $Y = \{x_{n+1}^b, \ldots x_{n+m}^b, x_{n+1}^a, \ldots x_{n+m}^a\}$.

The TCF is a function $\mathcal{T}(X,Y)$ that describes all possible transitions. We define $\mathcal{T}(x_1^b, \ldots, x_{n+m}^a)$ to be 1 if the values of the variables in the circuit $\{x_1, \ldots, x_{n+m}\}$ may experiment a transition from $x_1^b, x_2^b, \ldots, x_{n+m}^b$ to $x_1^a, x_2^a, \ldots, x_{n+m}^a$.

Consider the circuit with two inputs and two outputs, as shown in figure 1, and the correspondent list of all possible transitions.

In this circuit, it is possible to observe a transition in the variables x_1, x_2, x_3, x_4 from the combination $1,0,0,0$ to the combination $1,1,1,1$. Therefore, the minterm 10001111 exists in the TCF.

Let the function at internal node i be $f_i(x_1, \ldots, x_n)$. The function $\mathcal{T}(X,Y)$ can now be computed as

$$\mathcal{T}(X,Y) = \prod_{j=a,b} \prod_{i=n+1}^{n+m} x_i^j \equiv f_i(x_1^j, \ldots, x_n^j) \qquad (2)$$

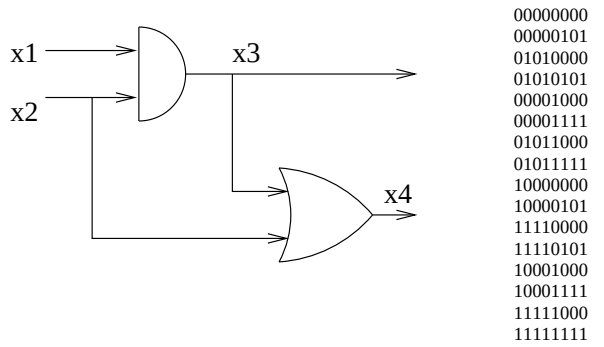

Figure 1: Example of a two input/two output circuit and the list of minterms in the Transition Consistency Function.

The dissipated power can now be easily computed using expression

$$C_{eff} = \sum_k C_k T_k \quad (3)$$

where T_k represents the total number of transitions in node k and is given by

$$T_k = \sum_i |(x_k^b \oplus x_k^a)\mathcal{T}(x_1^b, \ldots, x_{n+m}^a)F_i| \times \frac{P_i}{|F_i|} \quad (4)$$

where $|F_i|$ denotes the number of minterms in F_i.

By exchanging the summations, one can re-write the total power as:

$$C_{eff} = \sum_i \frac{P_i}{|F_i|} \sum_k C_k |(x_k^b \oplus x_k^a)S_i| \quad (5)$$

where S_i is given by:

$$S_i = \mathcal{T}F_i \quad (6)$$

The computation of the inner summation of expression (5) is performed with a single pass over the BDD defining S_i.

The method described takes into account only functional transitions, i.e., transitions between steady states of the signals. This is equivalent to the use of a zero-delay model for all the gates in the circuit. However, power dissipation due to spurious transitions is also important and may account for a significant fraction of the total dissipated power.

A generalization of this method to handle non-zero delay models has been described in [3]. The mechanism proposed to handle generic delay models does not change the structure of the method and therefore the application of the circuit partitioning technique, described in this paper, to the generic method version is straightforward.

3 Power computation using circuit partitioning

This section describes a circuit partition technique that avoids the need to compute global BDDs and keeps the full set of correlations between sub-circuits. Therefore, the application of this technique considerably extends the range of applicability of the method described in the previous section, while maintaining its high level of accuracy.

Consider an arbitrary combinational module M, with n inputs and p outputs, as depicted in figure 2. In this

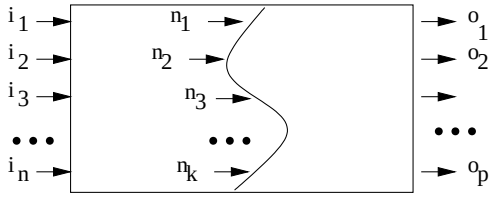

Figure 2: Combinational circuit module and cutset corresponding to nodes $\{n_1, \ldots, n_k\}$.

figure, the nodes $\{n_1, \ldots, n_k\}$ constitute a cutset of the circuit, i.e., a set of internal nodes that break every path from input to output.

Under these conditions, the computation of the dissipated power can be performed by computing the TCF of two simpler combinational modules:

- M_1 with inputs $\{i_1, \cdot\cdot, i_n\}$ and outputs $\{n_1, \cdot\cdot, n_k\}$
- M_2 with inputs $\{n_1, \cdot\cdot, n_k\}$ and outputs $\{o_1, \cdot\cdot, o_p\}$

For both M_1 and M_2, the BDD of each node is computed as a function of the sub-circuit inputs, thereby avoiding the need to compute the BDD for each node in M as a function of the primary inputs of the circuit.

Given the TCF $\mathcal{T}$ for block M_1, it is possible to compute the functions F_i^2 for block M_2 that are used to compute the power dissipated in block M_2 (using expression 5).

Recall that the functions F_i describe the correlations at the primary inputs of the circuit, and therefore the correlations at the inputs of block M_1. To compute the correlations at the inputs of block M_2, one needs to compute the functions F_i^2 that depend on the variables $\{n_1, \ldots, n_k\}$. Now, each F_i is transformed by block M_1 in accordance with the permissible transitions specified in its TCF. Therefore, the functions F_i^2 are obtained by computing the conjunction of the TCF of M_1 with functions F_i, and smoothing the internal variables of block M_1.

To formalize the procedure, we introduce some additional notation. Let the original circuit be partitioned in a sequence of circuits M_h, such that the outputs of M_{j-1} are connected to the inputs of M_j. Let X_j be the set of input variables ($\{x_1^b, \ldots x_n^a\}$) for circuit M_j, Y_j be the set of internal variables ($\{x_{n+1}^b, \ldots, x_{n+m}^a\}$) and Z_j be the subset of Y_j that corresponds to output nodes of M_j. Finally let $W_j = Y_j \setminus Z_j$ denote the set of variables that correspond to purely internal nodes of circuit M_j.

The set of functions F_i^h that is required to compute equation 4 can now be computed using the following ex-

pressions:

$$
\begin{aligned}
&F_i^1(X_1) = F_i(X_1) \\
&F_i^2(X_1, X_2) = [X_2 \leftarrow Z_1]\exists_{x \in W_1} \mathcal{T}^1(X_1, Y_1) F_i^1(X_1) \\
&F_i^{j+1}(X_1, X_{j+1}) = [X_{j+1} \leftarrow Z_j]\exists_{x \in X_j \cup W_j} \\
&\qquad\qquad \mathcal{T}^1(X_j, Y_j) F_i^j(X_1, X_j)
\end{aligned} \tag{7}
$$

Note that functions F_i^j depend on both the primary input variables and the input variables of circuit M_j.

Since the partition strategy can be applied recursively until blocks of reasonable size are obtained, the computation of the TCF represents no longer the limiting factor for this method. Note, however, that the size of the BDDs that can be obtained is not guaranteedly limited, since the computation of the F_i^j, for an arbitrary j can generate BDDs with a large number of nodes. By using a known result [7] that the size of a BDD with k minterms and n input variables cannot exceed kn nodes, it is easy to obtain the following bound on the size of the largest BDD generated in the process of computing the values of F_i^j:

$$\textsc{Bdd\_Size}(F_i^j) \leq |X_j||F_i^1| \tag{8}$$

where $|X_j|$ represents the number of variables in X_j and $|F_i^1|$ represents the number of minterms in the functions specifying the primary input correlations. This bound implies that the maximum size of the BDDs reached depends only on the size of the cutsets obtained and the number of minterms of the functions specifying the original input correlations.

3.1 Algorithm for circuit partitioning

As illustrated in figure 2, the objective is to partition the network in a number of smaller networks such that each network has a relatively small number of nodes and, as a whole, the circuit functionality is equivalent to the original. Given a maximum number of nodes for each circuit, $N_{\max}$ and a circuit M_0 to be partitioned, we obtain a list of circuits L by the following algorithm:

1. $L \leftarrow \{M_0\}$
2. For each $M_i \in L$ such that $\textsc{Size}(M_i) > N_{\max}$ do:
   Find a set of nodes $(\{n_1^i, \ldots, n_k^i\})$ that is a cutset.
   Partition circuit M_i in two circuits M_k and M_j.
   $L \leftarrow L \cup \{M_k, M_j\} \backslash M_i$

From expression 8, it is known that the size of the BDDs involved will depend on the number of nodes in the cutset. For that reason, at each step, a minimum cutset of the circuit is obtained using the maxflow/mincut algorithm [9], with the nodes being weighted using

$$W_k = |D_I(n_k) - D_O(n_k)| \tag{9}$$

where $D_I(n_k)$ represents the distance from node k to primary inputs and $D_O(n_k)$ represents the distance from node k to primary outputs. This cost function guides the maxflow/mincut algorithm to cut the circuit using preferentially the nodes that are at the same distance from the inputs and the outputs, i.e., cut through the middle of the circuit.

As described, the partition algorithm recursively breaks the circuit in two parts by finding a set of nodes that is a cutset of the circuit. Although expression 9 gives preference to cuts that include only internal nodes, in some cases primary inputs or primary outputs may also have to be included, as exemplified in figure 3. In these cases buffers are inserted after the primary inputs or before the primary outputs as shown in figure 4. In this example the

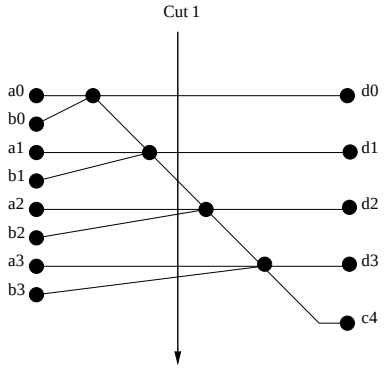

Figure 3: Cut example using a four bit adder circuit.

original circuit gets partitioned into two circuits where a set of additional buffers is inserted. In some cases a large

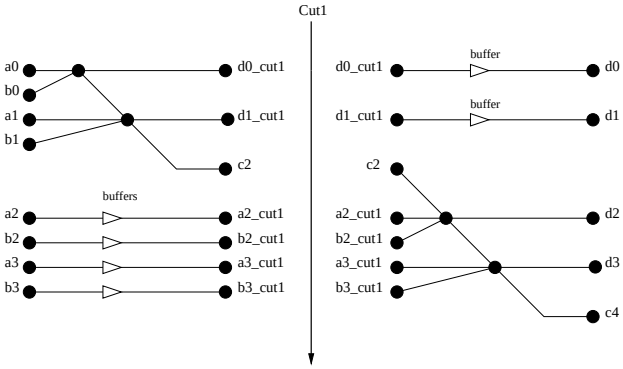

Figure 4: Circuits obtained after partition.

number of buffers may have to be inserted, leading to a significant increase on the actual number of nodes in the circuit. However, a simple analysis shows that this increase on the circuit size has only a moderate impact on the size of the BDD required to represent the TCF.

Consider a buffer with an input x_i and an output x_o. The TCF that represents that buffer is given by $(x_i^b \equiv x_o^b) \wedge (x_i^a \equiv x_o^a)$. If the BDD variable ordering is optimum, that is $(x_i^b, x_o^b, x_i^a, x_o^a)$, only 6 nodes are inserted in the BDD to represent the buffer function. Apart from these extra nodes, i.e., 6 BDD nodes per inserted buffer, there is no other impact on the size of the TCF required to describe the partitioned circuit. Note that these new variables must be inserted, in the BDD, before or after all

other circuit variables.

4 Results

This section presents preliminary results obtained using the approach described in section 3. To evaluate the applicability of the method the algorithm was integrated within the SIS logic synthesis system [8]. This preliminary implementation of the algorithm does not yet use the optimum ordering for the inserted buffer variables referred to in section 3.1. Instead, the ordering used is obtained by the dynamic reordering routines of the BDD package. For this reason the performance of the algorithm is still very far from what we expect to obtain in the final prototype.

As test cases, we used a set of circuits that include the MCNC 91 benchmark and some additional ALU type circuits. These circuits were mapped to the MSU library before power estimation. All tests were performed on a 300MHz Pentium running Linux with a CPU time limit of 1 hour and memory usage restricted to 90 MB. The partition algorithm was instructed to create partitions with a maximum of 25 nodes.

Table 2 lists the test circuits for which it was possible to compute all the TCFs. In this table, columns 2, 3 and 4 show the statistics of the circuits, namely the number of primary inputs, the number of primary outputs and the number of literals. Column 5 shows the number of sub-circuits generated by the partition process, column 6 the maximum BDD size of the TCF's obtained in the process and column 7 the BDD size of the TCF of the circuits when no partitions are used. Note that in this experiment the objective is not to estimate the power of the circuits, but to prove that the partitioning algorithm effectively extends the size of the circuits that can be handled.

The results presented in table 2 show that, despite the sub-optimum ordering used, the size of the TCF's created in the process is, in general, smaller than the one obtained with the original algorithm. Note that due to the different orderings used in the partitioned and un-partitioned case, in some examples the TCFs are actually larger in the partitioned circuit. However, in the larger examples, the difference in the size of the TCFs is important enough to allow the new algorithm to complete while the original fails.

In table 3 the power estimation method was applied to these circuits using three different input statistics specified by three different sets of F_i's. In some cases it is still not possible to compute the dissipated power even though it is possible to compute the TCF for each sub-circuits. The table includes only the circuits for which it was possible to compute the power for at least one input distribution.

The first set (Binary) of F_i's specifies a binary encoding at the inputs of the circuits. The second set (High/Low) specifies an encoding where two input bits exhibit high activity while the remaining input bits are stationary[1]. Finally the last set (Uniform) of F_i's specifies an uniform distribution where all transitions are equally probable.

Circuit	PI	PO	Lits	N	Max TCF	TCF orig
ex2	3	3	6	1	58	58
C17	5	6	12	1	792	792
majority	5	5	16	1	325	325
b1	3	6	20	1	311	311
cm82a	5	9	29	1	614	614
cm42a	4	13	35	1	178	178
cm138a	6	10	36	1	1464	1464
cm152a	11	7	36	1	195	195
cm151a	12	16	46	1	510	510
adder4	8	13	47	1	178	178
tcon	17	16	48	1	301	301
cm162a	14	21	60	1	512	512
cm163a	16	19	60	1	2157	2157
cm85a	11	20	60	1	541	541
parity	16	15	60	1	300	300
i1	25	21	61	1	424	424
decod	5	18	68	1	486	486
cmb	16	20	73	1	1460	1460
pm1	16	29	85	2	2122	938
x2	10	27	86	2	858	914
cm150a	21	31	92	3	1094	1789
pcle	19	37	98	2	789	1460
cu	14	30	102	2	1172	1503
cc	21	36	106	2	1600	1125
z4ml	7	33	115	2	2011	310
pcler8	27	55	132	3	2203	1374
cordic	23	36	137	3	2302	19353
mux	21	49	160	5	965	3702
i3	132	46	172	3	80685	29689
unreg	36	63	190	3	4527	4705
b9	41	65	198	3	10885	6124
frg1	28	58	198	5	37519	50928
comp	32	57	207	4	640	4036
sct	19	57	213	3	10805	7872
adder16	32	61	215	3	1605	1776
count	35	77	218	4	866	3111
myadder	33	64	228	3	1342	2110
f51m	8	65	233	4	1904	5273
lal	26	82	243	4	5629	6240
c8	28	80	283	4	11563	42998
i2	201	93	330	4	13896	36758
9symml	9	85	340	5	42109	43706
C432	36	99	353	4	14433	35483
i4	192	110	356	6	11368	-
cht	47	119	365	4	27353	28667
apex7	49	125	370	8	33051	-
example2	85	137	426	6	36585	28850
adder32	64	125	439	7	10614	-
ttt2	24	131	443	6	11554	-
alu2	10	210	632	3	87359	51501
adder64	128	253	887	9	20837	-
mult8	16	324	938	17	79151	-

Table 2: Circuit statistics, number of partitions, maximum TCF size obtained using the partition and TCF size of the original method.

The results in table 3 show that using this approach, it is possible to compute exactly the switching activity in conditions that would required a prohibitively large trace if simulation based methods were used. As an example, it was possible to estimate power for test circuits with more that 30 inputs taking into account the full set of correlations introduced by a binary input code.

By comparing the power dissipation figures for the three distributions presented, it becomes clear that taking into account all the correlations in the input stream is crucial to accurately perform power estimation.

[1] This code models a situation that is very common and very hard to address using either simulation based methods or random sampling.

	Binary		High/Low		Uniform	
Circuit	Power	CPU	Power	CPU	Power	CPU
ex2	18.1	0.0	14.4	0.1	16.6	0.1
C17	23.8	0.1	46.2	0.1	62.8	0.1
majority	29.4	0.1	11.2	0.0	52.2	0.1
b1	64.4	0.1	62.5	2.5	150.0	1.7
cm82a	68.3	0.4	53.1	0.7	109.4	0.7
cm42a	90.6	0.7	56.3	0.9	115.7	0.6
cm138a	51.9	0.1	30.6	0.2	149.6	0.2
cm152a	96.8	2.9	45.0	4.4	137.5	1.1
cm151a	51.5	7.9	15.6	8.1	193.4	2.9
adder4	93.5	1.9	-	-	745.1	1033.2
tcon	37.9	5.4	18.8	4.3	197.5	3.7
cm162a	45.1	10.8	20.8	21.0	188.7	6.2
cm163a	31.7	17.6	11.4	10.2	194.2	9.2
cm85a	61.5	4.0	40.2	4.9	182.8	5.7
parity	53.3	8.2	60.0	10.3	187.5	3.9
i1	77.9	20.7	40.0	11.4	206.3	7.4
decod	238.1	1.3	131.2	2.3	316.7	1.5
cmb	60.3	16.3	30.7	17.1	268.3	6.6
pm1	28.1	34.6	2.5	42.4	315.3	19.8
x2	282.4	10.1	185.4	14.9	332.5	7.9
cm150a	65.1	59.7	16.6	35.9	413.8	32.2
pcle	58.9	39.9	29.5	40.1	234.8	27.2
cu	109.7	58.2	57.6	45.1	386.2	17.6
z4ml	236.0	3.4	133.6	5.1	365.4	4.3
pcler8	66.1	812.3	33.4	815.7	293.5	195.1
cordic	80.4	51.5	52.5	27.7	459.8	26.0
mux	195.4	613.8	121.9	606.7	519.7	96.7
unreg	-	-	22.8	6266.2	634.0	193.1
b9	-	-	323.6	4710.6	1781.5	3918.8
frg1	-	-	-	-	691.3	697.9
comp	101.8	124.8	58.3	81.1	721.9	59.1
adder16	109.5	3653.0	14.5	0.1	34.3	0.1
count	-	-	-	-	598.3	415.4
myadder	96.9	2660.9	71.9	6019.8	800.0	4134.4
f51m	477.3	127.3	326.2	251.7	773.2	166.7
9symml	566.6	549.5	344.2	854.5	1372.7	449.8
C432	-	-	-	-	722.7	649.6
cht	-	-	-	-	1189.1	5412.9
alu2	788.1	3186.6	-	-	-	-

Table 3: Power and CPU time statistics for Binary, Uniform and High/Low encodings.

5 Conclusions and future work

In this paper we describe a circuit partitioning technique used to speed up a probabilistic method that performs power estimation considering both spatial and temporal correlations of the primary input signals. By using partitioning techniques, the method does not require the computation of global BDD representations for node functions, thereby extending considerably its range of applicability. This method requires only the computation of the BDDs for nodes in the sub-circuits in terms of the local inputs, while preserving the full set of correlations in the variables that connect the sub-circuits.

As the preliminary results show, the partitioning algorithm proposed effectively extends the size of the circuits that can be handled. In fact, the algorithm was able to compute exactly the switching activity in conditions that would require an extremely large trace if simulation based methods were used.

This preliminary implementation of the partition algorithm does not yet use the optimum ordering for the variables in the BDD required to represent the TCF. Work is in progress to implement the optimum ordering, which we expect will further improve significantly the performance of the algorithm.

Acknowledgments

Useful discussions on various aspects of power optimization with José C. Monteiro are gratefully acknowledged.

This work was supported in part by FCT and the PRAXIS-XXI program.

References

[1] L. Benini, A. Bogliolo, and G. De Micheli. Characterization free behavioral power modeling. In *Proceedings of the Design Automation and Test in Europe*, pages 767–773, 1998.

[2] K. Brace, R. Rudell, and R. Bryant. Efficient implementation of a BDD Package. In *Proc. of the ACM/IEEE Design Automation Conference*, pages 40–45. ACM Press, June 1990.

[3] A. T. Freitas, A. L. Oliveira, J. C. Monteiro, and H. C. Neto. Exact power estimation using word level transition probabilities. In *Proceedings of the Ninth International Workshop on Power and Timing Modelling, Optimization and Simulation*, pages 355–364, Kos Island, Greece, October 1999.

[4] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of Average Switching Activity in Combinational and Sequential Circuits. In *Proceedings of the* 29^{th} *Design Automation Conference*, pages 253–259, June 1992.

[5] D. Marculescu, R. Marculescu, and M. Pedram. Hierarchical sequence compaction for power estimation. In *33rd Design Automation Conference (DAC'97)*, pages 570–575, New York, June 1997. Association for Computing Machinery.

[6] R. Marculescu, D. Marculescu, and M. Pedram. Efficient Power Estimation for Highly Correlated Input Streams. In *Proceedings of the* 32^{nd} *Design Automation Conference*, pages 628–634, June 1995.

[7] A. L. Oliveira, L. Carloni, T. Villa, and A. Sanviovanni Vincentelli. Exact minimization of binary decision diagrams using implicit techniques. *IEEE Transactions on Computers*, 47(11):1282–1296, November 1998.

[8] E. Sentovich, K. Singh, C. Moon, H. Savoj, R. Brayton, and A. Sangiovanni-Vincentelli. Sequential Circuit Design Using Synthesis and Optimization. In *Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors*, pages 328–333, October 1992.

[9] Robert Endre Tarjan. *Data structures and network algorithms*, volume 44 of *CBMS-NSF Reg. Conf. Ser. Appl. Math.* SIAM, 1983.