Non-Stationary Effects in Trace-Driven Power Analysis

Radu Marculescu
Department of ECE
University of Minnesota
Minneapolis, MN 55455
radu@ece.umn.edu

Diana Marculescu
Department of ECE
University of Maryland
College Park, MD 20742
dianam@eng.umd.edu

Massoud Pedram
Department of EE - Systems
University of Southern California
Los Angeles, CA 90089
massoud@usc.edu

1. ABSTRACT

The objective of this paper is to present an analytic technique for power analysis under non-stationary conditions. We use the transitive closure calculation to identify the transient component in the behavior of the target machine and then, based on the fundamental matrix and a symbolic approach (or support from simulation), we find the actual power distribution that corresponds to the transient regime. The present technique complements the current techniques (either for average or peak power estimation) to handle the case when transient effects exist and cannot be ignored.

1.1 Keywords

power consumption, transient regime, Markov chains

2. INTRODUCTION

With the growing need for low-power electronic circuits and systems, power consumption has being given comparable weight to area and performance [1]. This trend is motivated by the remarkable success of personal computing devices and wireless communication systems which require complex functionalities with low-power consumption. For these applications, the *average power* dissipation is a critical design factor. To date, both dynamic [2][3] and static techniques [4]-[6] have been tried to estimate the average power consumption. On the other hand, *peak power* dissipation has become a very important design concern because it determines the thermal and electrical limits of the design, impacts the system cost, and the systems packaging/ cooling requirements. For peak power estimation several approaches have been proposed [7]-[9].

A common characteristic shared by both average and peak power estimation techniques proposed so far is that they consider only *stationary conditions*; that is, it is implicitly assumed that the circuit under consideration has reached its steady-state behavior and the effects of the transient behavior have completely disappeared. This type of analysis corresponds to an infinite observation time; that is, the period of time when an external observer is willing to monitor the behavior of the target circuit is considered to be far larger than the actual time when transient effects affect the behavior of the circuit. There are good reasons for this type of long-term analysis: first, it is in principle desirable to operate a circuit under stationary conditions as much as possible; second, from a theoretical point of view, assuming stationary conditions simplifies the treatment of some probabilistic approaches used for Finite State Machine (FSM) analysis [12][15]. Nevertheless, in real applications, transient regimes often occur and alternate with the normal operation of the system. For instance, let us consider the following piece of pseudocode and its associated graph given in the right hand side (p denotes the probability of branch "*then*" being taken).

```
program test
begin
  s = o;
  read n;
  for i = 1 to n do
     read a[i];
     read b[i];
     s = s + a[i] * b[i];
  if s ≠ 0 then
        aa = 0;
        bb = 0;
        for i = 1 to n do
          aa = aa + a[i] * a[i];
          bb = bb + b[i] * b[i];
        write 'cos = ' s/sqrt(aa * bb);
  else  write 'orthogonal vectors';
end;
```

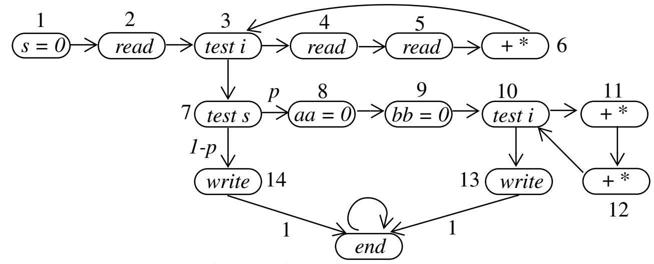
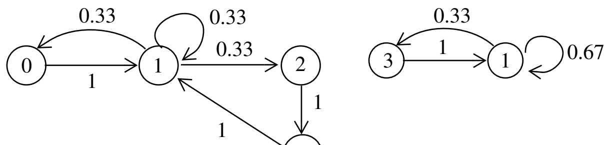


Fig.2.1: A typical example

The graph in Fig.2.1 contains a transient part and an *absorbing state* (labelled with "*end*") from where, once the program enters (this event happens with probability 1), it will never get out. More precisely, the absorbing state means that the control is returned to the system once the execution of this code is terminated. From the absorbing state, the system may proceed to run another code, having again a finite execution time and then, transient regimes appear for finite periods of time and alternate during the operation of the system. Obviously, this transient behavior determines power dissipation in both control and data-path circuitry that perform the computations. In addition, multiple on-chip units and power-down techniques have increased the variability in power consumption which is quantified by the transient power [10]. For example, the transient power effects are exacerbated when switchings between a normal and standby operation modes occur.

Going down to a lower level of abstraction, it is interesting to observe that the transient behavior can emerge either as a result of changes in the external environment where the target FSM is supposed to work or as a consequence of the behavior of the FSM itself. To illustrate the latter case, let us examine the following example.

<u>*Example 2.1*</u>: We consider the *mcnc91* benchmark *donfile* which has two primary inputs and five state lines. Consider now that this circuit is excited with a stationary sequence S_1 which has the State Transition Graph (STG) in Fig.2.2a. In this graph representation, each node is associated to a distinct pattern that occurs in S_1; that is, '0', '1', '2' and '3' stand for the decimal encodings '00', '01', '10' and '11', respectively, of the patterns that occur in S_1.



(a) Sequence S1 (b) Sequence S2

Fig.2.2: Sequence characterization with STG

Each edge represents a valid transition between any two valid patterns and has a nonzero probability associated with it. For instance, the pattern '3' in S_1 is always followed by '1' (thus the edge between nodes '3' and '1' has the probability 1), whereas it is equally likely to have either '0', '2' or '1' after '1'. For a particular encoding scheme of the state lines of the circuit *donfile*, we get the following STG for the primary inputs *and* state lines (Fig.2.3).

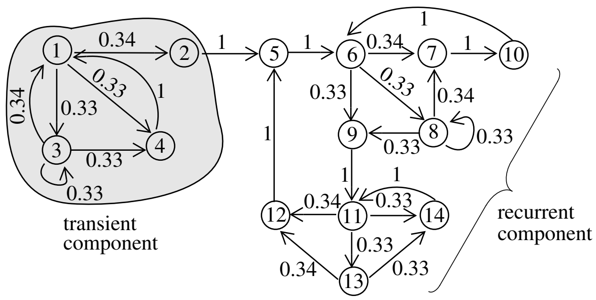


Fig.2.3: Transient component for the circuit in Example 2.1

As we can see, despite the fact that sequence S_1 is completely stationary, out of 14 states that correspond to the overall STG in Fig.2.3, there exists a transient component consisting of 4 transient states. As long as the above Markov chain (MC) stays within the transient component of the STG, we say that the circuit operates under the *transient regime*. Once the Markov chain exits the transient component, we say that the circuit has reached its *stationary* (*permanent*) *regime*. This is because, as we can see in Fig.2.3, once the process exits the highlighted component of the STG, it will never return there.

Generally speaking, the states of a MC can be classified into states that are visited infinitely often (the *recurrent component*) and those that are visited only for a finite number of times (the *transient component*). The long-term (permanent) behavior of a MC involves only the first set of states while the short-time (transient) behavior is related to the other. In addition, we have to consider *absorbing states*; in such a state, the MC will be trapped forever.

Of course, there is power dissipation while the circuits operates within the transient part of the STG. For instance, for this particular circuit and input sequence, the average power consumption per step in the transient regime is about 2726.09 μW, while the average power consumption per step in the permanent regime is about 2964.68 μW. More important, depending on how long the circuit will be staying in the transient regime before entering the permanent regime (which corresponds to the recurrent component of the STG), the contribution of this transient component to the overall power consumption can be quite significant.

In general, the average or the maximum power consumption during the transient regime may be quite different compared to the average power consumption under stationary conditions. For instance, if the circuit *donfile* receives the sequence S_2 instead of S_1 (see Fig.2.2b), then the average and the maximum power consumption per step during the transient regime become 1618.43 μW and 3064.02 μW, while under stationary conditions they become 1184.76 μW and 2197.45 μW, respectively. If only the average (or the maximum) power consumption in the permanent regime is considered as a design factor then, during the transient regime, this threshold will be violated. This shows that although the power consumption under the stationary conditions is what usually drives the power optimization step during the synthesis process, the transient power consumption should also be taken into consideration in the optimization step.

The present paper improves the-state-of the art by providing an original solution for the analysis of power dissipation under non-stationary conditions. The foundation of our approach relies on the theory of Markov chains with absorbing states. As a distinctive feature, we use the transitive closure calculation [16] to identify the transient component in the behavior of the target machine and then, using the fundamental matrix [11] and a symbolic approach or support from simulation, we find the average power consumption (or the actual power distribution) that corresponds to the transient regime.

We point out that the present technique does not represent a substitute for any other approach (either for average or peak power estimation) proposed so far. Instead, the method for transient power analysis proposed here *complements* the current techniques to handle the case when transient effects are present and cannot be ignored.

To conclude, both average and peak power estimation approaches can benefit from this research. The issues brought into attention are new and represent an important step toward a complete solution for power characterization. The paper is organized as follows: Section 3 presents the new method for power estimation under the transient regimes and discusses the basic steps involved in the transient power analysis. In Section 4 we present some experimental results obtained on known benchmarks. Finally, we conclude by summarizing our main contribution.

3. MCs with Absorbing States: Analysis of Transient Power Dissipation

In this section, we first present the model used for power analysis under transient regimes and then the main results needed to carry out our analysis.

3.1 An abstract model for power analysis under transient regimes

The cases that we should consider are sequential/combinational circuits with uncorrelated/correlated inputs. As we will see, a single abstract model will suffice to analyze all these cases.

A. Sequential circuits with temporally uncorrelated inputs

We start by first considering the well-known Huffman representation for FSMs (Fig.3.1). In this representation, the state *and* primary input lines define the random variable of interest (that is, the random variable $(xs)_n$) because this joint variable completely characterizes the behavior of the target FSM. More precisely, the switching activity on primary inputs *and* state lines needs to be correctly estimated in order to assess the power consumption in the combinational part of the FSM in Fig.3.1.

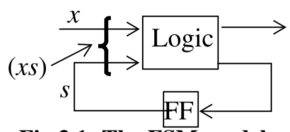


Fig.3.1: The FSM model

If the random variable x_n is temporally uncorrelated, then the model for the random variable $(xs)_n$ is a lag-one MC [15]. This MC may have a transient component which will determine a transient power consumption for the circuit. As it will be shown subsequently, this case is general enough to be applicable to the case of temporally correlated inputs as well.

B. Sequential circuits with temporally correlated inputs

When the input stream applied to the primary inputs of the FSM is temporally correlated, we need to first construct a separate FSM which correctly models the data at the primary inputs and then consider the product machine of the two interacting machines. Indeed, it has been shown that any finite-order Markov chain can be modeled as a stochastic sequential machine (SSM) [13]. Thus, to find the transient power consumption of the target machine, it suffices to analyze the transient behavior of the product machine.

C. Combinational circuits

In this case, since combinational circuits do not have internal storage elements, the transient behavior can only be induced by the input sequence. Again, for the particular case of combinational circuits receiving input correlated data, the input data can be modeled by a stand-alone FSM; then, using this compact representation, we can construct the composite machine and use it for the purpose of transient power analysis. Indeed, analyzing the transient regime for the resulting machine will be equivalent to characterizing the transient regime for the target combinational circuit.

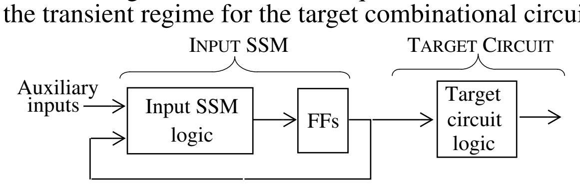


Fig.3.2: The product machine (*input_data_FSM* × *target_circuit*) for combinational/sequential circuits

In summary, the model in Fig.3.1 under uncorrelated input streams is general enough for the purpose of our analysis. In what follows, we consider that the observation time *is finite* and it is comparable with the period of time when the transient behavior of the machine is manifest.

3.2 Mathematical considerations

Homogeneous MCs are described by *single-step transition probabilities* $p_{ij} = p(x_n = j | x_{n-1} = i)$ which are independent of n for all n = 1, 2,... [11]. The matrix P, formed by placing p_{ij} in row i and column j, for all i and j, is called the *transition probability matrix*. We note that P is a *stochastic matrix* because its elements satisfy the following two properties: $0 \le p_{ij} \le 1$ and $\sum_j p_{ij} = 1$.

The MCs of interest for our analysis form the class of MCs with absorbing states. This is because the transient regime is expected to eventually terminate and, after that, the target circuit will operate under stationary conditions. The MC that can be associated to the joint variable $(xs)_n$ for the transient regime contains all the transient states, while the other component, which assumes an infinite observation time, contains only recurrent states.

As shown in Fig.2.3, if the sequence S_1 is fed at the primary inputs of the benchmark *donfile*, the transient component consists of 4 states ({1, 2, 3, 4}) while the recurrent component has 10 states ({5, 6, 7, 8, 9, 10, 11, 12, 13, 14}). Once the target circuit reaches its steady-state behavior, the influence of the 'start-up' transients disappears. However, we focus our attention only on the short-time behavior when the circuit still operates under transient conditions. For the sake of the analysis, we assume that the MC with n states $(s_1, s_2, ..., s_n)$[1] has s_n as the absorbing state (thus s_n corresponds to the stationary regime), and the remaining states are all transient[2]. We should point out that usually the number of recurrent states is larger than one and they form a *terminal strongly connected component* [12]. However, for the sake of simplicity, we consider that all terminal or recurrent states are lumped together in a single absorbing state s_n, as considered above. The transition probability matrix of such a chain may be partitioned as:

$$P = \begin{bmatrix} Q & | & C \\ - & | & - \\ \mathbf{0} & | & 1 \end{bmatrix} \qquad (3.1)$$

where Q is an (n-1) by (n-1) *substochastic* matrix (that is, a matrix with at least one row sum less than 1) describing the transition probabilities only among the transient states, C is a column vector and $\mathbf{0}$ is a row vector of (n-1) zeros. Consequently, the k-step transition probability matrix P^k has the form:

$$P^k = \begin{bmatrix} Q^k & | & C' \\ - & | & - \\ \mathbf{0} & | & 1 \end{bmatrix} \qquad (3.2)$$

where C' is a column vector whose elements will be of no further use and hence do not need to be computed.

The (i, j) entry in the matrix Q^k denotes the probability of arriving in transient state s_j after exactly k steps when starting from transient state s_i. It can be shown that $\sum_{k=0}^{t} Q^k$ converges as t approaches infinity [11]. This implies that the inverse matrix $(I - Q)^{-1}$, called the *fundamental matrix M*, exists and is given by:

$$M = (I - Q)^{-1} = I + Q + Q^2 + \ldots = \sum_{k=0}^{\infty} Q^k \qquad (3.3)$$

where I is the identity matrix of order (n-1).

---

[1] Every state denotes in fact a pair (xs); that is a (*primary_input, present_state*) tuple.

[2] For the absorbing state s_n there is no outgoing edge. To have the equation $\sum_j p_{ij}$ satisfied for all i, we have to add a self-loop (with probability 1) around the state s_n.

Example 3.1: For the same circuit in Example 2.1, we can collapse all the recurrent states in a single absorbing state s_n as shown in Fig.3.3.

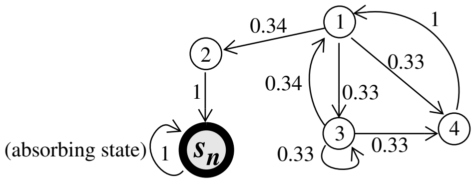


Fig.3.3: The new STG after collapsing all recurrent states

Using the notation in eq. (3.1) we have that

$$Q = \begin{bmatrix} 0 & 0.34 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0 \\ 0.34 & 0 & 0.33 & 0.33 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \text{ and } M = \begin{bmatrix} 2.94 & 1 & 1.45 & 1.45 \\ 0 & 1 & 0 & 0 \\ 2.94 & 1 & 2.94 & 1.94 \\ 2.94 & 1 & 1.45 & 2.45 \end{bmatrix}.$$

The fundamental matrix M represents a rich source of information about the MC. Let n_{ij} be the average number of times the circuit visits the transient state s_j before entering the absorbing state, given that it started in transient state s_i. Based on the information provided by the fundamental matrix, the following result holds:

Theorem 3.1. [11] The average number of visits to state s_j before entering the absorbing state s_n, given that the circuit was initialized in state s_i, is given by $n_{ij} = m_{ij}$, where m_{ij} represents the (i,j) entry in the fundamental matrix M. ■

Example 3.2: Using the data in Example 3.1 and assuming that the circuit starts in state 1, before entering the absorbing state s_n, we expect an average number of 2.94 visits to state 1, 1 visit to state 2, 1.45 visits to state 3 and 1.45 visits to state 4.

Having available n_{ij} and matrix Q, one can compute the average number of times when the *transition* from state s_j to state s_k is taking place before entering the (final) absorbing state. Indeed the following result holds:

Theorem 3.2. The average number of times the circuit visits the *transition* from state s_j to state s_k before entering the absorbing state (given it was started in state s_i) is given by:

$$\eta_{jk} = m_{ij} \cdot p_{jk} \tag{3.4}$$

where p_{jk} is the (j, k) entry in the stochastic matrix P characterizing the behavior of the $(xs)_n$ Markov chain and m_{ij} is the (i, j) entry in the fundamental matrix M. ■

Once we calculate η_{ij}, using the average weight (power) per vector pair (denoted by α_{ij}), one can estimate the average power dissipation (per step) for the transient regime. Indeed, the following result holds for any weighting function associated with the transitions of the MC.

Theorem 3.3. The average power consumption for the transient component of a sequential circuit is given by:

$$P_{transient} = \left(\sum_{\substack{i,j \\ i \text{ transient}}} \eta_{ij} \cdot \alpha_{ij} \right) \Big/ \left(\sum_{\substack{i,j \\ i \text{ transient}}} \eta_{ij} \right) \tag{3.5}$$

We note that $P_{transient}$ is dependent on the actual input data (via α_{ij}). This dependence implies that detailed information about the target circuit should be available at the time when the transient analysis is performed. In the following section, we will describe a practical procedure to evaluate $P_{transient}$.

3.3 Practical considerations

The basic steps involved in transient power analysis are state classification and evaluation of the weight function α_{ij} associated to each transition. While the first step needs only a high-level description of the behavior of the FSM under consideration (and can be performed completely symbolically), finding the power consumption associated to each transition needs detailed information about the target circuit. Let us consider the two steps in more detail.

A. State classification for Markov chains

To perform transient power analysis for different circuits, an initial and essential step involves a structural analysis of the Markov chain. To do this, a *reachability analysis* of the underlying FSM is first done to reduce the set of states to only those that actually will appear in the behavior of the FSM. The reachability analysis step is done completely symbolically using BDDs, either for unconstrained inputs [14] or for a very specific input stream [15].

Next, only the topological information of the Markov chain is needed to classify states into transient or recurrent. To this end, two types of approaches have already been proposed. The first, proposed in [12] uses the recursive paradigm in [16] to compute the transitive closure of the transition relation. The procedure is completely symbolic and uses BDDs as a compact and canonical representation of the transition relation. A more efficient technique has been recently presented in [17]. Both approaches can be used for our purpose of finding the transient component of the underlying MC.

When all states have been classified, the set of recurrent states can be further collapsed into a single *absorbing state* to reduce the state space which has to be analyzed. This reduced MC has to be further characterized in terms of power consumption associated to every outgoing transition from a transient state.

B. Finding the power consumption per transition

Using theorems 3.1 and 3.2, the reduced MC obtained by means of state classification can be characterized in terms of number of visits to each state (or each transition) before the absorbing state is reached. To find the average power consumption that corresponds to the transient regime, every possible transition between the transient states has to be characterized in terms of its power consumption.

As other researchers have observed [17], the size of the transient component is usually very small compared to the entire reachable state space (typically from a few tens to a few hundred vector pairs). Consequently, to characterize every transition in terms of power dissipation, one can use a simulation-based approach. More precisely, for each transition, the circuit is simulated for the corresponding input pair and the obtained power value is used to compute the average power consumption in the transient regime.

Alternatively, the FSM can be symbolically analyzed for power consumption using an ADD-based technique as in [7]. Or, if the circuit under consideration is pre-characterized by a *cycle-based power macro-model* [18], for each input vector pair, one can obtain in constant time an approximate value for the power consumption associated to each transition.

Our framework is open to any of the above alternatives. The most accurate approach is, of course, to find the power values directly from simulation, but this may not be desirable if the transient component is large.

4. EXPERIMENTAL RESULTS

The overall strategy is depicted in Fig.4.1; it has two main parts: a *preprocessing stage* (which provides the transient component) and the *transient power analyzer* itself.

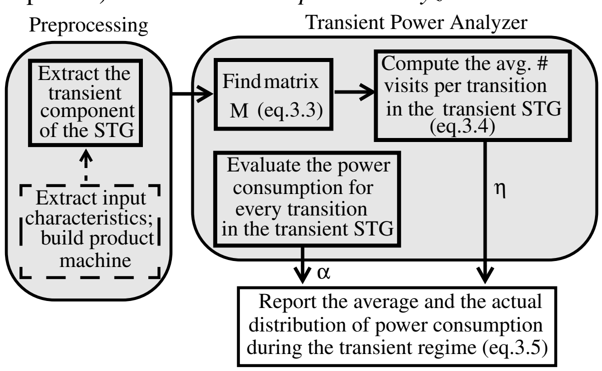


Fig.4.1. The experimental setup

Regarding the preprocessing module, we note that if the inputs of the target circuit are temporally correlated and exhibit a transient regime on their own, an optional first step for building the FSM characterizing the input stream is needed. After that, the composite product machine is built and analyzed for state classification purposes.

Next, the fundamental matrix M for the set of transient states is built and, based on it, the average number of visits to every state and every transition is extracted. Finally, with some support from a power analyzer, the average and maximum power values under the transient regime are reported. Alternatively, the actual distribution of power for the entire transient regime can be also reported.

For the purpose of our experimental analysis, the input characteristics for the circuits under consideration are summarized below:

- For some of the sequential circuits (*minmax2, minmax5*), the inputs are considered temporally uncorrelated and then randomly generated, with the exception of some control bits which were set either to 0 or 1 (this is basically the same strategy as in [12]). The signal probabilities of the randomly generated bits vary uniformly between 0 and 1. For the rest of sequential circuits, the inputs are considered temporally correlated and generated by a finite-order MC with transition probabilities between 0 and 1 [3].

- For each of the combinational circuits, a stand-alone FSM exhibiting transient behavior has been used to model the input characteristics of the circuit under consideration. The inputs to the stand-alone FSM are considered temporally uncorrelated and randomly generated with signal probabilities between 0 and 1.

To test our approach, we conducted two types of experiments:

1) The first set of experiments offers an extensive analysis of different sequential and combinational circuits (from *mcnc91* and ISCAS85 suites) for transient power analysis. For the case of combinational circuits, the inputs exhibiting transient behavior are generated by an input FSM which is analyzed for transient and stationary regimes. For the case of sequential circuits, the transient behavior is induced by the structure of the FSM itself (e.g., circuit *minmax2*) or by the behavior of the input stream (e.g., circuit *donfile*). For both sequential and combinational circuits, we report the average power consumption for the transient component, as well as under stationary conditions. In both cases, for accuracy reasons, we used a real-delay gate-level simulation-based power analyzer developed under SIS.

As it can be seen in Table 4.1, the transient component contains only a small number of states in all cases (column 'Trans. states'), and thus performing a simulation-based power analysis for the transient regime is not computationally expensive (columns denoted as 'Transient regime'). For comparison, we also report the power values obtained under stationary conditions (columns 'Permanent regime'). The CPU time for the transient power analyzer in Fig.4.1 has been below 10 sec. in all cases on an Ultra SPARC 2 with 128 Mbytes of memory.

				Transient regime		Permanent regime	
Circuit	Inp./FFs	Gates	Trans. states	Avg. power	Max. power	Avg. power	Max. power
bbara	4/4	59	4	535	829	747	1437
bbsse	7/4	92	5	130	316	1775	3599
bbtas	2/3	23	6	512	725	123	507
donfile	2/5	121	4	1618	3064	1184	2197
ex4	6/4	65	1	2071	2071	1371	2523
s400	3/21	126	1	185	185	61	123
s526	3/21	186	4	1560	2937	1292	2642
s953	16/29	165	17	597	2097	1110	2134
minmax2	5/6	78	25	828	1805	550	2340
minmax5	8/15	225	25	1417	3254	796	2381

Table 4.1: **Power analysis for sequential circuits (μW@5V, 20 MHz)**

As we can see in Table 4.1, in most cases there is a big difference between the average power consumption during the transient regime and under stationary conditions, respectively. For example, this difference can be up to 50% in the case of *minmax2* circuit, with a larger power consumption under transient conditions.On the other hand, the transient power consumption may be much smaller than the one under stationary conditions (e.g., circuit *bbsse*). In such cases, if the transient and stationary regimes are comparable in terms of their duration, the average power value for the stationary regime is no longer accurate. This observation is in perfect agreement with what other researchers have reported for power analysis under deterministic input streams [19].

There are also quite a few examples where the maximum power consumption can be up to 50% larger under transient regime (e.g. circuit *s400*). This analysis shows that average or maximum power consumption under stationarity conditions is not a reliable indicator for the power dissipation of the circuit when transient regimes appear.

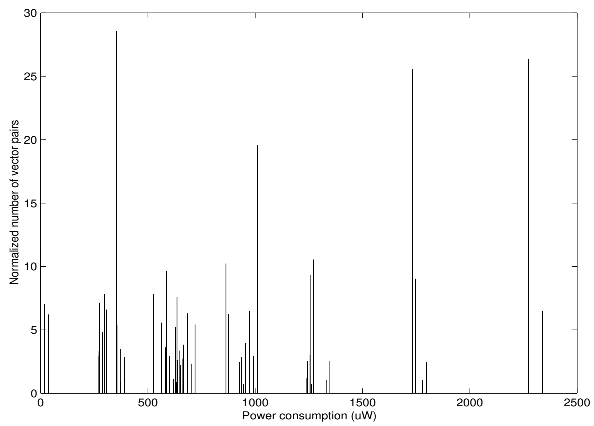
For combinational circuits, the same observations apply. As it can be seen in Table 4.2, the average power consumption under transient conditions can be up to 52% larger than the one under stationary conditions (e.g., circuit *dalu*), while the transient maximum power values are up to 116% larger than the stationary ones (e.g. circuit *too_large*).

2) The second set of experiments involves a comparison of the distribution of power consumption in the transient and stationary regimes. A typical example is circuit *minmax2* which has 25 transient states from the total of 41 reachable combinations (*xs*).

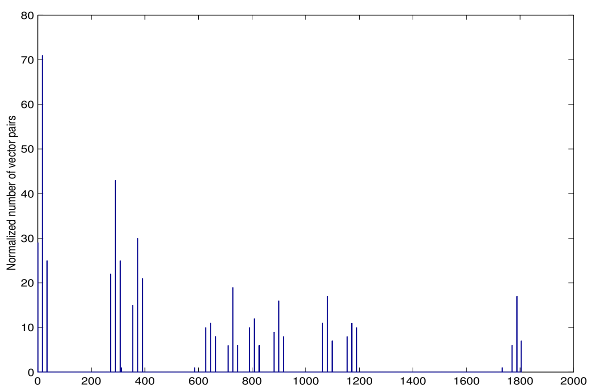
			Transient regime		Permanent regime	
Circuit	Inp.	Gates	Avg. power	Max. power	Avg. power	Max. power
apex7	49	105	1204	4767	926	4476
dalu	75	79	826	2333	546	1797
too_large	38	88	181	725	123	335
C17	5	6	43	112	53	127
C432	36	174	4053	11614	3229	7881
C499	41	352	2653	5832	2274	5407
C1355	41	292	3199	12517	2663	9888
C1908	33	384	1302	4803	1352	4803
C3540	50	1021	6759	17840	5441	12429
C6288	32	2972	82425	211280	83803	191570

Table 4.2: **Power analysis for combinational circuits (μW@5V, 20 MHz)**

Analyzing power consumption under both regimes, we found two different behaviors (Fig.4.2). We note that not only the average power consumption is very different for the two regimes (828.54 μW for the transient and 550.19 μW for the stationary regime), but also the distribution of power values is very different.



a) Transient regime



b) Stationary regime

Fig.4.2. Power distribution for transient and stationary regimes (circuit *minmax2*)

We also note that the maximum power value is only around 1805 μW under stationarity conditions, but may be up to 2340 μW under transient conditions (almost 30% difference). Thus, neglecting the transient power consumption or analyzing the circuit only for stationary conditions may give erroneous results in terms of both average and peak power consumptions.

5. Conclusion

We addressed the problem of power dissipation in sequential and combinational circuits under transient regimes. The foundation of our approach relies on the theory of MCs with absorbing states. As distinctive feature, we use the transitive closure calculation to identify the transient component in the behavior of the target machine and then, using a symbolic approach or support from simulation, we find the average/maximum power consumption or the actual power distribution that corresponds to the transient regime. The issues brought into attention represent an important step towards an integrated framework which considers both stationary and non-stationary conditions for power analysis.

References

[1] J. Rabaey, and M. Pedram, Eds., ‘Low Power Design Methodologies’, Kluwer Academic Publishers, 1996.

[2] C. X. Huang, B. Zhang, A.-C. Deng, and B. Swirski, ‘The Design and Implementation of PowerMill’, in *Proc. Intl. Workshop on Low Power Design,* pp. 105-110, Apr. 1995.

[3] F. N. Najm, ‘A Monte Carlo Approach for Power Estimation’, in *IEEE Trans. on VLSI Systems*, 1(1):63-71, Mar.1993.

[4] F. N. Najm, ‘Transition Density: A New Measure of Activity in Digital Circuits’, in *IEEE Trans. on CAD*, 12(2):310-323, Feb.1993.

[5] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin, ‘Power Estimation Methods for Sequential Logic Circuits’, in *IEEE Trans. on VLSI Systems*, 3(3):404-416, Sept. 1995.

[6] R. Marculescu, D. Marculescu, and M. Pedram, ‘Probabilistic Modeling of Dependencies During Switching Activity Analysis’, in *IEEE/ACM Trans. on CAD*, 17(2):73-83, Feb. 1998.

[7] S. Manne, A. Pardo, R. I. Bahar, G. D. Hachtel, F. Somenzi, E. Macii, and M. Poncino, ‘Computing the Maximum Power Cycles of a Sequential Circuit’, in *ACM/IEEE Design Automation Conf.*, pp. 23-28, June 1995.

[8] Q. Qiu, Q. Wu, and M. Pedram, ‘Maximum Power Estimation Using the Limiting Distributions of Extreme Order Statistics’, in *ACM/IEEE Design Automation Conf.*, pp. 684-689, June 1998.

[9] C.-Y. Wang and K. Roy, ‘Maximum Power Estimation for CMOS Circuits Using Deterministic and Statistical Approaches’, in *IEEE Trans. on VLSI Systems*, 6(1):134-140, Mar. 1998.

[10] V. Tiwari, D. Singh, S. Rajagopal, G. Mehta, R. Patel, and F. Baez, ‘Reducing Power in High-Performance Microprocessors’, in *ACM/IEEE Design Automation Conf.*, pp. 732-737, June 1998.

[11] K. Trivedi, ‘Probability and Statistics with Reliability, Queuing, and Computer Science Applications’, Prentice-Hall, Inc., Englewood Cliffs, 1982.

[12] G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, ‘Markovian Analysis of Large Finite State Machines’, in *IEEE Trans. on CAD*, 15(12):1479-1493, Dec. 1996.

[13] D. Marculescu, R. Marculescu, and M. Pedram, ‘Stochastic Sequential Machine Synthesis Targeting Constrained Sequence Generation’, in *Proc. ACM/IEEE Design Automation Conf.*, pp. 696-701, June 1996.

[14] O. Coudert, C. Berthet, and J. C. Madre, ‘Verification of Sequential Machines Based on Symbolic Execution’, in J. Sifakis, Ed., *Automatic Verification Methodologies for Finite State Systems*, Springer-Verlag, June 1989.

[15] D. Marculescu, R. Marculescu, and M. Pedram, ‘Trace-Driven Steady-State Probability Estimation in FSMs with Application to Power Estimation’, in *Proc. Design, Automation and Test in Europe Conf.,* pp. 774-779, France, Feb. 1998.

[16] Y. Matsunaga, P. C. McGeer, R. K. Brayton, ‘On Computing the Transitive Closure of a State Transition Relation’, in *Proc. ACM/IEEE Design Automation Conf.,* pp. 260-265, June 1993.

[17] Aiguo Xie, Peter A. Beerel, ‘Efficient State Classification of Finite State Markov Chains’, in *IEEE Trans. on CAD*, 17(12):1334-1339, Dec. 1998.

[18] Q. Qiu, Q. Wu, M. Pedram and C-S. Ding, ‘Cycle-accurate macro-models for RT-level power analysis’, in Proc. *Intl. Symposium on Low Power Electronics and Design*, pp. 125-130, Aug. 1997.

[19] L. Benini, G. De Micheli, E. Macii, M. Poncino, and R. Scarsi, ‘Fast Power Estimation for Deterministic Input Streams’, in *Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design,* pp. 494-501, Nov. 1997