Cycle-Accurate Test Power Modeling and its Application to SoC Test Scheduling

Soheil Samii[1], Erik Larsson[1], Krishnendu Chakrabarty[2], Zebo Peng[1]

[1] Embedded Systems Laboratory
Linköping University, Sweden
{sohsa,erila,zpe}@ida.liu.se

[2] Dept. of Electrical & Computer Eng.
Duke University, USA
krish@ee.duke.edu

Abstract*

Concurrent testing of the cores in a modular core-based System-on-Chip reduces the test application time but increases the test power consumption. Power models and scheduling algorithms have been proposed to schedule the tests as concurrently as possible while respecting the power budget. The commonly used global peak power model, with a single value capturing the power dissipated by a core when tested, is pessimistic but simple for a scheduling algorithm to handle. In this paper, we propose a cycle-accurate power model with a power value per clock cycle and a corresponding scheduling algorithm. The model takes into account the switching activity in the scan chains caused by both the test stimuli and the test responses during scan-in, launch-and-capture, and scan-out. Further, we allow a unique power model per wrapper chain configuration as the activity in a core will be different depending on the number of wrapper chains at a core. Extensive experiments on ITC'02 benchmarks and an industrial design show that the testing time can be substantially reduced (on average 16.5% reduction) by using the proposed cycle-accurate test power model.

1 Introduction

Long test application times for SoCs (Systems-on-Chip) is acknowledged as a major problem in the industry. An efficient way to reduce the test application time for core-based SoCs is to schedule the tests concurrently so that several cores are tested at the same time. However, concurrent testing leads to an increased switching activity in the chip, hence higher power consumption. Several factors during the manufacturing process impose a power limit which must not be exceeded [1], otherwise the device could be damaged due to overheating during test. The factors limiting the test power are, among others, the availability and choices of materials, different package (or wafer) cooling solutions, and also the cooling costs. Thus, in test scheduling there is a power constraint that must be carefully considered and not violated. Typically, there exists a trade-off between the power consumption and the testing time. Compared to concurrent testing, scheduling the module tests in a sequential fashion will result in longer testing time but lower power consumption.

The most commonly used power model is the global peak power model where a single power value, the highest for the power curve, is used to represent the power curve of each test [1]. Such a model guarantees that the total power budget is not exceeded; however, the model is rather pessimistic. The advantage with the model is that the SoC test scheduling algorithm only has to keep track of a single value per test.

In this paper we propose a cycle-accurate power model where there is a power value per clock cycle. The power model is based on information from both the test stimuli and the test responses for each testable unit. We consider, at each clock cycle, the scan chain switching activity produced by both the test stimuli and the test responses. Hence, we consider the switching activity during scan-in/scan-out, as well as during launch-and-capture. Further, as the same test stimuli and test responses for a given core may still consume different power depending on the number of wrapper chains, we have a power profile for each wrapper chain configuration; hence there are several power profiles for each core.

To demonstrate the usefulness of the proposed cycle-accurate power model, and to compare it with the single-value model, we also propose a test scheduling algorithm that makes use of the proposed power model. We have implemented the proposed test scheduling algorithm. The experimental results on several ITC'02 benchmarks [2], [3], and an industrial design show that by making use of the proposed power

*The research is partially supported by the Swedish Foundation for Strategic Research (STRINGENT project).

modeling technique the test application time can be substantially reduced compared to using the single-value power model.

The rest of the paper is organized as follows. Section 2 gives an overview of related prior work and Section 3 describes the proposed test power modeling technique. In Section 4 the proposed test scheduling heuristic is discussed, while the experimental results are given in Section 5. Finally, the paper is concluded in Section 6.

2 Related Work

Chou *et al.* [1] approximate the test power consumption for each block (core) to a single fixed value, the peak power consumption. Rosinger *et al.* [4] refer to this as the "global peak power (approximation) model". Figure 1 shows the actual power consumption and the modeled power consumption based on a single value for a design example. The false power is the mismatch between the actual power consumption and the modeled power consumption. The single-value power model is rather pessimistic, but it guarantees that the maximum power consumption will not be violated, and is very simple to be handled by a test scheduling algorithm.

An alternative is to use the average value for estimating the test power consumption. It is suitable for many applications, however not for test scheduling, since it may lead to that the power limit will be exceeded in certain time intervals [1].

Rosinger *et al.* [4] propose a double-value test power model (one value representing a constant low power consumption, and the other value representing a constant high power consumption for a test), and further propose a step towards the integration of this power model into an existing test scheduling technique introduced by Chou *et al.* [1]. On top of this double-value test power model they use test pattern reordering to reshape the power profiles. Further, they also consider test sequence expansion for lowering peak power values, that is, they insert a new test pattern between two test patterns that generate high peak power values. This new test pattern will not increase the test coverage, but it is merely used for the purpose of lowering the power consumption in different time intervals. Since this results in a longer testing time, the test sequence expansion technique is clearly a trade-off between power consumption and testing time. These techniques are, however, not extensively evaluated in applications, such as test scheduling algorithms.

In some applications, low power consumption is a design objective, rather than a constraint. Dabholkar *et al.* [5] and Ghosh *et al.* [6] studied scan cell

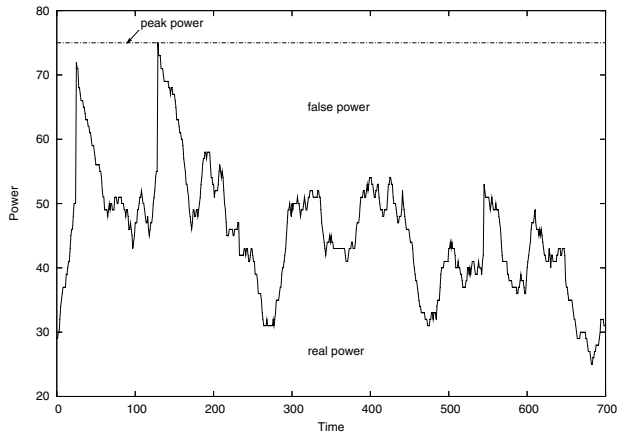

Figure 1: Global peak power model

reordering (reordering of the flip-flops in a core's scan chains) targetting minimization of the test power consumption. For power estimation, their techniques are merely based on the test inputs to the circuit, that is, they consider only the test stimuli for the different cores. Moreover, they consider the switching activity during test that is due to the transitions in the scan chains. In other approaches to test power minimization [7], [8], the power consumption is modeled based on the Hamming distances between the test stimuli.

Huang *et al.* [9] and Pouget *et al.* [10] also studied power-constrained test scheduling. The proposed algorithms assume that each core is assigned a fixed value for its test power consumption, hence the global peak power model is used. The global peak power model has also been used in several other types of test scheduling algorithms [1], [11], [12].

We should also mention that there are other ways of dealing with chip overheating during test. Recently, Rosinger *et al.* [13] discussed local heating, due to the non-uniform distribution of power across a chip. They proposed a thermal model, guiding the test scheduler to produce test schedules which will not lead to overheating of the circuit under test. Another thermal-aware test scheduler for core-based systems has recently been proposed by Liu *et al.* [14].

3 Cycle-Accurate Power Modeling

The power consumption is the sum of a static part and a dynamic part. For most current CMOS technologies, the static part is constant and dominated by the dynamic part. Usually, the dynamic part is proportional to the switching activity α (the number of zero-to-one and one-to-zero transitions) in the circuit [15]; hence we will concentrate on how we can determine α on the basis of the given test stimuli and

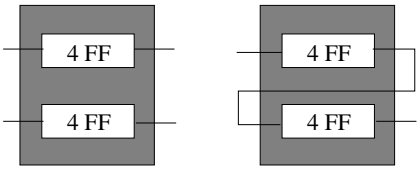

Figure 2: A core with two scan chains connected as (a) two wrapper chains, and (b) one wrapper chain

the given expected test responses for a core.

Sankaralingam *et al.* [16] empirically showed that the number of transitions in the logic of a core when applying a test, α_{logic}, is approximately linear to the transitions in the core's flip-flops α_{ff}. Hence, $\alpha = \alpha_{\text{ff}} + \alpha_{\text{logic}} = \alpha_{\text{ff}} + k\alpha_{\text{ff}} + l$, where k and l are constants. This conclusion has recently been strengthened by extensive power simulations [17]. We will, therefore, in the rest of this paper focus on the transitions α_{ff} at the cores' wrapper input/output cells and scan chains.

In the rest of this section we will first introduce a transition count model considering one single scan chain. Then we will extend the model to handle testing of a core, consisting of a set of wrapper input/output cells (flip-flops) and, for sequential cores, a set of scan chains.

Before that, however, we illustrate how the power profiles vary depending on the wrapper chain configuration. We make use of an example core with two scan chains (Figure 2). The scan chains may be configured into two wrapper chains as in Figure 2(a) or as a single wrapper chain as in Figure 2(b). We consider that the core is tested with five test patterns. We show in Figure 3 the power profile for each wrapper chain configuration. Note that the two power profile are different both in the time domain and in the power domain.

3.1 Transitions in a Scan Chain

Testing a core, equipped with scan chains, means shifting in a test stimulus to the core's scan chains, launching the test (one capture cycle), whereafter the test response is shifted out while shifting in the next test stimulus. Transitions occur due to the shifting in of the test stimuli, launch-and-capture, and the shifting out of the test responses.

Consider the scan chain in Figure 4 with initial values 0 in all flip-flops, that is, $\boldsymbol{x} = (0, 0, 0)$. We show how the values in the flip-flops change while shifting in the bit sequence $\boldsymbol{y} = (0, 0, 1)$, starting with $y_3 = 1$. The sequence $\boldsymbol{y}$ can be viewed as the current test stimulus for the scan chain. Similarly $\boldsymbol{x}$ is the produced test response from the previous captured test stimulus.

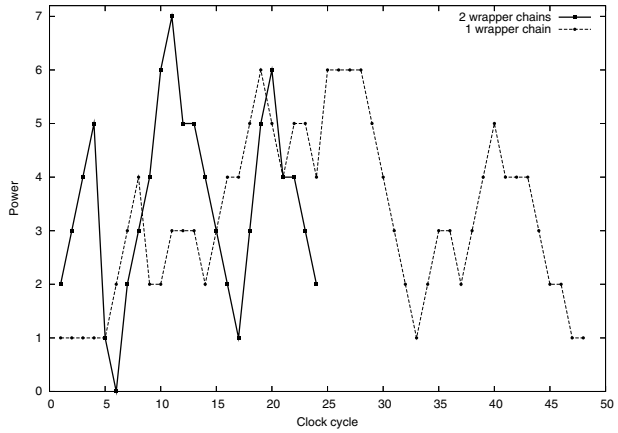

Figure 3: The power profiles (scan in and scan out) for the two different wrapper chain configurations illustrated in Figure 2 when tested with 5 test patterns

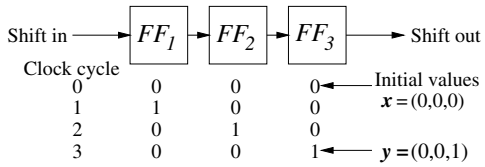

Figure 4: Example of scan chain transitions

For example, at the second clock cycle (from one to two) there are two transitions (at FF_1 and FF_2). In general, shifting in m bits in a scan chain of length n takes m clock cycles. The transitions during shifting can be modeled according to the following definition.

Definition 1: Consider shifting in a bit sequence $\boldsymbol{y} = (y_1, \ldots, y_m)$, starting with y_m, into a scan chain with initial value $\boldsymbol{x} = (x_1, \ldots, x_n)$, that is, flip-flop j has value x_j. For the transitions that occur during shifting we associate an $m \times n$ *transition matrix*

$$T(\boldsymbol{x}, \boldsymbol{y}) = \begin{pmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & & \vdots \\ t_{m1} & \cdots & t_{mn} \end{pmatrix},$$

where $t_{ij} = 1$ if during clock cycle i there has been a transition in flip-flop j, otherwise $t_{ij} = 0$. ∎

For the transition matrix in Definition 1 we associate a *transition function* tr, representing the scan chain transitions α_{ff} at each clock cycle, where

$$tr(i) = \sum_{j=1}^{n} t_{ij} \qquad (1)$$

Cycle	FF_1	FF_2	$\cdots$	FF_j	$\cdots$	FF_{n-1}	FF_n
0	x_1	x_2	$\cdots$	x_j	$\cdots$	x_{n-1}	x_n
1	y_n	x_1		x_{j-1}		x_{n-2}	x_{n-1}
2	y_{n-1}	y_n		$\vdots$		$\vdots$	x_{n-2}
$\vdots$	$\vdots$	y_{n-1}		x_1		$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$		y_n		$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		x_1	$\vdots$
$n-1$	y_2	$\vdots$		$\vdots$		y_n	x_1
n	y_1	y_2	$\cdots$	y_j	$\cdots$	y_{n-1}	y_n

Table 1: Values in a scan chain at shifting

(i)	t_{11}	$=$	$x_1 \oplus y_n$	
(ii)	t_{1j}	$=$	$x_{j-1} \oplus x_j$	$(1 < j \leqslant n)$
(iii)	t_{i1}	$=$	$y_{n-i+2} \oplus y_{n-i+1}$	$(1 < i \leqslant n)$
(iv)	t_{ij}	$=$	$t_{(i-1)(j-1)}$	$(1 < i, j \leqslant n)$

Figure 5: Equations for a transition matrix

is the number of transitions during clock cycle i (the sum of row i in the transition matrix).

The elements in the transition matrix can be determined by knowing the initial values and the bits to be shifted in. We will first consider the case where the number of bits to be shifted in is the same as the number of flip-flops in the scan chain. Thus we will consider a scan chain with n flip-flops, where x_j is the initial value in flip-flop j. The bit sequence to be shifted in (the test stimulus) is $(y_1, \ldots, y_n)$, starting from y_n. As in Figure 4 we note the values in the flip-flops after each clock cycle during shifting in Table 1. Note that the first row (cycle 0) corresponds to the values in the scan chain before shifting.

From Table 1 we see that a transition which occurs in a flip-flop implies that the next flip-flop in the scan chain will have a transition during the next clock cycle. Thus, a transition matrix is determined by its first row and first column. We have collected the equations determining the transition matrix in Figure 5. The first three equations determine the first row and first column, while the last equation means that the elements in a diagonal are identical. Observe that the symbol $\oplus$ denotes the XOR operator on $\{0, 1\}$.

Thus far we have considered the length of the shift-in sequence to be equal to the number of flip-flops in the scan chain. In the next subsection we will consider testing a core, with a configuration consisting of a set of wrapper chains, that is, scan chains and wrapper input/output cells connected together. In this general case, the length of the shift-in sequence is not always equal to the lengths of each of the wrapper chains. Therefore, we will now study the general case where the length of the shift-in sequence, m, is not equal to the number of flip-flops, n, in the scan chain. As before, let $\boldsymbol{x} = (x_1, \ldots, x_n)$ represent the value of the scan chain before shifting, that is, x_i is the value of flip-flop i. Further, let $\boldsymbol{y} = (y_1, \ldots, y_m)$ be the bits to be shifted in. We have already studied the case $m = n$, and thus it remains to study the cases (I) $m > n$ and (II) $m < n$. We will in the following show that the transition matrix for these cases can be calculated by using the methods presented for the case $m = n$.

(I) In this case we first consider the first n bits that will be shifted in, that is, the bits $\boldsymbol{y}_1 = (y_{m-n+1}, y_{m-n+2}, \ldots, y_m)$, and then the rest of the bits $\boldsymbol{y}_2 = (y_1, \ldots, y_{m-n})$. Now we can write the whole $m \times n$ transition matrix as a composition of two matrices, that is,

$$T(\boldsymbol{x}, \boldsymbol{y}) = \begin{pmatrix} T(\boldsymbol{x}, \boldsymbol{y}_1) \\ T(\boldsymbol{y}_1, \boldsymbol{y}_2) \end{pmatrix}. \tag{2}$$

(II) In this case we will first consider the transition matrix for the first m flip-flops of the scan chain, whereafter we consider the rest of the $n - m$ flip-flops. So let $\boldsymbol{x}_1 = (x_1, \ldots, x_m)$ and $\boldsymbol{x}_2 = (x_{m+1}, \ldots, x_n)$. Then the whole transition matrix is, again as a composition of two matrices,

$$T(\boldsymbol{x}, \boldsymbol{y}) = \begin{pmatrix} T(\boldsymbol{x}_1, \boldsymbol{y}) & T(\boldsymbol{x}_2, \boldsymbol{x}_1) \end{pmatrix}. \tag{3}$$

With the results obtained thus far we can see that the transitions in the upper diagonals of a transition matrix only depend on the initial value of the scan chain, that is, the previous test response. Similarly the lower diagonals only depend on the values that are shifted in, that is, the test stimulus. Finally, the main diagonal depends on the initial value of the first flip-flop and the first bit that is shifted in. Thus, the transition function can be divided into three parts, $tr = tr_{\boldsymbol{x}} + tr_{\boldsymbol{x},\boldsymbol{y}} + tr_{\boldsymbol{y}}$, where for clock cycle i we have

$$tr_{\boldsymbol{x}}(i) = \sum_{i<j\leqslant n} t_{ij}, \tag{4}$$

$$tr_{\boldsymbol{x},\boldsymbol{y}}(i) = t_{ii} \in \{0, 1\}, \text{ and} \tag{5}$$

$$tr_{\boldsymbol{y}}(i) = \sum_{1\leqslant j<i} t_{ij}. \tag{6}$$

Figure 6 shows an example of how the different parts of the transition function vary with time. For this example we have $tr_{\boldsymbol{x},\boldsymbol{y}}(i) = 0$.

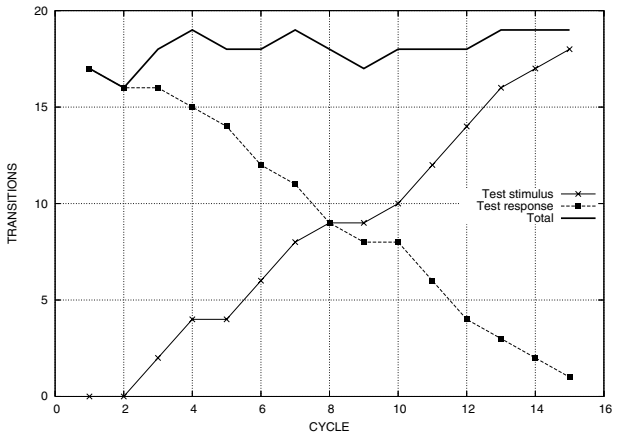

Figure 6: The two parts constituting the transition function

3.2 Transitions in Wrapper Chains

We will now consider testing a core and thus assume that the scanned elements (wrapper input/output cells and scan chains) are connected as w wrapper chains. The scan-in and scan-out times of the core are $si = \max\{si_1, \ldots, si_w\}$ and $so = \max\{so_1, \ldots, so_w\}$ respectively, where si_k and so_k are the scan-in and scan-out of wrapper chain k respectively. Further, let l_k be the length of wrapper chain k. Depending on which test pattern that is applied, the shift-in time is different. This time is (in clock cycles)

$$L = \begin{cases} si & \text{first stimulus is shifted in,} \\ so & \text{last response is shifted out,} \\ \max(si, so) & \text{otherwise.} \end{cases} \quad (7)$$

Thus, for some wrapper chains we need to shift in some idle bits before shifting in the actual test stimulus. The testing time (in clock cycles) for a core with p test patterns is calculated as [18]

$$\tau = (\max(si, so) + 1) \cdot p + \min(si, so). \quad (8)$$

We will denote the initial value of wrapper chain k, that is, the test response of the previous captured test stimulus, by $\boldsymbol{r}^{(k)} = (r_1^{(k)}, \ldots, r_{l_k}^{(k)})$. Similarly the test stimulus bits are

$$\boldsymbol{s}^{(k)} = (s_1^{(k)}, \ldots, s_{si_k}^{(k)}, \underbrace{X, X, \ldots, X}_{L - si_k \text{ idle bits}}),$$

where the idle bits (the X's) are determined using MT-fill (minimum transition fill) [6], in order to reduce the switching. When shifting out the last test response we assume that zeros are shifted in.

Now we can treat the wrapper chains as larger scan chains and calculate the number of transitions with the methods in Subsection 3.1. Thus, for each wrapper chain wc_k we have a transition function tr_k, each one constituting a part in the total transition function

$$tr(i) = \sum_{k=1}^{n_{wc}} tr_k(i), \quad (9)$$

where n_{wc} denotes the number of wrapper chains.

Thus, we calculate the transition function for each wrapper chain configuration, and we have different power profiles for different wrapper configurations (connections of the wrapper input/output cells and scan chains). Further, the number of transitions during a launch-and-capture cycle is given as the number of bits that differ in the test stimulus and its corresponding expected test response (the Hamming distance between the test stimulus and the expected test response).

4 Test Scheduling

In order to show how the proposed power model can be used in test scheduling, we will in this section describe a test scheduling heuristic for given fixed-width test bus architectures. The problem that we will study in this section is stated as follows.

Problem formulation: Given an SoC with N cores $C_1, \ldots, C_N$, a maximum power limit $P_{\max}$, and a test bus architecture with M TAMs where w_j is the width of TAM$_j$. For each core, design a wrapper (partition the scan chains and wrapper cells into a given number of wrapper chains), assign the core to a TAM, and determine the order of test execution such that the total testing time of the SoC is minimized, while the test power consumption does not exceed $P_{\max}$.[1] ■

The power curve for a core depends on the switches caused by the test stimuli and the test responses. The transitions also depend on the number of wrapper chains as the wrapper chain configuration determines the organization of test bits. We have, for the subproblem of finding the wrapper chain configurations for a core, made use of the Design_Wrapper algorithm by Iyengar *et al.* [19].

Figure 7 shows the pseudo-code for the proposed test scheduling algorithm. In Table 2 we have collected explanations of some of the notations that are used in the algorithm description.

The main idea of the heuristic is to select a core and a TAM such that we will get *best fit* to the current schedule. This is illustrated in Figure 8 where TAM 1 is used and there are two alternatives, A and B for

[1] With this problem definition we are not only dealing with classical scheduling, but also test architecture design.

Notation	Description
$C_i^{(w_j)}$	Core i with wrapper designed by Design_Wrapper [19] and wrapper width w_j.
$P_{i,j}$	The test power consumption function for core $C_i^{(w_j)}$. The domain of definition, $\mathscr{D}(P_{i,j})$, is the clock cycles for which the core is tested.
$\tau\left(C_i^{(w_j)}\right)$	Testing time for core $C_i^{(w_j)}$ according to equation (8).
$\tau\left(\mathrm{TAM}_j\right)$	Scheduled time on TAM_j. This is initially zero.
$Area(P_{i,j})$	The area under the graph of the discrete function $P_{i,j}$, that is, $\sum_{k\in\mathscr{D}(P_{i,j})} P_{i,j}(k)$.

Table 2: Notations for the scheduling algorithm

1: **repeat**
2: Find TAM_{max} with current most scheduled time
3: Schedule core C_I on TAM_J such that

$$\tau\left(\mathrm{TAM}_{max}\right) - \left(\tau\left(\mathrm{TAM}_J\right) + \tau\left(C_I^{(w_J)}\right)\right)$$

is minimum, > 0, and without exceeding $P_{\max}$.
4: **if** there is no such pair (C_I, TAM_J) **then**
5: **for all** unscheduled cores C_i **do**
6: Choose TAM_{J_i} such that

$$\tau\left(\mathrm{TAM}_{J_i}\right) + \tau\left(C_I^{(w_{J_i})}\right) - \tau\left(\mathrm{TAM}_{max}\right)$$

is minimum, and without exceeding $P_{\max}$.
7: **end for**
8: Choose an unscheduled core C_I which maximizes $Area\left(P_{I,J_I}\right)$; schedule C_I on TAM_{J_I}.
9: **end if**
10: **until** all cores are scheduled

Figure 7: The test scheduling algorithm

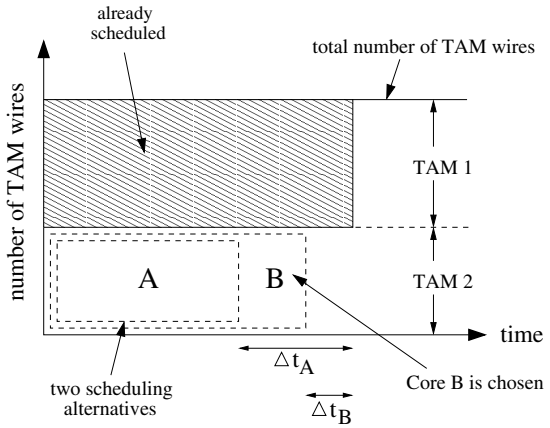

Figure 8: Illustration of the best fit principle used in the test scheduling algorithm

TAM 2. For this example core B will be scheduled since it gives best fit to the current schedule, that is, $\Delta t_B < \Delta t_A$.

The scheduling algorithm keeps track of the total power consumption profile for the current test schedule. The total power consumption is initially zero, since no tests are scheduled at the beginning of the algorithm. When a core is considered for scheduling, say core $C_i^{(w_j)}$, the corresponding power consumption function $P_{i,j}$ is accumulated to the total power consumption profile, whereafter the power constraint is checked (lines 3 and 6 in Figure 7). For the global peak power model this is fast, since we only need to keep track of one value per test, that is, $P_{i,j}$ is a constant function. For the proposed cycle-accurate model, however, the power consumption function $P_{i,j}$ is non-constant and given by the cycle-accurate power modeling technique described in Section 3. Therefore, we need to check each clock cycle, and consequently this will have a larger impact on the runtime of the test scheduling heuristic.

5 Experimental Results

We have implemented the test scheduling technique and performed experiments to demonstrate the gain in total test application time for core-based SoCs when using the proposed cycle-accurate model compared to using the global peak power model.

We used the SoCs d695, p22810, and p93791 of the ITC'02 benchmarks, as well as an industrial design (8 cores). For d695 and the industrial design we used real test data and filled the X's in the test stimuli according to MT-fill (minimum transition fill) [6] and the test responses in the most pessimistic way (maximum transition fill) as responses cannot be controlled as test stimuli. The test patterns for d695 are generated by Miyase and Kajihara [20]. For p22810 and p93791 no netlists are available; hence we filled the given number of test patterns of the given length with randomly generated test data. In order to compare with Huang *et al.* [9] and Pouget *et al.* [10], we scaled the transition count to correspond to the power values.

The experiments were conducted with different values of $P_{\max}$, the maximum allowed power consumption, and w_{TAM}, the total number of TAM wires. For each value of w_{TAM} we have run the test scheduling algorithm (Figure 7) on all possible fixed-width test bus architectures with a given maximum number of TAMs.

The CPU times for scheduling of tests, where the power profiles for each wrapper chain configuration are given and the test architecture (TAM partitioning) is given, is low, only seconds. As we try all pos-

d695	$P_{\max} = 1500$				$P_{\max} = 1800$			
w_{TAM}	[9]	GP	New	$\Delta t/t_{GP}$	[9]	GP	New	$\Delta t/t_{GP}$
16	45560	47009	44936	4.4 %	44341	45466	44502	2.1 %
24	31028	31458	30663	2.5 %	29919	30926	30663	0.9 %
32	27573	27544	23169	15.9 %	24454	25048	22544	10.0 %
40	20914	23937	19200	19.8 %	20467	21344	18799	11.9 %
48	20914	20842	17013	18.4 %	18077	19607	16686	14.9 %
56	16841	18909	15230	19.5 %	14974	18553	13185	28.9 %
64	16841	16875	12941	23.3 %	14899	16450	11526	29.9 %
d695	$P_{\max} = 2000$				$P_{\max} = 2500$			
w_{TAM}	[9]	GP	New	$\Delta t/t_{GP}$	[9]	GP	New	$\Delta t/t_{GP}$
16	43221	44870	44502	0.8 %	43221	44502	44502	0 %
24	29419	30926	30506	1.4 %	29023	30926	30336	1.9 %
32	24171	25048	22544	10.0 %	23721	23525	22544	4.2 %
40	19206	20925	18799	10.2 %	19206	18988	18799	1.0 %
48	17825	18553	16506	11.0 %	15847	16506	16506	0 %
56	14128	17013	13185	22.5 %	14128	14834	13185	11.1 %
64	14128	14397	11526	19.9 %	12933	13098	11526	12.0 %

Table 3: Power constrained test scheduling on d695 (the average improvement is 10.4 %)

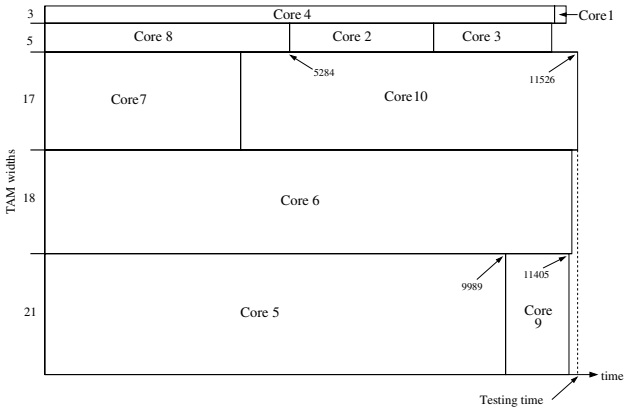

Figure 9: The resulting test schedule for d695 with the constraints w= 64 and $P_{\max}$= 1800

sible TAM partitions (for a given maximum number of TAMs), the runtime for test scheduling and TAM architecture design is higher (up to hours). The CPU time to construct the power profile for a given wrapper chain configuration is reasonable, minutes, however, as all configurations are needed, it becomes time consuming, up to hours.

In Figure 9, we have depicted the resulting test schedule for the case $w = 64$ and $P_{\max} = 1800$. We have collected the obtained testing times in Tables 3–6, where *GP* indicates the global peak power model and *New* indicates the proposed cycle-accurate test power model. We have also included the relative improvements of the cycle-accurate power model compared to the global peak power model. The percentages are calculated as $\Delta t/t_{GP} = (t_{New} - t_{GP})/t_{GP}$, where t_{GP} and t_{New} are the testing times obtained from our test scheduling algorithm when we use the global peak power model and the proposed cycle-accurate model respectively.

In general, we can see that it is possible to decrease the test application time by using the cycle-accurate power model, compared to the global peak power model. Note also that when we have very loose power constraints, that is, when $P_{\max}$ is large, the difference in the obtained testing times for the cycle-accurate power model and the global peak power model is small. This is because the test scheduler can produce near-optimal test schedules even with a single-value power model. Whenever the power constraint is tight, we conclude that the cycle-accurate power model helps the test scheduler to produce test schedules with lower testing times.

Occasionally, we get negative results in the test application times for the two power models. This is because of the particularities of the test scheduling heuristic. Note that the scheduling algorithm does not exclude the possibility of producing better test application times for the global peak power model than the cycle-accurate model, because the algorithm can get stuck in local optima. However, in general we get reduced testing times by using the cycle-accurate power model, especially for low values on $P_{\max}$. More concrete, the average reduction in testing time for the cycle-accurate power model is, for all presented experiments, 16.5 % relative to the global peak power model.

p22810	$P_{\max} = 3000$				$P_{\max} = 4000$			
w_{TAM}	[10]	GP	New	$\Delta t/t_{GP}$	[10]	GP	New	$\Delta t/t_{GP}$
8	948481	1012296	901607	10.9 %	891457	971411	901607	7.7 %
16	482963	664511	536978	19.2 %	480223	486288	490705	-0.9 %
24	392525	607451	395389	34.9 %	389243	439415	373392	15.0 %
32	309255	543358	349530	35.7 %	324478	373375	340972	8.7 %
40	356215	427876	314067	26.6 %	285307	317522	294380	7.3 %
48	311632	363299	287642	20.8 %	285814	280548	262461	6.4 %
56	293528	362487	280548	22.6 %	268272	280548	261224	6.9 %
64	293021	350162	280548	19.9 %	268856	280548	261224	6.9 %
p22810	$P_{\max} = 5000$				$P_{\max} = 6000$			
w_{TAM}	[10]	GP	New	$\Delta t/t_{GP}$	[10]	GP	New	$\Delta t/t_{GP}$
8	891457	893226	893231	0 %	893050	893293	893231	0 %
16	475026	504509	458812	9.1 %	475951	481217	460463	4.3 %
24	382507	365562	331169	9.4 %	346461	343160	320316	6.7 %
32	321930	320386	275106	14.1 %	250487	287407	253338	11.9 %
40	264038	289649	240829	16.9 %	209559	240125	199748	16.8 %
48	266166	229998	225179	2.1 %	175928	216632	187576	13.4 %
56	257600	219244	215183	1.9 %	159636	207606	180063	13.3 %
64	246110	219244	197593	9.9 %	157568	185309	175418	5.3 %
p22810	$P_{\max} = 8000$				$P_{\max} = 10000$			
w_{TAM}	[10]	GP	New	$\Delta t/t_{GP}$	[10]	GP	New	$\Delta t/t_{GP}$
8	892713	893296	881724	13.0 %	892713	893296	881724	13.0 %
16	473418	450546	450051	0.1 %	473418	450546	450051	0.1 %
24	352834	329419	308374	6.4 %	352834	329419	308374	6.4 %
32	236186	260711	241841	7.2 %	236186	260711	241841	7.2 %
40	195733	241182	199748	17.2 %	195733	241182	199748	17.2 %
48	159994	216632	184728	17.1 %	159994	216632	179482	17.1 %
56	147535	207606	166585	19.8 %	138542	207606	166585	19.8 %
64	142056	185309	160485	13.4 %	128332	185309	160485	13.4 %

Table 4: Power constrained test scheduling on p22810 (the average improvement is 11.9 %)

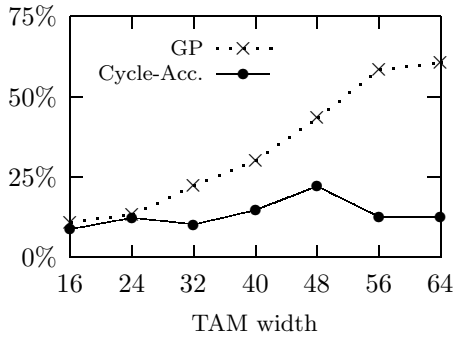

Figure 10: Relative difference between testing times and the lower bounds [21] for d695 with $P_{\max}$= 1800

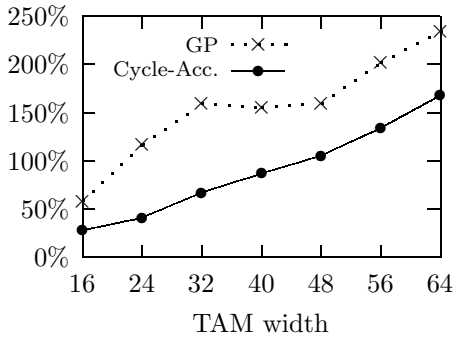

Figure 11: Relative difference between testing times and the lower bounds [21] for p22810 with $P_{\max}$= 3000

p93791	$P_{\max} = 15000$			
w_{TAM}	[10]	GP	New	$\Delta t/t_{GP}$
8	3574150	3638610	3608165	0.8 %
16	1827816	1875531	1835205	2.2 %
24	1220469	1238982	1238983	0 %
32	1014616	947134	934069	1.4 %
40	848050	780133	773848	0.8 %
48	631214	645532	639232	1.0 %
56	598481	627670	539815	14.0 %
64	486469	485297	497739	-2.6 %
p93791	$P_{\max} = 20000$			
w_{TAM}	[10]	GP	New	$\Delta t/t_{GP}$
8	3574150	3585140	3585267	0 %
16	1827816	1835416	1829232	0.3 %
24	1220469	1233680	1233716	0 %
32	957921	932323	934069	-0.2 %
40	821575	766353	769378	-0.4 %
48	658132	640602	640615	0 %
56	549481	550636	539815	2.0 %
64	472653	485297	492463	-1.5 %
p93791	$P_{\max} = 25000$			
w_{TAM}	[10]	GP	New	$\Delta t/t_{GP}$
8	3574150	3638610	3638708	0 %
16	1827816	1829176	1829232	0 %
24	1220469	1233680	1233716	0 %
32	965383	929974	934069	0 %
40	821475	748140	748154	0 %
48	639217	612046	625476	-2.7 %
56	549669	545322	539815	1.0 %
64	493599	485297	481893	0.7 %

Table 5: Power constrained test scheduling on p93791 (the average improvement is 0.7 %)

industrial	$P_{\max} = 12000$		
w_{TAM}	GP	New	$\Delta t/t_{GP}$
24	153410308	62299271	59.4 %
32	76859936	47860926	37.7 %
40	76859936	47860926	37.7 %
48	76859936	47860926	37.7 %

Table 6: Power constrained test scheduling on the industrial design (the average improvement is 43.1 %)

We should mention that in the tables with the obtained test application times, we have also presented the results from some related work on power constrained test scheduling [9], [10]. In these studies, however, the authors consider a flexible-width test bus architecture, which allows for more test time reduction than the fixed-width test bus architecture. Therefore, our approach is not fully comparable with their. Nevertheless, we can still see indications that using the cycle-accurate power model would improve the results even in other existing test scheduling algorithms [9], [10].

We have also tried to quantify the quality of our results. We do this by comparing the obtained test application times with the lower bounds [21] on the optimal testing times. The lower bounds are for the test scheduling problem without power constraints (that is, $P_{\max} = \infty$). Figure 10 illustrates the relative difference between the testing times and the lower bounds for d695 with $P_{\max} = 1800$. In Figure 11 we illustrate the same principle for p22810 with the power constraint $P_{\max} = 3000$. We can see that using the cycle-accurate power model gives testing times closer to the lower bounds than using the global peak power model.

6 Conclusions

The increasing test application times for testing modular core-based SoCs can be minimized by concurrent execution of the tests. However, concurrent test application leads to higher power consumption, which must be taken into account in order to not violate the power constraint. The power consumption has previously been modeled as a single value per test. In this paper we proposed a power model with a power value per clock cycle that is based on an analysis of the test stimuli and the test response. The power model takes the scan chain switching activity generated by the test stimuli and the test responses into account. Further, the model provides a separate power profile per wrapper chain configuration. We have implemented the power model and included it in an SoC test scheduling algorithm. We have made extensive experiments on several ITC'02 benchmarks and an industrial design, where we compare the testing time when using a single-value (global peak) power model and the proposed cycle-accurate power model. The results demonstrate that significant testing time can be saved by making use of our more elaborate power model.

Acknowledgments

The authors thank *Prof. Seiji Kajihara* and *Kohei Miyase* at Kyushu Institute of Technology in Japan, for providing test pattern files for the SoC d695.

References

[1] R. Chou, K. Saluja, and V. Agrawal, "Scheduling Tests for VLSI Systems Under Power Constraints," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 2, pp. 175–185, 1997.

[2] ITC'02 SOC Test Benchmarks, `http://www.hitech-projects.com/itc02socbenchm/`.

[3] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SOCs," in *Proceedings IEEE International Test Conference*, 2002, pp. 519–528.

[4] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, "Power Profile Manipulation: A New Approach for Reducing Test Application Time Under Power Constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 10, pp. 1217–1225, 2002.

[5] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for Minimizing Power Dissipation in Scan and Combinatorial Circuits During Test Application," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 12, pp. 1325–1333, 1998.

[6] S. Ghosh, S. Basu, and N. A. Touba, "Joint Minimization of Power and Area in Scan Testing by Scan Cell Reordering," in *Proceedings IEEE Computer Society Annual Symposium on VLSI*, 2003, pp. 246–249.

[7] P. Flores, J. Monteiro, and J. Marques-Silva, "Assignment and Reordering of Incompletely Specified Pattern Sequences Targetting Minimum Power Dissipation," in *Proceedings Twelfth Conference On VLSI Design*, 1999, pp. 37–41.

[8] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Power Driven Chaining of Flip-flops in Scan Architectures," in *Proceedings IEEE International Test Conference*, 2002, pp. 796–803.

[9] Y. Huang, S. Reddy, W. Cheng, P. Reuter, N. Mukherjee, C. Tsai, O. Samman, and Y. Zaidan, "Optimal Core Wrapper Width Selection and SOC Test Scheduling Based on 3-D Bin Packing Algorithm," in *Proceedings IEEE International Test Conference*, 2002, pp. 74–82.

[10] J. Pouget, E. Larsson, and Z. Peng, "SOC Test Time Minimization Under Multiple Constraints," in *Proceedings Asian Test Symposium (ATS)*, 2003, pp. 312–317.

[11] K. Chakrabarty, "Design of system-on-a-chip test access architectures under place-and-route and power constraints," in *DAC '00: Proceedings of the 37th conference on Design automation*, 2000, pp. 432–437.

[12] E. Larsson and Z. Peng, "Test Scheduling and Scan-Chain Division Under Power Constraints," in *Proceedings Asian Test Symposium (ATS)*, 2001, pp. 259–264.

[13] P. Rosinger, B. Al-Hashimi, and K. Chakrabarty, "Rapid generation of thermal-safe test schedules," in *Proceedings of the Design, Automation and Test in Europe Conference*, 2005, pp. 840–845.

[14] C. Liu, K. Veeraraghavan, and V. Iyengar, "Thermal-Aware Test Scheduling and Hot Spot Temperature Minimization for Core-Based Systems," in *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2005, pp. 552–560.

[15] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital integrated circuits – a design perspective*, 2nd ed. Prentice Hall, 2003.

[16] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," in *Proceedings IEEE VLSI Test Symposium*, 2000, pp. 35–40.

[17] M. Selkälä, "Test Data Analysis for Accurate Power Estimation," Master's thesis, Linköping University, 2006.

[18] E. J. Marinissen, S. K. Goel, and M. Lousberg, "Wrapper Design for Embedded Core Test," in *Proceedings IEEE International Test Conference*, 2000, pp. 911–920.

[19] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip," in *Proceedings IEEE International Test Conference*, 2001, pp. 1023–1032.

[20] K. Miyase and S. Kajihara, "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 2, pp. 321–326, 2004.

[21] S. K. Goel and E. J. Marinissen, "Effective and Efficient Test Architecture Design for SOCs," in *Proceedings IEEE International Test Conference*, 2002, pp. 529–538.