Partition Based SoC Test Scheduling with Thermal and Power Constraints under Deep Submicron Technologies

Chunhua Yao, Kewal K. Saluja, Parameswaran Ramanathan
Department of Electrical and Computer Engineering, University of Wisconsin-Madison
{yao,saluja,parmesh}@ece.wisc.edu

Abstract

For core-based System-on-Chip (SoC) testing, conventional power-constrained test scheduling methods do not guarantee a thermal-safe solution. Also, most of the test scheduling schemes make poor assumptions about power consumption. In deep submicron era, leakage power and wake-up power consumption can not be neglected. In this paper, we propose a partition based thermal-aware test scheduling algorithm with more realistic assumptions of recent SoCs. In our test scheduling algorithm, each test is partitioned and the earliest starting time of each partition is searched. To reduce the execution time of thermal simulation, we also exploit superposition principle to compute the power and thermal profile rapidly and accurately. We apply our test scheduling algorithm to ITC’02 SoC benchmarks and the results show improvements in the total test time over scheduling schemes without partitioning.

Keywords: SoC Test, Test scheduling, Test partition, Thermal simulation, Superposition

1. Introduction

With scaling technology and increasing design sizes, power consumption during test and test data volume have grown dramatically, especially for core-based SoC testing which allows multiple cores in testing mode concurrently. In the deep submicron era, the situation becomes worse because there are leakage power consumptions even for those idle cores. System level test scheduling strategies are normally adopted in order to reduce the test cost (specifically, the test application time) and to meet the limit power budget of the SoC system. Chou et al. [1] constructed test compatibility graph with power information and divided the problem into equal test length scheduling. Iyengar et al. [2] formulated the power-constrained test scheduling problem into a mixed-integer linear programming (MILP). A more recent solution to the power-constrained test scheduling problem by Zhao et al. [3] proposed a scheduling algorithm with a dynamical partitioning of the tests and a rectangle packing heuristic.

However, traditional power-constrained test scheduling algorithm can not guarantee a thermal safe solution. Under the deep submicron technologies, the dramatic increase in power density on chip can result in the chip temperatures easily exceeding 100°C, especially during test mode with high switching activities [4]. Thermal hotspots can cause many problems such as high cooling cost, performance degradation, reliability issues, and device damage [5].

Thermal-aware test scheduling problem considers test resource conflicts, power consumption constraint, and the temperature constraint. Rosinger et al. [6] addressed this problem and proposed a method for the generation of thermal-safe test schedules. In their later work [7], they adopted the clique cover algorithm to find a initial test schedule meeting resource and power constraints, then eliminated the cliques which violate the temperature constraint by running thermal simulations. They also proposed a thermo-resistance model to approximately and quickly compute the thermal profile. Liu et al. [8] proposed a thermal-aware test scheduling scheme based on rectangle packing heuristic and also a scheme to spread the heat more evenly over the chip and reduce hot spots. He et al. [9] assumed that running an individual long test will exceed the temperature constraint so they proposed a test set partitioning and interleaving technique and employed constraint logic programming (CLP) to generate thermal-aware test schedules. In their later work, they proposed a heuristic [10] and added the assumption that there is thermal impact on neighboring modules [11]. Bild et al. [12] developed an optimal MILP formulation for the thermal-aware test scheduling problem. They also proposed a seed-based clustering test scheduling heuristic with a phased steady-state thermal model to reduce the thermal simulation time.

In this paper, we propose a partition based thermal-aware...
test scheduling scheme while taking into account the effects of deep submicron technologies. The main contributions of this paper includes:

(i) We propose a partition based method to improve the performance of the SoC test scheduling under power and thermal constraints, and

(ii) We consider more realistic constraints for SoC cores under deep submicron technologies, including leakage power and wake-up power for idle cores, partition overhead, and power variations.

The rest of the paper is organized as follows: Section 2 discusses the limitations of related works and Section 3 explains the superposition principle to compute the power and thermal profile. The partition based thermal-aware test scheduling algorithm is described in Section 4. Section 5 shows the experimental results and the paper concludes with Section 6.

2. Limitations of related works

A simple formulation of the test scheduling problem can be described as follows:

A test set \( S \), consisting of \( n \) tests \( t_i \), \( i = 1 \) to \( n \), needs to be applied to a device under test. Each test \( t_i \) has 1) test length \( l_i \), which often is the number of test vectors in test \( t_i \), 2) test power consumption \( P_i \), which indicates the power consumption when test \( t_i \) alone is applied to the device, 3) test compatibility with a set of known tests. A test \( t_i \) is compatible with test \( t_j \) if there is no resource conflict between \( t_i \) and \( t_j \) and they can be applied to the device simultaneously.

The test scheduling problem is to find a test schedule such that:

1. The total test application time is minimized. Total test application time is defined as the earliest time at which all tests have completed.
2. There is no resource conflict between any two simultaneously running tests.
3. The total power consumption of the device at all times is smaller than the power constraint, \( P_{\text{max}} \).
4. The temperature at every location in the device at all times is smaller than the temperature constraint, \( T_{\text{max}} \).

In reality, the power consumption for a test varies in every clock cycle. However, almost all power-constrained test schedule schemes assume uniform power consumption during the lifetime of each individual test. The uniform power consumption value can be either the maximum power consumption or the average power consumption of the test. This assumption works well under a loose power constraint. However, with the sharply increased power density and tighter power constraint, a more realistic model is needed. In this paper, we do not assume uniform power consumption, instead we account for the actual power consumption in every test cycle within each individual test. We also assume that each test has different stages which will have different average power consumptions. This is true because there are different testing requirements at different stages. For example, the power consumption of the portion of test that targets hard-to-detect faults is larger than the power consumption of the portion of the same test that targets easy-to-detect faults. This is the basis of our proposed partition based scheme and will be discussed more in section 4. Another shortcoming of uniform power consumption assumption is that a new test need only to be scheduled after the completion of the old test if scheduling the new test and the already scheduled test violates the power constraint. Under the more realistic assumption that power consumption changes each cycle, we may be able to schedule a test with partial overlap without violating the power constraint.

It’s well known that the leakage power consumption becomes more important under deep submicron technologies. The idle cores, which have no test running on them, still consume large amount of leakage power and can not be neglected. Also, the power consumed in wake-up a core before applying a test can not be ignored. The power consumption to wake up the idle core can be larger than the active power consumption and further the wake-up latency can be as long as hundreds of microseconds [13]. In this paper, we consider both leakage power consumption and wake-up power consumption of idle cores. We also assume a reasonable wake-up latency before starting a new test.

For thermal-aware test scheduling, thermal simulation is the bottleneck. There are two common ways to perform thermal simulation during a thermal-aware test scheduling: 1) Invoke thermal simulation tools, such as HotSpot [14]. The advantage of this method is the accuracy but the obvious shortcoming is the execution time. Every time a thermal simulation is needed, the tool must be invoked. For test scheduling of SoC with large number of cores, we may have to invoke thermal simulation a large number of times. 2) Develop simplified thermal model. This method can integrate the thermal simulation into the test scheduling algorithm and reduce the entire execution time. However, those models are not normally very accurate and will affect the performance of the thermal-aware test scheduling. In this paper, we exploit a method to compute the thermal profile rapidly and accurately using the well-known superposition principle. Our method is based on the algorithm we proposed in [15]. The method in [15] did not consider leakage and wake-up power consumption and overhead and we also used the simple uniform power consumption assumption.

Partition based test scheduling are also discussed in [3] and [9]. In [3], the tests are dynamically and “virtually” partitioned. Each test is still run-to-completion but it can cross into two or more test sessions. In [9], the authors assumed that running an individual long test alone will exceed the temperature constraint so they proposed a test set par-
tioning and interleaving technique based test scheduling. Even another test (or partitioned test) can be "interleaved" into the cooling period of the current test, only one core can be tested during each testing cycle (or testing session). In this paper we assume that execution of each individual test will not violate the temperature constraint, which is true in most of the SoC test cases, and we show that partition each test into small partitions can further improve the performance of power and thermal constrained test scheduling. Another realistic assumption we made is the partition overhead, which is not considered in [9]. Partitioning the test can be realized by preemption, which means stopping and resuming the test within the duration of the test. Preemption requires save/load test states and will result in partition time overhead.

In this paper, we propose a partition based thermal-aware test scheduling algorithm which can find the earliest starting time of each test partition. The starting time can be arbitrary time as long as scheduling the test partition will not violate any constraint.

3. Superposition principle

3.1 Power profile

When the SoC is powered on, all the cores are assumed to be in the idle state consuming leakage power $P_{\text{leakage}}$. When a test for one core is scheduled at time $t$, a wake-up period $l_{\text{wake}}$ with wake-up power $P_{\text{wake}}$ is added to the initial power profile from time $t$. After that, the actual test is scheduled at time $t + l_{\text{wake}}$ with test power $P_{\text{test}}$. For example, Figure 1 shows a core-based SoC with three cores: $C1$, $C2$ and $C3$. Test for $C1$, $T1$, has been scheduled at time 0 and test for $C2$, $T2$, has been scheduled at time 6. The power trace for the wake-up period is set to be $P_{\text{wake}} - P_{\text{leakage}}$. The reason to do so is that instead of "replacing" the leakage power with the wake-up power, we can simply add the $P_{\text{wake}} - P_{\text{leakage}}$ to the initial leakage power $P_{\text{leakage}}$ and then it becomes the wake-up power $P_{\text{wake}}$. This will further simplify the computation complexity of the superposition operations. Similarly, during the test period the power trace is set to be $P_{\text{test}} - P_{\text{leakage}}$. The superposition principle to compute the power profile of the entire SoC chip works as follows: First, the power profile for the entire SoC is initialized with leakage power consumptions of each core, as seen in the left most figure of Figure 1. When $T1$ is scheduled at time 0, the power consumption for core $C1$ from time 0 is added with the wake-up power consumption and then the test power consumption of test $T1$ until the end of the test. Similarly, the power consumption for core $C2$

\[ P_{\text{test}} = P_{\text{test}} + P_{\text{leakage}} \]

from time 6 is added with the wake-up power consumption and then the test power consumption of test $T2$. At each cycle, the sum of power consumption of each core is the total power consumption of the entire SoC. For a power constrained test scheduling, a valid schedule should ensure that the total power consumption of the entire SoC at each cycle is less than the power constraint $P_{\text{max}}$.

3.2 Thermal profile

The popular model used in thermal simulation is RC model [4]. It is well know that the RC model is a linear model. As a result, we exploit the superposition principle to compute the thermal profile. The superposition method is fast and as accurate as the thermal simulation tools. The computation of thermal profile of the entire chip using the superposition principle works as follows:

1. Initialize all cores with leakage power $P_{\text{leakage}}$ and generate the initial thermal profile.
2. Apply each individual test to the SoC alone and start at time 0. Generate the thermal profile for each test with an initial temperature of 0. Each test consists of the wake-up and partition overhead $l_{\text{overhead}}$ and the actual test duration $l_{\text{test}}$. For the same reason as for the power profile, the power trace from for wake-up is set to be $P_{\text{wake}} - P_{\text{leakage}}$ and the power trace for test is set to be $P_{\text{test}} - P_{\text{leakage}}$. Thermal simulations for wake-up power and test power are performed separately because of possible partition mergers in our partition based test scheduling. Figure 2 shows example thermal profiles for a SoC consisting of three modules. Figure 2(a) shows the initial thermal profile for leakage power with an initial temperature of $45^\circ C$. Figure 2(b) and 2(c) shows thermal profiles of test $T1$ and $T2$ when they are applied individually with an initial temperature of $0^\circ C$.
3. To compute the thermal profile when there are more than one test running simultaneously, the superposition principle is used. For a test which starts at time $t$, the thermal profile of that test is shifted by $t$ cycles and added to
the initial thermal profile. For example in Figure 2, test $T_1$ starts at time 0 and test $T_2$ starts at time 6. The thermal profile of running them together is computed by shifting the thermal profile of $T_2$ by 6 and adding the thermal profile of $T_1$ and adding the initial thermal profile of leakage power. The final thermal profile of the test schedule is shown in Figure 2(d).

The use of the superposition principle implies that the thermal simulation tool is run only once at the beginning. Then the computation of thermal profile becomes simple matrix addition operations, which significantly reduce the execution time of thermal simulation. In our study we use the thermal simulation tool, HotSpot, and we verified that it follows the superposition principle.

4. Partitioned based test scheduling

4.1 Test partition

As mentioned in Section 2, test for each core can be divided into different stages and each stage has different average power consumptions. The power consumption of the test that targets hard-to-detect faults can be much larger than the power consumption of the test that targets easy-to-detect faults. Partitioning the tests according to testing stages will result in test partitions with significantly different power consumptions thus benefit the power and thermal constrained test scheduling. A simple motivation example can be seen in Figure 3. Two tests, $T_1$ and $T_2$ are to be scheduled. $T_1$ has a test power consumption of 10 and test length of 20. $T_2$ has a test power consumption of 15 and test length of 20. Supposing a power constraint of 20, the final schedule without partition is shown in Figure 3(c) and total test length is 40. By partitioning $T_1$ and $T_2$ into three partitions, each partition will have different power and different length, as shown in Figure 3(a) and 3(b) where $P$ means test power consumption and $L$ means test length. Under the same power constraint of 20, the test schedule with partition is shown in Figure 3(d), which has a smaller total test length of 32. The reduction of total test length by partition comes from overlapping between two test partitions with lower power consumptions. In this example each partition is assumed to have uniform power consumption. However, it is easy to see that this is not a strict requirement.

4.2 Test scheduling algorithm

The pseudo-code of the proposed thermal-aware test scheduling algorithm is given in Figure 4. First, all the tests are partitioned into partitions. Then the power and thermal profile of running each partition alone is generated using thermal simulation tool. During the scheduling process, the resource conflict is checked first. If there is a resource conflict, the next possible starting time is the latest completion time of all conflicting tests. If there is no resource conflict, the power and the temperature constraints are checked by using the superposition principle for power and thermal profile computation. Actually, it is because of this efficient superposition power and thermal profile computation that it is easy to see that this is not a strict requirement.
tions from the same original test are scheduled together, there is no extra partition and wake-up overhead. If two partitions are scheduled separately, the wake-up and partition overhead are added before the actual test is started. In the proposed partition based test scheduling algorithm, we use the same list schedule scheme as described in [15].

Partition each test into \( n \) partitions;
Initial power and thermal profile generation;
For each test partition:
\[
\text{Start\_Time} = 0; \quad \text{Scheduled} = \text{FALSE};
\]
\[
\text{while (Scheduled} = \text{FALSE})
\]
\[
\text{Schedule partition at Start\_Time;}
\]
\[
\text{if (Resource conflict} = \text{TRUE})
\]
\[
\text{Start\_Time} = \text{Completion time of conflicting partitions;}
\]
\[
\text{Scheduled} = \text{FALSE};
\]
\[
\text{else} \quad \text{if (Validate power or thermal constraint)}
\]
\[
\text{Start\_Time} += \text{Completion time of conflicting partitions;}
\]
\[
\text{else}
\]
\[
\text{Scheduled} = \text{FALSE};
\]
\[
\text{else}
\]
\[
\text{Scheduled} = \text{TRUE};
\]
\[
\text{Endif}
\]
\[
\text{// All test partitions scheduled}
\]
\[
\text{Output test schedule;}
\]

Figure 4: Partition based test scheduling algorithm

5. Experimental results

We ran our thermal-aware test scheduling algorithm on ITC’02 SoC benchmarks [16]. We first specify the necessary information needed by the thermal-aware test scheduling such as SoC floorplan, power trace, resource conflict graph etc. The test length and test resource conflict information are specified as described in [15]. In our study each test is divided into three partitions. Each partition can be one of the relatively high, medium, or high power test for a given core. Note that it is possible that all partitions of a test consume high, medium, low, or a mixture of these powers with appropriate variations in power. For a more realistic assumption, we consider both temporal and lateral variations of power consumptions. We divide each core into several modules and assign different power consumptions to each module. Meanwhile, as we mentioned in Section 2, the power consumptions at each cycle are also different. Table 1 gives the specified information for all the SoC benchmarks. The first column is the name of the benchmark. The second column \( N_{\text{core}} \) gives number of cores in each SoC benchmark. The range of test length is shown at column 3 as \( t_{\text{test}} \). The range of power consumptions are given in last three columns, where \( P_{\text{test}} \) means test power consumption, \( P_{\text{wake}} \) means wake-up power consumption and \( P_{\text{leakage}} \) means leakage power consumption respectively.

The thermal-aware test scheduling results of the modified SoC benchmarks are shown in Figure 5 and Figure 6. Figure 5 shows the results without considering partition and wake-up overhead. In this figure, we also compare the results of traditional test scheduling scheme which assumes an uniform power consumption for each test. For uniform power we set the maximum power consumption value within the test as the uniform power for that test and we use the test scheduling scheme of [12]. In the figure, the bar labeled Without Partition and With Partition show the total test length of proposed test scheduling which considers power variation at each cycle, with and without partition respectively. The figure clearly shows that considering the power variation at each cycle will largely reduce the total length of test scheduling. Figure 6 shows the results with partition and wake-up overhead. From both figures we can see that for all the SoC benchmarks, test partition benefits the power and thermal constrained test scheduling. When no overhead is considered, the average reduction of total test length is about 10%. When the partition and wake-up overhead is considered, the average reduction of total test length is about 8%. The improvement becomes smaller when considering overhead because each partition will bring extra partition and wake-up test time overhead. However, for the SoC benchmark \( q12710 \), a larger reduction of test length with overhead is seen. The reason is that our algorithm can search the earliest starting time for each partition so the partition and wake-up overhead can be potentially overlapped with other tests.

As mentioned before, we assume each test is partitioned into three partitions. For further research, the optimal number of partitions need to be found by considering the trade-off between the increase of extra partition overhead and reduction of total test length. In the extreme case, the test can be partitioned into one-cycle test vector and then merge them back to find the optimal number of partitions and optimal location to partition.

6. Conclusions

In this paper, we propose a partition based thermal-aware SoC test scheduling algorithm with realistic assumptions.
under deep submicron technologies. We consider the leakage power consumption and wake-up power consumption of idle cores. By using the test partition method, the algorithm can adaptively search the earliest starting time of each test partition and generate test schedule which meets resource compatibility, power constraint and temperature constraint. We also exploit superposition principle to rapidly and accurately compute the power and thermal profile. Applying our algorithm to ITC’02 SoC benchmarks shows that the proposed algorithm can reduce the total test time as compared to test scheduling methods without test partitioning.

References


