

CHALLENGES AND SOLUTIONS FOR THERMAL-AWARE SOC TESTING

Zebo Peng, Zhiyuan He and Petru Eles

Embedded Systems Laboratory, Linköping University, Sweden

INVITED PAPER

MIDEM 2007 CONFERENCE - WORKSHOP ON ELECTRONIC TESTING

12.09. 2007 - 14.09. 2007, Bled, Slovenia

Key words: electronic testing, SoC devices, thermal-aware SoC testing techniques, test efficiency

Abstract: High temperature has negative impact on the performance, reliability and lifespan of a system on chip. During testing, the chip can be overheated due to a substantial increase of switching activities and concurrent tests in order to reduce test application time. This paper discusses several issues related to the thermal problem during SoC testing. It will then present a thermal-aware SoC test scheduling technique to generate the shortest test schedule such that the temperature constraints of individual cores and the constraint on the test-bus bandwidth are satisfied. In order to avoid overheating during the test, we partition test sets into shorter test sub-sequences and add cooling periods in between. Further more, we interleave the test sub-sequences from different test sets in such a manner that the test-bus bandwidth reserved for one core is utilized during its cooling period for the test transportation and application of the other cores. We have developed a heuristic to minimize the test application time by exploring alternative test partitioning and interleaving schemes with variable length of test sub-sequences and cooling periods. Experimental results have shown the efficiency of the proposed heuristic.

Izzivi in rešitve pri testiranju sistemov na čipu

Ključne besede: testiranje elektronike, SoC sistemi na čipu, tehnike testiranja SoC z obvladovanjem pregrevanja, učinkovitost testiranja

Izveček: Visoke temperature negativno vplivajo na lastnosti, zanesljivost in življensko dobo sistemov na čipu. Med testiranjem lahko pride do pregrevanja čipa zaradi povečanega števila vklopov z namenom skrajšati čas testiranja. V prispevku opišemo tovrstne probleme in predstavimo tehnike, s katerimi načrtujemo take testne procedure, ki vodijo računa o tem, da ne prihaja do pregrevanja posameznih delov čipa. Z namenom preprečiti pregrevanje smo teste razdelili na krajše testne periode z vmesnimi pavzami za hlajenje elektronike. Dodatno smo programirali testna zaporedja tako, da so se signali širili preko testnih poti selektivno k posameznim delom elektronike med tem, ko so se drugi deli hladili. Izdelali smo ustrezno metodologijo, ki nam je omogočila skrajšati testne čase. Eksperimentalni rezultati so pokazali uspešnost predlagane metode.

1. Introduction

The rapid development of System-on-Chip (SoC) techniques has led to many challenges to the design and test community. The challenges to the designers have been addressed by the development of the core-based design method, where pre-designed and pre-verified building blocks, called embedded cores, are integrated together to form a SoC. While the core-based design method has led to the reduction of design time, it entails several test-related problems. How to address these test problems in order to provide an optimal test solution is a great challenge to the SoC test community /1/.

A key issue for SoC testing is the selection of an appropriate test strategy and the design of a test infrastructure on chip to implement the selected test strategy. For a core embedded in a SoC, the direct test access to its peripheries is impossible. Therefore, special test access mechanism must be included in a SoC to connect the core peripheries to the test sources and sinks. The design of the test infrastructure, including the test access mechanism, must be considered together with the test scheduling prob-

lem, in order to reduce the silicon area used for test access and to minimize the total test application time /2/, /3/, /4/, /5/, /6/, /7/, /8/.

The rapid increasing test data volume needed for SoC testing is another issue to be addressed, since it contributes significantly to long test application times and huge ATE memory requirements /7/. This issue can be addressed by sharing the same test set among several cores as well as test data compression. Both test set sharing and test compression can exploit the large percentage of don't care bits in typical test patterns generated for complex SoC designs in order to reduce the amount of test data needed /9/.

The issue of power dissipation should also be considered in order to prevent a SoC chip from being damaged by overheating during test /2/, /9/, /10/, /11/. High temperature has become a technological barrier to the testing of high performance SoC, especially when deep submicron technologies are employed. In order to reduce test time while keeping the temperature of the cores under test within a safe range, thermal-aware test scheduling tech-

niques are required, and this paper discussed several issues related to thermal-aware SoC testing.

Thermal-aware testing has recently attracted many research interests. Liu et al. proposed a technique to evenly distribute the generated heat across the chip during tests, and therefore avoid high temperature /12/. Rosinger et al. proposed an approach to generate thermal-safe test schedules with minimized test time by utilizing the core adjacency information to drive the test scheduling and reduce the temperature stress between cores /13/. In our previous work /14/, we proposed a test set partitioning and interleaving technique, and employed constraint logic programming (CLP) to generate thermal-aware test schedules with the minimum test application time (TAT).

In our work, we assume that a continuous test will increase the temperature of a core to pass a limit beyond which the core may be damaged. In order to avoid overheating during tests, we partition the entire test set into a number of test sub-sequences and introduce a cooling period between two consecutive test sub-sequences. As the test application time substantially increases when long cooling periods are introduced, we interleaved different partitioned test sets in order to generate a shorter test schedule. In /14/, we restricted the length of test sub-sequences that belong to the same test set to be identical. Moreover, we also restricted the cooling periods between test sub-sequences from the same test set to have equal length. The main purpose of these restrictions was to keep the size of the design space small and, by this, to reduce the optimization time, so that the CLP-based algorithm will be able to generate the optimal solutions in a reasonable time.

However, these restrictions have resulted in less efficient test schedules, and longer test application times. In our recent work, we have eliminated these restrictions so that both test sub-sequences and cooling periods can have arbitrary lengths. Since breaking the regularity of test sub-sequences and cooling periods dramatically increases the size of exploration space, the CLP-based test scheduling approach proposed in /14/ is not feasible any more, especially for practical industrial designs. Therefore, new, low-complexity heuristics are needed which are able to produce efficient test schedules under the less restricted and more realistic assumptions.

The rest of this paper is organized as follows. The next section discusses the thermal issue related to SoC testing, and some solutions. It will also motivate the importance of test partitioning and interleaving with arbitrary partition/cooling lengths. Section 3 defines formally the thermal-aware test scheduling problem we are addressing in this paper. Section 4 presents the overall strategy of our thermal-aware scheduling approach, and Section 5 the proposed test scheduling heuristic. The experimental results are described in Section 6, and conclusions in Section 7.

2. The thermal issue

High temperature can be observed in most high-performance SoCs due to high power consumption. High power consumption results in excessive heat dissipation, and elevates the junction temperature which has large impacts on the operation of integrated circuits /15/, /16/, /17/, /18/.

The performance of the integrated circuits is proportional to the driving current of CMOS transistors, which is a function of the carrier mobility. Increasing junction temperature decreases the carrier mobility and the driving current of the CMOS transistors, which consequently degrades the performance of circuits /19/.

At higher junction temperature, the leakage power increases. The increased leakage power in turn contributes to an increase of junction temperature. This positive feedback between leakage power and junction temperature may result in thermal runaway and destroy the chip /19/.

The long term reliability and lifespan of integrated circuits also strongly depends on junction temperature. Failure mechanisms in CMOS integrated circuits, such as gate oxide breakdown and electro-migration, are accelerated in high junction temperature. This may result in a drop of the long term reliability and lifespan of circuits /18/.

Advanced cooling system can be one solution to the high temperature problems. However, the cost of the entire system will substantially increase, and the size of the system is inevitably large.

The thermal issue becomes even more severe in the case of testing than in normal functional mode, since testing dissipates more power and heat due to a substantial increase of switching activities /18/. In order to prevent excessive power during test, several techniques have been developed. Low power DFT and test synthesis techniques can be utilized, including low-power scan chain design /20/, /21/, as well as scan cell and test pattern reordering /21/, /23/, /24/. Although low power DFT can reduce the power consumption, such techniques usually add extra hardware into the design and therefore can increase the delay and the cost of the produced chips.

For many modern SoC designs, when a long sequence of test patterns is continuously applied to a core, the temperature of this core may increase and pass a certain limit beyond which the core will be damaged. In such scenarios, the test has to be stopped when the core temperature reaches the limit, and can be restarted later when the core has been cooled down. Thus, by partitioning a test set into shorter test sub-sequences and introducing cooling periods between them, we can avoid the overheating during test. Figure 1 illustrates the temperature profile of a core under test when the entire test set for the core is partitioned into four test sub-sequences, TS_1 , TS_2 , TS_3 , and TS_4 , and cooling periods are introduced between the sub-

sequences. In this way, the temperature of the core under test remains within the imposed temperature limit.

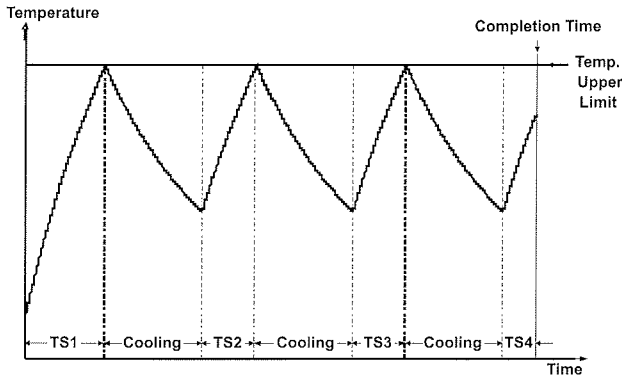


Fig. 1. Illustration of test set partitioning

It is obvious that introducing long cooling periods between test sub-sequences will substantially increase the test application time (TAT). To address this problem, we can reduce the TAT by interleaving the partitioned test sets such that the test-bus bandwidth reserved for a core C_i , during its cooling periods, are utilized to transport test data for another core C_j ($j \neq i$), and thereafter to test the core C_j . By interleaving the partitioned test sets belonging to different cores, the test-bus bandwidth is more efficiently utilized. Figure 2 gives an example where two partitioned test sets are interleaved so that the test time is reduced with no need for extra bus bandwidth.

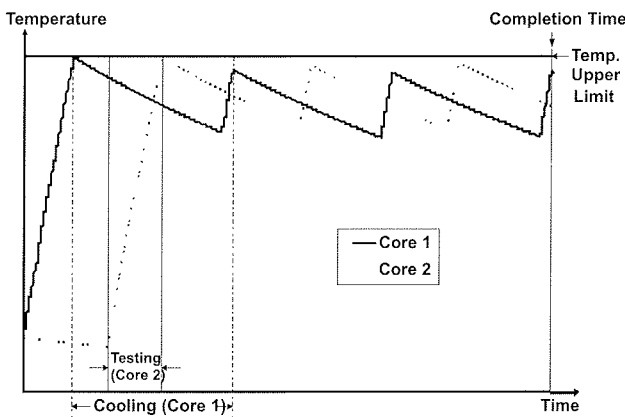


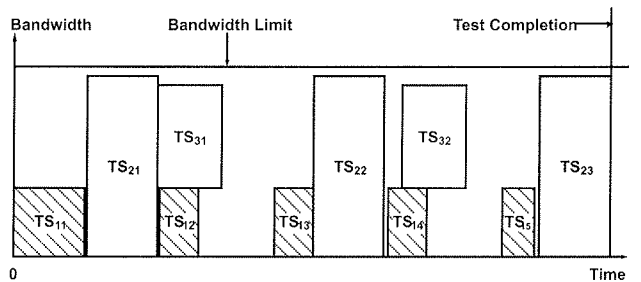
Fig. 2. Illustration of test set interleaving

There are many design alternatives which can be used to implement the above basic ideas of test partitioning and interleaving for thermal-aware test scheduling. In general, the objective is to minimize the test application time by generating an efficient test partitioning/interleaving scheme and to schedule the individual test sub-sequences which avoids violating the temperature limits of individual cores, and, at the same time, satisfies the test-bus bandwidth constraint. This is a complex optimization problem, and we have developed, as mentioned in the previous section, a solution for the problem with the restriction that the length of the test sub-sequences from the same test set should be identical and the cooling periods between test sub-

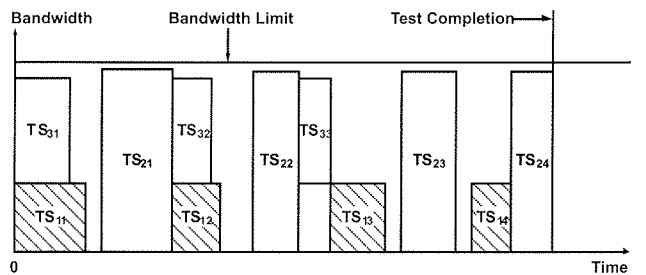
sequences from the same test set should have equal length /14/. However, this restriction has resulted in less efficient test schedules, and thus longer test application times.

To illustrate the usefulness of eliminating this restriction so that both test sub-sequences and cooling periods can have arbitrary lengths, each test sub-sequence can be considered as a rectangle, with its height representing the required test-bus bandwidth and its width representing the test time.

Figure 3 gives an example where three test sets, TS_1 , TS_2 , and TS_3 , are partitioned into 5, 3, and 2 test sub-sequences, respectively. Note that the partitioning scheme which determines the length of test sub-sequences and cooling periods has ensured that the temperature of each core will not violate the temperature limit, by using a temperature simulation /19/. Figure 3(a) shows a feasible test schedule under the regularity assumption (identical test sub-sequence length and identical cooling periods for each core). In Figure 3(b), an alternative test schedule is depicted, where the test sub-sequence and the cooling periods can have arbitrary lengths. This example shows the possibility to find a shorter test schedule by exploring alternative solutions, where the number and length of test sub-sequences, the length of cooling periods, and the way that the test sub-sequences are interleaved are different from those in Figure 3(a).



(a) A test schedule with regular partitioning scheme



(b) An alternative test schedule with irregular partitioning scheme

Fig. 3. Comparison of two different test schedules

3. Problem formulation

We have assumed a test architecture using a single test bus to transport test data between the tester and the cores

under test. A tester can be either an external automated test equipment (ATE) or an embedded tester integrated on the chip. Each core under test is connected to the test bus with a number of dedicated TAM wires. The test patterns, together with a generated test schedule, are stored in the tester memory. A test controller controls the entire test process according to the test schedule, sending test patterns to and receiving test responses from the corresponding cores through the test bus and the TAM wires.

Suppose that a system S , consisting of n cores C_1, C_2, \dots, C_n , employs the test architecture defined above. In order to test core C_i , a test set TS_i consisting of l_i generated test patterns is transported through the test bus and the dedicated TAM wires to/from core C_i , utilizing a bus bandwidth W_i . The test bus is designed to allow transporting several test sets in parallel but has a bandwidth limit BL ($BL \geq W_i, i = 1, 2, \dots, n$). We assume that continuously applying test patterns belonging to TS_i may cause the temperature of core C_i to go beyond a certain limit TL_i so that the core can be damaged. In order to prevent overheating during tests, as discussed before, we partition a test set into a number of test sub-sequences and introducing a cooling period between two partitioned test sub-sequences, such that no test sub-sequence drives the core temperature higher than the limit and the core temperature is kept within a safe range.

The problem that we address in this paper is to generate a partitioning scheme and a test schedule for system S such that the test application time is minimized while the bus bandwidth constraint is satisfied and the temperatures of all cores during tests remains below the corresponding temperature limits.

4. Overall strategy

We have proposed an approach to solve the formulated problem in two major steps. First, we generate an initial partitioning scheme for every test set by using temperature simulation and the given temperature limits. Second, the test scheduling algorithm explores different test schedules by selecting alternative partitioning schemes, interleaving test sub-sequences, and squeezing them into a two-dimensional space constrained by the test-bus bandwidth.

In order to generate thermal-safe partitioning schemes, we have used a temperature simulator, HotSpot /17/, /25/, /26/, /27/, to simulate instantaneous temperatures of individual cores during tests. HotSpot assumes a circuit packaging configuration widely used in modern IC designs, and it computes a compact thermal model /27/ based on the analysis of three major heat flow paths existing in the assumed packaging configuration /26/, /27/. Given the floorplan of the chip and the power consumption profiles

of the cores, HotSpot calculates the instantaneous temperatures and estimates the steady-state temperatures for each unit. In this paper, we assume that the temperature influences between cores are negligible since the heat transfer in the vertical direction dominates the transferring of dissipated heat, which has been validated by the simulation results with HotSpot /14/, /19/.

When generating the initial thermal-safe partitioning scheme, we have assumed that a test set TS_i is started when the core is at the ambient temperature TM_{amb} . Then we start the temperature simulation, and record the time moment t_{h1} when the temperature of core C_i reaches the given temperature limit TL . Knowing the latest test pattern that has been applied by the time moment t_{h1} , we can easily obtain the length of the first thermal-safe test sub-sequence TS_{i1} that should be partitioned and separated from the test set TS_i . Then the temperature simulation continues while the test process on core C_i has to be stopped until the temperature goes down to a certain degree.

Usually a relatively long time is needed in order to cool down a core to the ambient temperature, as the temperature decreases slowly at a lower temperature level (see the dashed curve in Figure 4). Thus, we let the temperature of core C_i go down only until the slope of the temperature curve reaches a given value k^1 , at time moment t_{c1} . At this moment, we have obtained the duration of the first cooling period $d_{i1} = t_{c1} - t_{h1}$. Restarting the test process from time moment t_{c1} , we repeat this heating-and-cooling procedure throughout the temperature simulation until all test patterns belonging to TS_i are applied. Thus we have generated the initial thermal-safe partitioning scheme, where test set TS_i is partitioned into m test sub-sequences $\{TS_{ij} \mid j = 1, 2, \dots, m\}$ and between every two consecutive test sub-sequences, the duration of the cooling period is $\{d_{ij} \mid j = 1, 2, \dots, m-1\}$, respectively. Figure 4 depicts an example of partitioning a test set into four thermal-safe test sub-sequences with three cooling periods added in between.

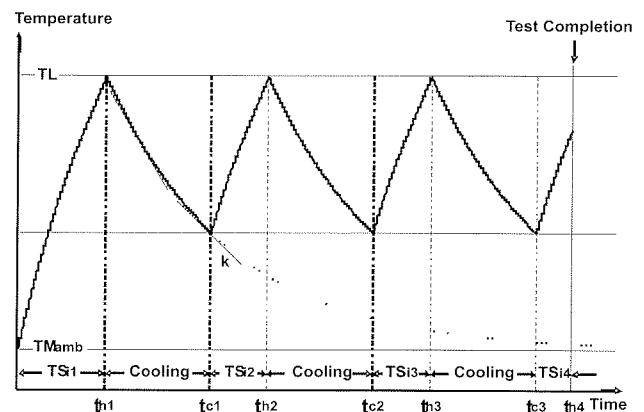


Fig. 4. An example of generating initial partitioning scheme

Once the initial thermal-safe partitioning scheme is obtained, our focus is on how to schedule all the test sub-

¹ The value of k can be experimentally set by the designers.

sequences such that the test application time is minimized under the constraint on the test-bus bandwidth. In this paper, since we consider each test sub-sequence as a rectangle, the problem of generating a test schedule with minimized TAT while satisfying the constraint on the test-bus bandwidth can be formulated as a rectangular packing (RP) problem /28/, /29/, /30/. However, our test scheduling problem is not a classical RP problem, since the length of the sub-sequences, and the cooling periods are not fixed /31/. This makes our problem even more difficult to be solved.

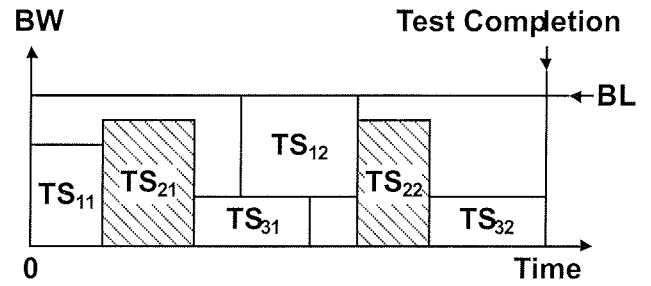
Interleaving test sub-sequences belonging to different test sets can also introduce time overheads /32/, /11/, when the test controller stops one test and switches to another. Therefore, partitioning a test set into more test sub-sequences may lead to a longer test application time, since more time overheads and more cooling periods are introduced into the test schedule. On the other hand, partitioning a test set into more test sub-sequences results in a shorter average length of the individual test sub-sequences, which in principle can be packed in a more compact way and thus lead to shorter test application times. Thus, we need a global optimization algorithm, in which different numbers and lengths of test sub-sequences as well as variant cooling periods are explored, while taking into account the time overheads introduced by test sub-sequence interleaving.

5. Test scheduling heuristic

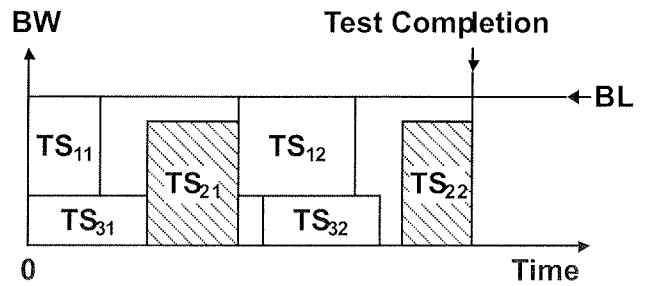
We have proposed a heuristic to do the test scheduling with test set repartitioning and interleaving. Since the order in which the test sets are considered for test scheduling has a large impact on the final test schedule, we construct an iterative algorithm to obtain a good scheduling consideration order (SCO) for all partitioned test sets, and thereafter schedule the test sub-sequences according to the obtained SCO /31/.

Figure 5 shows a simple example illustrating the impact of different scheduling consideration order on the test schedule of three test sets, TS_1 , TS_2 , and TS_3 , each of which is partitioned into two test sub-sequences. Figure 5(a) and Figure 5(b) respectively depicts the test schedule when the test sets are considered for scheduling in the order of $\{TS_1, TS_2, TS_3\}$ and $\{TS_3, TS_2, TS_1\}$. It is obvious that using the second SCO results in a shorter test schedule. Note that in this example the test sets are scheduled to the earliest available time moments.

It should also be noted that the scheduling consideration order refers to the precedence of partitioned test sets to be considered for scheduling. However, when a test set is taken into account for scheduling, we do not schedule all the test sub-sequences of this test set at one time. Instead, we always take the first unscheduled test sub-sequence of the currently considered test set for scheduling, and thereafter take the first unscheduled test sub-sequence of



(a) Test schedule with the SCO = $\{TS_1, TS_2, TS_3\}$



(b) Test schedule with the SCO = $\{TS_3, TS_2, TS_1\}$

Fig. 5. Illustration of how SCO affects test length

the next test set into account. Thus, in this example, the overall scheduling consideration order (OSCO) for all test sub-sequences of all test sets is $\{TS_{11}, TS_{21}, TS_{31}, TS_{12}, TS_{22}, TS_{32}\}$ and $\{TS_{31}, TS_{21}, TS_{11}, TS_{32}, TS_{22}, TS_{12}\}$, for the case of Figure 5(a) and Figure 5(b) respectively. The main concern of not scheduling all test sub-sequences of one test set at one time is to avoid generating low efficient test schedule due to unnecessarily long cooling periods, inappropriate partition length, and inefficient test-set interleaving.

The basic idea of the proposed heuristic is to iteratively construct a queue that finally consists of all partitioned test sets in a particular order. Given a set of test sets $U = \{TS_i \mid i = 1, 2, \dots, n\}$, the heuristic iteratively selects test sets and inserts them into a queue Q . The positions of the test sets in Q represents the order in which the test sets are considered for test scheduling (SCO); the closer to the queue head, the earlier to be considered.

The heuristic starts with an empty queue $Q = \emptyset$. At each iteration step, the objective is to select one test set TS_k from U , and insert it into Q at a certain position POS , such that the $|Q| + 1$ test sets are put in a good order while the precedence between test sets excluding the newly inserted one remains unchanged. The algorithm terminates when all test sets in U have been moved into Q , and thereafter it schedules the partitioned test sets according to the SCO obtained in Q_{best} .

For each iteration step, there are $|U|$ alternative test sets for selection, where $|U|$ is the current number of test sets remaining in U . For each selected test set, there are $|Q| + 1$ alternative positions which the selected test set

can be inserted to, where $|Q|$ is the current number of test sets that have already been inserted into Q in the previous iteration steps. Thus, at one iteration step, there are $|U| \times (|Q| + 1)$ alternative solutions, in which a selected test set is associated with an insertion position in Q .

The proposed algorithm evaluates the obtained scheduling consideration order by the efficiency of the generated partial test schedule; the higher efficiency, the better the SCO. The partial test schedule is generated by a basic scheduling algorithm. Based on the test-schedule efficiency, we explore different solutions and make decisions according to the efficiency of the generated partial test schedules.

We define the efficiency of a test schedule, denoted with η , as follows. Suppose x is the size of the area covered by all scheduled test sub-sequences, and y is the total area size constrained by the bus bandwidth limit and the completion time moment of the test schedule. The efficiency of the test schedule is the value of x / y . The larger value of η represents the better test schedule.

Given a queue Q of test sets, the basic scheduling algorithm takes the first unscheduled test sub-sequence from every test set for scheduling, in a round-robin fashion. More concretely, the strategy of the basic scheduling algorithm is explained as follows. According to the SCO given in Q , the scheduler considers one test set at a time for scheduling. When considering each test set, the scheduler only schedules the first unscheduled test sub-sequence to the earliest available time moment, and thereafter turns to consider the next test set. When one round is finished for all the test sets in Q , the scheduler takes the next round for consideration of scheduling test sub-sequences of all the test sets, in the same SCO. This procedure repeats until all test sub-sequences are scheduled. The detailed description of the heuristic and the basic scheduling algorithm and their pseudo-code can be found at [31].

6. Experimental results

We have done experiments using SoC designs with randomly selected cores from the ISCAS'89 benchmarks. The designs for our experiments have 12 to 78 cores.

The main objective of our experiments is to check how efficient the test schedules generated by our heuristic are. We compare the results of our heuristic with those of other two algorithms, a straightforward algorithm (SF) and the simulated annealing algorithm (SA). All the three algorithms employ flexible partitioning of test sets and arbitrary length of cooling periods.

Further more, all the three algorithms employ the same basic scheduling algorithm used in the inner iteration of the algorithms. The difference between them is therefore how they generate the SCO for all test sets. The straightforward algorithm sorts all test sets decreasingly by the

lengths of the entire test sets with the initial partitioning schemes. According to the obtained SCO, the scheduler chooses each test set and schedules the first unscheduled test sub-sequences to the earliest available time moment, until all test sub-sequences of every test set are scheduled.

In the SA algorithm, the SCO of test sets is generated based on a simulated annealing strategy. When a randomly generated SCO is obtained, the scheduler is invoked to schedule the test sub-sequences according to the current SCO. During iterations, the best SCO that leads to the shortest test schedule is recorded and the algorithm returns this recorded solution when the stopping criterion is met.

The experimental results are listed in Table 1, where column 1 lists the number of cores for the SoC. Column 2 shows the test application time of the generated test schedule when the straightforward algorithm is employed, and column 3 lists the corresponding CPU times to obtain the test schedules. Similarly, columns 4 and 5 are the TAT and CPU times for our heuristic, respectively. Columns 6 and 7 list the TAT and execution times for the simulated annealing algorithm. In columns 7 and 8, the percentage of reduced TAT of the test schedules generated by our heuristic are listed, compared to those generated by the straightforward algorithm and the simulated annealing algorithm, respectively.

Table 1. Comparison of different approaches.

# cores	SF		Our heuristic		SA		TAT gain (%)	
	TAT	CPU Times (s)	TAT	CPU Times (s)	TAT	CPU Times (s)	From SF	From SA
12	1213	0.01	1048	2.74	992	148.31	13.6%	-5.6%
18	1716	0.01	1535	5.41	1513	208.06	10.5%	-1.5%
24	2632	0.01	2318	21.88	2234	229.94	11.9%	-3.8%
30	2274	0.01	1915	32.41	1869	417.08	15.8%	-2.5%
36	3161	0.01	2539	67.52	2494	540.48	19.7%	-1.8%
42	3846	0.01	3334	101.39	3292	631.00	13.3%	-1.3%
48	4328	0.01	3509	151.33	3485	898.77	18.9%	-0.7%
54	4877	0.01	4290	244.36	4051	675.44	12.0%	-5.9%
60	5274	0.01	4692	371.73	4457	2171.73	11.0%	-5.3%
66	5725	0.01	5069	511.88	4917	2321.39	11.5%	-3.1%
72	6538	0.01	5822	720.53	5689	1994.56	11.0%	-2.3%
78	6492	0.01	5769	987.75	5702	3301.45	11.1%	-1.2%
AVG	N/A	N/A	N/A	N/A	N/A	N/A	13.4%	-2.9%

The comparison between our heuristic and the straightforward algorithm aims to show how much TAT can be reduced by the advanced heuristic we have developed. On the other hand, the comparison between our heuristic and the simulated annealing algorithm is to find out how close our generated test schedule is to a solution which is very

close to the optimal one. In order to generate close-to-optimal solutions, the SA algorithm has been run for long optimization times.

It can be seen that, when using our heuristic, the TAT is in average 13.4% shorter than those using the straightforward algorithm. The TAT is in average 2.9% longer than those using the simulated annealing algorithm which however needs much longer execution times.

7. Conclusions

In this paper, we have discussed several issues related to the thermal problem when a SoC is being tested. We have also proposed a heuristic to generate thermal-safe test schedules for SoC in order to minimize the test application time. Based on the initial partitioning scheme generated by a temperature simulation guided procedure, the heuristic utilizes the flexibility of changing the length of test subsequences and the cooling periods between test subsequences, and interleaves them to generate efficient test schedules. Experimental results have shown the efficiency of our heuristic.

8. References

- /1/ Y. Zorian, S. Dey and M. Rodgers. "Test of future system-on-chips," *Proc. ICCAD*, 2000.
- /2/ E. Larsson, and Z. Peng. "An integrated framework for the design and optimization of SoC test solutions". *Journal of Electronic Testing; Theory and Applications (JETTA)*, Vol. 18, No. 4/5, pp. 385-400, 2002.
- /3/ B. T. Murray, and J. P. Hayes. "Testing ICs: Getting to the core of the problem". *IEEE Transactions on Computer*, Vol. 29, pp. 32-39, Nov. 1996.
- /4/ Y. Zorian, E. J. Marinissen, and S. Dey. "Testing embedded core-based system chips". *International Test Conference (ITC)*, 1998, pp. 130-143.
- /5/ Y. Huang, W.-T. Cheng, C.-C. Tsai, N. Mukherjee, O. Samman, Y. Zaidan, and S. M. Reddy. "Resource allocation and test scheduling for concurrent test of core-based SoC design". *Asian Test Symposium (ATS)*, 2001, pp. 265-270.
- /6/ J. Aerts, and E. J. Marinissen, "Scan chain design for test time reduction in core-based ICs", *International Test Conference (ITC)*, 1998, pp. 448-457.
- /7/ V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test access mechanism optimization, test scheduling, and test data volume reduction for System-on-Chip", *IEEE Transactions on Computer*, Vol. 52, No. 12, Dec. 2003.
- /8/ P. Varma, and B. Bhatia, "A structured test re-use methodology for core-based system chips", *International Test Conference (ITC)*, 1998, pp. 294-302.
- /9/ B. Pouya and A. Crouch. "Optimization trade-offs for vector volume and test power". *International Test Conference (ITC)*, 2000, pp. 873-881.
- /10/ R. Chou, K. Saluja, and V. Agrawal. "Scheduling tests for VLSI systems under power constraints". *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5(2):175-184, June 1997.
- /11/ Z. He, Z. Peng, and P. Eles. "Power constrained and defect-probability driven SoC test scheduling with test set partitioning". *Design Automation and Test in Europe Conference (DATE)*, 2006, pp. 291-296.
- /12/ C. Liu, K. Veeraraghavan, and V. Iyengar. "Thermal-aware test scheduling and hot spot temperature minimization for core-based systems". *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, 2005, pp. 552-560.
- /13/ P. Rosinger, B. M. Al-Hashimi, and K. Chakrabarty. "Thermal-safe test scheduling for core-based System-on-Chip integrated circuits". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Vol. 25, No. 11, pp. 2502-2512, Nov. 2006.
- /14/ Z. He, Z. Peng, P. Eles, P. Rosinger, and B. M. Al-Hashimi. "Thermal-aware SoC test scheduling with test set partitioning and interleaving". *International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, 2006, pp. 477-485
- /15/ S. Gunther, F. Binns, D. M. Carmen, and J. C. Hall. "Managing the impact of increasing microprocessor power consumption". *Intel Technology Journal*. 2001.
- /16/ R. Mahajan. "Thermal management of CPUs: A perspective on trends, needs and opportunities". Keynote presentation at the *8th Int'l Workshop on THERMal INvestigations of ICs and Systems (THERMINIC)*. 2002.
- /17/ K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. "Temperature-aware microarchitecture: Modeling and implementation". *ACM Trans. Architecture and Code Optimization (TACO)*. Vol. 1, No. 1, pp. 94-125, 2004.
- /18/ C. Shi and R. Kapur. "How power-aware test improves reliability and yield". *EE Times*, Sept. 15, 2004. <http://www.eetimes.com/showArticle.jhtml?articleID=47208594>.
- /19/ Z. He. *System-on-Chip Test scheduling with Defect-Probability and Temperature Considerations*, Licentiate thesis, Dept. of Computer Science, Linköping University, No. 1313, 2007.
- /20/ S. Gerstendörfer and H. J. Wunderlich. "Minimized power consumption for scan-based BIST," in Proc. IEEE International Test Conference, 1999, pp. 77-84.
- /21/ J. Saxena, K. M. Butler, and L. Whetsel, "An analysis of power reduction techniques in scan testing," in Proc. IEEE International Test Conference, 2001, pp. 670-677.
- /22/ P. Rosinger, B. M. Al-Hashimi, and N. Nicolici, "Power profile manipulation: A new approach for reducing test application time under power constraints," *IEEE Trans. CAD*, vol. 21, no. 10, pp. 1217-1225, 2002.
- /23/ P. Flores, J. Costa, H. Neto, J. Monteiro, and J. Marques-Silva, "Assignment and reordering of incompletely specified pattern sequences targeting minimum power dissipation," in Proc. International Conference on VLSI Design, 1999, pp. 37-41.
- /24/ P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac, "Reducing power consumption during test application by test vector ordering," in Proc. IEEE International Symposium on Circuits and Systems, 1998, pp. 296-299.
- /25/ W. Huang, S. Ghosh, K. Sankaranarayanan, K. Skadron, and M. R. Stan. "HotSpot: Thermal modeling for CMOS VLSI systems." *IEEE Trans. on Component Packaging and Manufacturing Technology*. 2005.
- /26/ K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. "Temperature-aware microarchitecture." *International Symposium on Computer Architecture*, 2003, pp. 2-13.
- /27/ W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy. "Compact thermal modeling for temperature-aware design". *Design Automation Conference (DAC)*, 2004. pp. 878-883.
- /28/ B. S. Baker, E. G. Coffman Jr., and R. L. Rivest. "Orthogonal packings in two dimensions". *SIAM Journal on Computing*, Vol. 9, No. 4, pp.846-855, 1980.

- /29/ H. Dyckhoff, "A typology of cutting and packing problems". *European Journal of Operational Research*, Vol. 44, No. 2, pp. 145-159. 1990
- /30/ N. Lesh, J. Marks, A. McMahon, and M. Mitzenmacher. "Exhaustive approaches to 2D rectangular perfect packings", *Elsevier Information Processing Letters*, Vol. 90, No. 1, pp. 7-14, Apr. 2004.
- /31/ Z. He, Z. Peng and P. Eles, "A heuristic for thermal-safe SoC test scheduling." *International Test Conference (ITC)*, 2007.
- /32/ S. K. Goel, and E. J. Marinissen. "Control-aware test architecture design for modular SoC testing". *European Test Workshop (ETW)*, 2003. pp. 57-62.

Zebo Peng, Zhiyuan He and Petru Eles
Embedded Systems Laboratory
Linköping University, Sweden
zpe@ida.liu.se

Prispelo (Arrived): 15.07.2007 Sprejeto (Accepted): 01.09.2007