Simulation-Driven Thermal-Safe Test Time Minimization for System-on-Chip

Zhiyuan He, Zebo Peng, and Petru Eles
Embedded Systems Laboratory (ESLAB)
Linköping University, Sweden
{zhihe, zebpe, petel}@ida.liu.se

Abstract [1]

Thermal safety has become a major challenge to the testing of systems-on-chip with deep sub-micron technologies. In order to avoid overheating the devices under test while reducing test application times, new techniques are needed. In this paper, we propose a test scheduling technique to minimize the test application time such that the temperatures of individual cores are kept below a given limit. The proposed approach takes into account thermal influences between cores, and thus accurate temperature evolution information of all cores in a system-on-chip is needed for the test scheduling. In order to avoid overheating, we have employed a thermal simulation driven scheduling algorithm, in which instantaneous thermal simulation results are used to guide the partitioning of test sets into test sub-sequences and to determine cooling periods inserted between the partitions. Furthermore, the partitioned test sets for different cores are interleaved such that a cooling period reserved for one core can be utilized for the test-data transportations and test applications for other cores. Experimental results have shown that by using the proposed technique, the test application time is minimized and the temperatures of cores under test are kept below the temperature limit during the entire test process.

1. Introduction and related work

Nanoscale technology has become the mainstream in the design and production of integrated circuits (ICs). In the latest generation of IC designs, the power density has been substantially increased [1], [2]. As a consequence of the elevated power density, high temperature in the chip becomes a critical challenge [3], [4]. In particular, compared to the normal functional mode, testing has been expected to consume more power [5], [6], which leads to an even higher temperature on silicon dies. Therefore, rigid temperature control during test is required in order to prevent possible damages to the circuits under test. Some advanced cooling techniques are proposed to reduce the temperature in the chips, but they substantially increase the overall cost. Other techniques such as lower frequency and reduced speed can partly solve the high temperature problem, while making them inapplicable to at-speed test and leading to longer test application time.

In the case of system-on-chip (SoC) test, the problems of long test time and high temperature become more severe. Due to the high power consumption and high temperature in the latest generation of SoCs, novel techniques are proposed to tackle the test time minimization problem in the new context. In [7], [8], low power test techniques are proposed to reduce the power consumption during tests. Some other works focus on power-constrained test scheduling [9], [10], [11], [12], targeting test time minimization restricted in a fixed power envelope. However, only using the power-aware techniques cannot fully avoid the overheating problem because of the complex thermal phenomenon in modern electronic chips [13].

Thus, thermal-aware test techniques have been proposed in order to solve the overheating problem during SoC test. Rosinger et al. proposed an approach [13] to generate thermal-safe test schedules with minimized test application time (TAT). Information about the neighborhood relationship of the cores under test (CUTs) is used to generate shortest test schedules which also reduce the temperature variances among cores. In [14], Yu et al. proposed a thermal-safe TAM/wrapper co-optimization and test scheduling approach, in which the thermal influences between cores are taken into account and a thermal cost model is improved from [13] to generate more accurate results. Despite obtaining substantial reduction in test time, these approaches make the strong and simplifying assumption that one test can never produce overheating on the CUT. In our previous works [15], [16], a test set partitioning and interleaving technique was proposed to avoid high temperature and minimize the TAT, assuming that continuously applying a single test set may burn the CUT. In these works, it was assumed that the circuit layout and the employed technology are such that the thermal influence between cores can be neglected. However, in many other SoC designs, especially those which have a relatively large contact area between cores, the lateral thermal influences cannot be ignored.

Thus, in this paper, we address the thermal-safe test scheduling issue in the context that continuous application of tests for a core can lead to excessively high temperature and that the lateral thermal influence between cores is not negligible. Due to the temporal and spatial thermal interdependencies [17], [18], coarse grained thermal models cannot solve the problem. Thus, we have employed a fast thermal simulator, ISAC [19], to obtain accurate instantaneous temperature values and have used them to guide the partitioning and interleaving of test sets during the test scheduling. A finite state machine (FSM) model has been developed to control the partitioning and interleaving process, based on which a heuristic has been developed to generate the shortest thermal-safe test schedules.

[1] This work has been partially supported by the Swedish Foundation for Strategic Research (SSF) under the Strategic Integrated Electronic Systems Research (STRINGENT) program.

The rest of this paper is organized as follows. The next section presents the assumed basic test architecture. In Section 3, the motivation for the thermal-safe test scheduling problem is demonstrated. Section 4 gives the problem formulation, and Section 5 illustrates the proposed heuristic for the thermal-safe test scheduling. Experimental results are presented in Section 6 and the paper is concluded in Section 7.

2. Basic test architecture

We have assumed that the tester employed for a SoC test is either an automatic test equipment (ATE) or an embedded tester in the chip. In the tester, a memory is used to store the generated test patterns and a test schedule. A test controller integrated in the tester controls the transportation of test data and the application of test patterns according to the test schedule. A test bus is employed for the test data transportation between the tester and the CUTs, and each core is connected to the test bus through dedicated TAM wires. Through the test bus and TAM wires, the test controller sends test patterns to the destination cores and receives test responses from the cores when the test patterns have been applied.

3. Motivation

The state of the art of SoC test has shown that the large test data volume and the long test application time substantially increase the testing cost. When considering SoC test in a thermal safe context, a very long test process applied to a core may lead to a very high temperature even before the test is completed. This means that the CUT may be damaged if its temperature goes beyond a certain limit and the test is not interrupted in time. Thus, in order to prevent overheating, an individual test has to be stopped when the temperature of the core reaches the temperature limit, denoted with *TL*, and a cooling period is needed before the test can be continued. In this paper, we refer to the cooling as a passive cooling, meaning that the core is not activated and does not consume dynamic power. Thus, by partitioning an individual test set into a number of test sub-sequences and inserting cooling periods between them, we can avoid overheating during the entire test process.

When test set partitioning is employed to avoid overheating, the efficiency of the utilization of the test bus should also be considered for test scheduling. It is obvious that introducing long cooling periods between test sub-sequences of a core can substantially increase the test application time. On the other hand, during the cooling periods of a core, the bandwidth of the test bus previously allocated to this core is not utilized. Thus we can release the bus bandwidth reserved for a core during its cooling periods, and allocate the released bus bandwidth to other cores for their test-data transportations and test applications. In this way, the test sets of different cores are interleaved and thus the TAT can be reduced. Figure 1 gives an example where two partitioned test sets are interleaved such that both the bandwidth limit and temperature limit are satisfied.

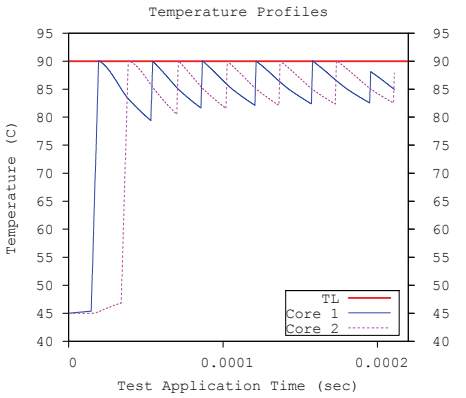

Figure 1. An example of test set partitioning and interleaving

In our previous work [16], it is assumed that lateral heat flows between cores can be neglected. This assumption fits a category of SoCs that have relatively large area size and small thickness of the silicon die. However, when the technology scales, the area size decreases while the die thickness is not reduced in the same order of magnitude. For such a category of SoCs, the lateral heat flow takes a larger portion in the overall thermal flows, and therefore cannot be ignored. Thus, in this paper, we take into account the thermal influences between cores and develop a new test scheduling technique in order to guarantee the thermal safety in this new context.

Figure 2 depicts a result of thermal simulation performed for a SoC design with the die thickness equal to 200 micrometers. The SoC consists of two adjacent cores, both of which have an equal area size. In this experiment, only Core 1 is applied with test patterns. It can be seen that Core 2 is passively heated by Core 1 and the temperature rise is about 19 degrees. This example confirms our concerns that for this category of SoCs, the lateral thermal influences should not be ignored.

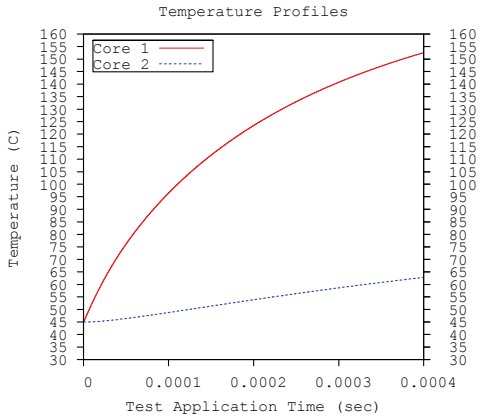

Figure 2. An example illustrating lateral thermal influence

A direct impact due to the thermal influence between cores is that an inactive core at a lower temperature can be passively heated by its neighbors which have a higher temperature. In such cases, the temperature of the inactive core can be elevated (see Figure 2). The temperature elevation effect becomes more significant when the inactive core is passively heated by a larger number of high-temperature neighbors at the same time. Another factor that affects the extent of the temperature elevation is the time duration of tests applied on the neighboring

cores: the longer test time, the higher degree of temperature elevation.

When taking into account the lateral thermal influence and the resulted temperature elevation effect for test scheduling, the spatial distribution of cores and their temperatures, as well as the temporal relations between individual test applications are critically important. They make the thermal-safe test scheduling problem highly complex. In [16], we proposed an approach, denoted with *ALG0*, which determines the initial test-set partitioning schemes according to the thermal simulation results of individual cores, and thereafter generates the test schedule with minimized TAT. However, *ALG0* cannot be directly used to solve the thermal-safe test scheduling problem when lateral thermal influence is taken into account. Figure 3 depicts thermal simulation results for a test schedule generated by *ALG0*. It can be seen that the temperature curves of the CUTs exceed the temperature limit at several points. This example illustrates that *ALG0* no longer guarantees the thermal safety in the new context where the lateral thermal influence becomes significant.

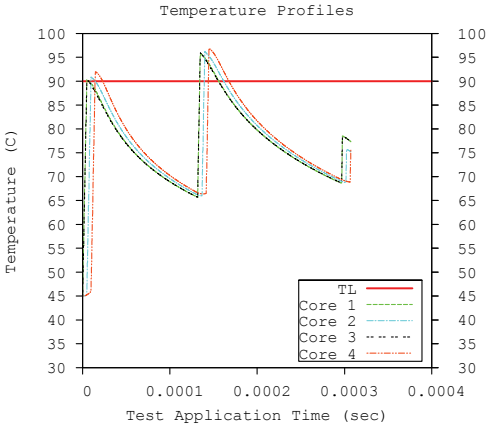

Figure 3. An example showing that ALG0 cannot guarantee thermal safety when lateral thermal influence is significant

In this paper, we aim to minimize test application times by generating efficient test schedules with temperature and bandwidth constraints. We have proposed a thermal-simulation driven test scheduling technique. During the test scheduling, test sets are partitioned and interleaved on-the-fly according to instantaneous thermal simulation results.

As shown in Figure 1, when the temperature of a core reaches the temperature limit, the test for this core is interrupted and a cooling period is started. The temperature of the core decreases until reaching a lower temperature level, and thereafter the test for the core can be resumed. In this paper, such a lower temperature level is called the stop-cooling temperature, denoted with *CL*. The distance between *CL* and *TL* has a large impact on the length of cooling periods and test sub-sequences. Cooling periods are usually started at *TL*, and last until the core temperature deceases to *CL*. Test sub-sequences, except the first one, usually are started from *CL* and stopped at *TL*.

Figure 4 illustrates a scenario where the individual test schedule for one of the cores in a SoC changes when various stop-cooling temperatures are used for test scheduling. When making a comparison between test schedules 1 and 3, we can see that test schedule 1 uses a lower *CL* which leads to longer but fewer test sub-sequences and cooling periods. Test schedule 3 uses a higher *CL* which results in shorter but more test sub-sequences and cooling periods. Both test schedules have a longer TAT than test schedule 2 with a *CL* between those used for test schedules 1 and 3.

The main reason why a higher *CL* may lead to a longer test schedule is the time overhead [20], [12] needed when the test controller stops one test and starts or resumes another. When a higher *CL* is employed, a larger amount of time overhead is more likely to appear, because a larger number of test sub-sequences are to be interleaved with test sets for other cores. On the other hand, a lower *CL* does not necessarily result in a shorter test schedule, though the reduced number of test sub-sequences should lead to less time overhead due to the switchings among different cores. This is because the temperature of a core decreases slower at lower temperature levels and the increased cooling period may not be sufficiently compensated by the benefits from having reduced number of cooling periods and less time overhead. Thus, the different stop-cooling temperatures should be explored together with the test set partitioning and interleaving schemes, in order to obtain efficient test schedules.

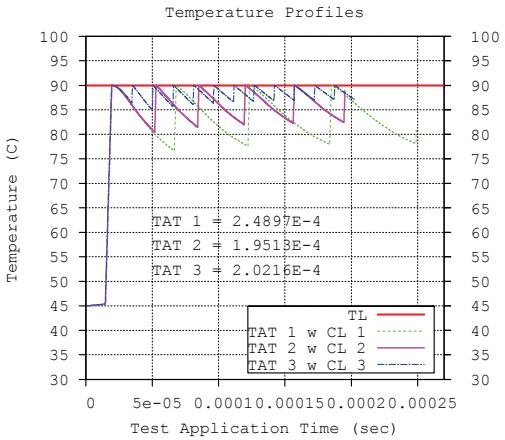

Figure 4. Alternative test schedules w.r.t. various CLs

4. Problem formulation

Suppose that a system-on-chip, denoted with S, consists of n cores, denoted with $C_1, C_2, \ldots, C_n$, respectively, which are placed according to a floorplan, denoted with *FLP*. In order to test core C_i $(1 \le i \le n)$, l_i test patterns are generated, and the test set is denoted with TS_i. The test patterns/responses are transported through the test bus and the dedicated TAM wires to/from core C_i, and the amount of required test-bus bandwidth is denoted with W_i. The test bus is designed to transport test data for different cores in parallel and the bandwidth limit is denoted with BL ($BL \ge W_i$, $i = 1, 2, \ldots, n$). We assume that continuously applying test patterns belonging to TS_i increases the temperature of core C_i, approaching a temperature limit, denoted with *TL*. If the temperature of core C_i goes beyond *TL*, the core is likely to be damaged.

In order to prevent overheating during tests, a test set needs to be partitioned into a number of shorter test sub-sequences and a cooling period needs to be inserted between two partitioned test sub-sequences. The problem that we address in this paper is to generate a test schedule

for a SoC such that the test application time is minimized while the required amount of test-bus bandwidth of the concurrently applied tests is constrained by the bandwidth limit and the temperature of each core is kept below the temperature limit. Figure 5 gives the problem formulation.

Input:
SoC floorplan *FLP*
Set of test set for each core $\{TS_i \mid i = 1, 2, \ldots, n\}$,
Set of required test-bus bandwidth for each test $\{W_i \mid i = 1, 2, \ldots, n\}$,
Test-bus bandwidth limit *BL*,
Temperature limit *TL*

Output:
Test schedule with the minimized test application time (TAT)

Subject to the following two constraints:
1. At any time moment t before the test is completed, total amount of allocated test-bus bandwidth $BW(t)$ is less than or equal to bandwidth limit *BL*, i.e. $\forall t, BW(t) \leq BL$ *where* $BW(t) ::= \Sigma_j BW_j(t)$;
2. At any time moment t before the test is completed, instantaneous temperature $TEM_i(t)$ of every core C_i is less than or equal to temperature limit *TL*, i.e. $\forall t, \forall i, TEM_i(t) \leq TL$.

Figure 5. Problem formulation

5. Heuristic for thermal-safe test scheduling

As mentioned in previous sections, the lateral thermal influence and the corresponding temperature elevation effect make the thermal-safe test scheduling problem highly complex. Thus, we have proposed a simulation driven test scheduling approach, in which instantaneous thermal simulation is employed to guide the test set partitioning and interleaving. For thermal simulation, we use the ISAC system [19]. ISAC takes the floorplan of a chip and the power consumption profiles of cores as inputs, and calculates the temperature values of cores cycle by cycle.

We have developed a finite state machine model to control the test set partitioning and interleaving during the thermal-simulation driven test scheduling process, as illustrated in Figure 6. There are three states for a core, namely *inactive*, *active*, and *finished*, which correspond to the cases that the core is not being tested, the core is being tested, and the test application is completed on the core, respectively. When the test scheduling process starts, we assume that all cores are at the *inactive* state and their temperatures are equal to the ambient temperature. When a core is selected for test and the required test-bus bandwidth is allocated for the test, a flag *start_test* is set to 1 and the state of the core moves from *inactive* to *active*. While test patterns are applied to the core, the temperature of the core, denoted with *TEM*, increases, and the state of the core remains *active* until the temperature reaches temperature limit *TL* or the test is completed. As soon as the test is completed, the state of the core moves from *active* to *finished*. Otherwise, when the core temperature reaches *TL*, the core state moves from active to *inactive* and remains unchanged until the core temperature decreases to stop-cooling temperature *CL*, from which a new round of state transitions between *active* and *inactive* is repeated until the test is completed. The test scheduling process terminates when all cores are at the *finished* state. Figure 7 depicts the thermal simulation result of a test schedule generated by using the FSM model for a SoC with 4 cores.

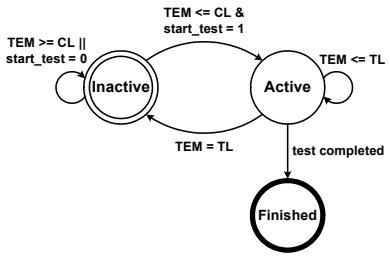

Figure 6. Finite state machine to control temperature

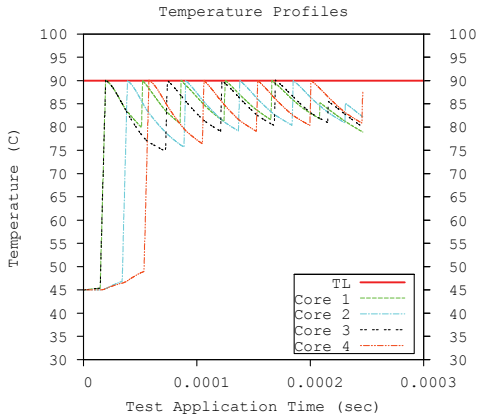

Figure 7. An example of test schedule for a SoC with 4 cores

Using the FSM model to control test set partitioning can generate thermal-safe test schedules. However, the scheduling of test sub-sequences should also take the test-bus bandwidth constraint into account. This is solved by the heuristic given in Figure 8.

```
ALG1. ACTIVATE(Queue of inactive cores ready for test :: Q)
01  if (IsNotEmpty(Q) then
02     Sort Q decreasingly according to the ratio of the remaining
         number of test patterns to the current temperature;
03     while (GetRemainingBandwidth() > 0 & IsNotEmpty(Q)) loop
04        CurrentCore = GetFirstElement(Q);
05        ReqBwd = GetBandwidthRequirement(CurrentCore);
06        if (ReqBwd <= GetRemainingBandwidth()) then
07           Transit the state of CurrentCore to active;
08           ReduceRemainingBandwidth(ReqBwd);
09           Remove(CurrentCore, Q);
10        else
11           break loop;
12        end if-then-else
13     end while
14  end if
```

Figure 8. Pseudo-code of the heuristic that allocates bandwidth to and activates the cores ready to be tested

The heuristic takes a queue of all inactive cores that are ready for test as input and allocates the bus bandwidth to some of the cores and change their states to *active*. The heuristic first sorts the queue decreasingly according to the ratio of the remaining number of test patterns to the current temperature of the core (Line 2). This means that a higher priority is given to a core which has a larger number of remaining test patterns and a lower temperature. Then the heuristic allocates all the currently available bandwidth to the cores according to their priorities in the queue (Lines 3 to 13).

The overall strategy to solve the test time minimization problem is illustrated in Figure 9. The test scheduling algorithm iteratively explores alternative solutions by using different stop-cooling temperatures. Within one

iterative step, the test-set partitioning and interleaving scheme is determined according to the result of the instantaneous thermal simulation with the imposed *TL* and *CL*, and a test schedule is generated by using *ALG1*. We have used a counter, denoted with *CNT*, to count the number of consecutive iteration steps in which no TAT reduction is larger than a small positive number, denoted with *e*. If the TAT of the newly generated test schedule is smaller than the minimal TAT of the best solution obtained through previous iteration steps, the current solution is recorded as the best solution. Furthermore, if the reduction in TAT is larger than *e*, counter *CNT* is reset to 0. In the cases that the current TAT is larger than the minimal TAT or the reduction is smaller than *e*, *CNT* is incremented by 1. This procedure repeats until *CNT* is larger than a given threshold, denoted with *THD*, and thereafter the optimized test schedule is output and the test scheduling process terminates.

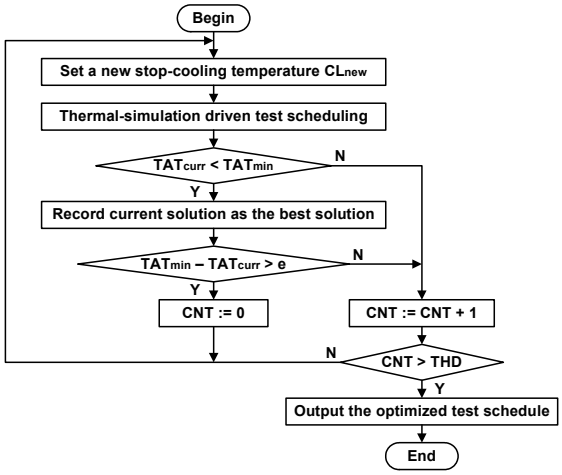

Figure 9. Illustration of the overall solution strategy

During test scheduling, the thermal simulation results are checked at every cycle such that the temperature of core C_i $(1 \le i \le n)$ should be restricted between stop-cooling temperature *CL* and temperature limit *TL*, except for the first test sub-sequences. By using different *CL*, various test partitioning schemes are generated and consequently alternative test schedules are explored. Figure 10 shows experimental results for a SoC with four cores. The TATs with respect to the stop-cooling temperatures used for test scheduling are depicted. The optimal *CL* is 84.065°C and the corresponding minimal TAT is 2.4629×10^{-4} seconds.

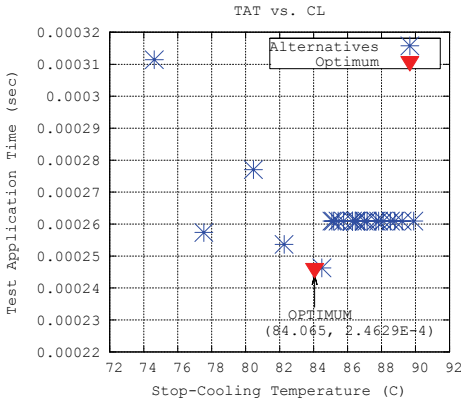

Figure 10. Test application time vs. stop-cooling temperature

6. Experimental results

We have performed experiments for SoC designs consisting of cores randomly selected from the ISCAS'89 benchmarks. The numbers of cores in these designs varies from 4 to 36. The amount of power consumption of a test is obtained through a cycle accurate method proposed in [21] which takes the amount of switching activity as an input and calculates the power consumption in watt. With the obtained power consumption values, the thermal simulator ISAC has been used to calculate instantaneous temperatures at every cycle during test. The imposed temperature limit (*TL*) is 90°C and the assumed frequency of test application is 100MHz. Thermal simulation results have confirmed that the temperatures of all cores under test are below the imposed temperature limit.

We compare our heuristic with a straight-forward approach, in short SFA, which is based on *ALG0*. The basic idea of the SFA is the following. Since *ALG0* ignores lateral thermal influence and directly applying *ALG0* cannot generate thermal safe test schedules, we need to compensate the high temperature by reducing the originally imposed temperature limit, denoted with TL_{orig}, to a lower level. We assume that the reduction from TL_{orig} is corresponding to the heating during test. By running the thermal simulation with generated test schedules, we can obtain the maximum temperature, denoted with *MAXTEM*. The degree of the temperature-limit reduction, denoted with *D*, should equal $MAXTEM - TL_{orig}$. Thereafter, *ALG0* is invoked again with the newly imposed temperature limit, denoted with TL_{new}, and the new test schedule is check by thermal simulation in order to ensure the thermal safety. This procedure is repeated until the first thermal-safe test schedule is generated. However, the thermal-safe test schedule generated in this way can be pessimistically long because the adjusted temperature limit may be lower than needed. In order to reduce the pessimism in terms of the TAT, we use the same procedure to increase the temperature limit until *MAXTEM* is sufficiently close to but still below TL_{orig}. Figure 11 depicts the flowchart of SFA, where *d*, *CNT*, *THD* denotes the given limit for *D*, the number of iteration steps, and the threshold for the total number of iteration steps, respectively. It should be noted that *D* can be either a positive or negative number, corresponding to cases that *MAXTEM* is higher or lower than TL_{orig}, respectively.

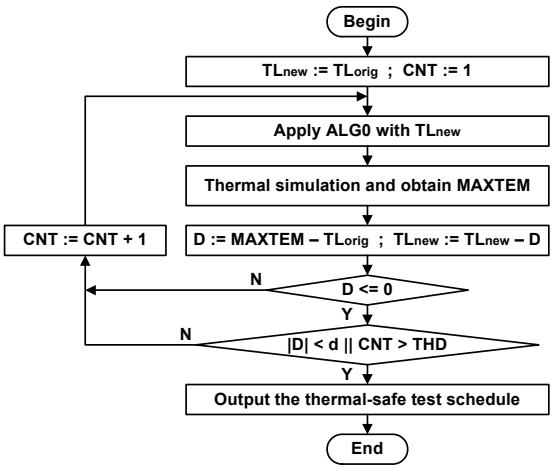

Figure 11. Illustration of the straight-forward approach (SFA)

Experimental results are shown in Table 1. The first column in the table lists the number of cores used in the designs. Columns 2 and 4 show the test application times of the generated test schedules for the corresponding designs, by using the SFA and the proposed heuristic, respectively. Columns 3, 5 list the CPU times needed for executing the corresponding algorithms. Column 6 shows the percentage of TAT reduction by using our heuristic against the SFA. It can be seen that by using our heuristic, the TAT is reduced by about 25% to 61% for different designs. The CPU times of the proposed heuristic are usually shorter than those of the SFA. This is because, in the SFA, each time when *ALG0* is invoked, a thermal simulation is performed for every core in order to generate the initial partitioning schemes according to the new temperature limit.

Table 1. Proposed heuristic vs. SFA

# of Cores	SFA		Proposed Heuristic		TAT Gain
	TAT (s)	CPU Time (s)	TAT (s)	CPU Time (s)	
6	3.9129E-4	1078	2.1013E-4	1118	46.298%
8	3.2827E-4	4122	2.4474E-4	1222	25.446%
12	4.4911E-4	3118	2.3117E-4	1265	48.527%
18	3.6927E-4	7458	2.0832E-4	1193	43.586%
24	4.5970E-4	6681	2.1004E-4	1259	54.309%
30	5.4901E-4	12705	2.2601E-4	1357	58.833%
36	5.7715E-4	11760	2.2360E-4	1400	61.258%

7. Conclusions

In this paper, we have proposed a thermal-safe technique to minimize test application times for systems-on-chip while taking into account thermal influences between cores. The test scheduling employs a thermal simulation to partition and interleave test sets on-the-fly and a heuristic is developed to control the scheduling procedure such that the test application time is minimized and both the temperature limit and test-bus bandwidth limit are not violated. Experimental results have shown the efficiency of the proposed technique.

References

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, Vol. 19, No. 4, pp. 23-29, 1999.

[2] S. Gunther, F. Binns, D. M. Carmen, and J. C. Hall, "Managing the impact of increasing microprocessor power consumption," *Intel Technology J.*,2001.

[3] R. Mahajan, "Thermal management of CPUs: a perspective on trends, needs and opportunities," *Int. Workshop THERMal INvestigations of ICs and Systems*, 2002.

[4] 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*. Vol. 1, No. 1. pp. 94-125, Mar. 2004.

[5] B. Pouya and A. Crouch, "Optimization trade-offs for vector volume and test power," *Int. Test Conf.*, 2000, pp. 873-881.

[6] C. Shi and R. Kapur, "How power-aware test improves reliability and yield," *EE Times*, Sep. 15, 2004. [Online] http://www.eetimes.com/showArticle.jhtml?articleID=47208594.

[7] P. Girard, C. Landrault; S. Pravossoudovitch, and D. Severac, "Reducing power consumption during test application by test vector ordering," *Int. Symp. Circuits and Systems*, 1998, pp. 296-299.

[8] P. Rosinger, B. M. Al-Hashimi, and N. Nicolici, "Scan architecture with mutually exclusive scan segment activation for shift- and capture-power reduction," *IEEE Trans. CAD of ICs and Systems*, Vol. 23, No. 7, pp. 1142-1153, July 2004.

[9] E. Larsson and Z. Peng, "Power-aware test planning in the early system-on-chip design exploration process," *IEEE Trans. Computers*, Vol. 55, No. 2, pp. 227-239, Feb. 2006.

[10] K. Chakrabarty, "Design of system-on-a-chip test access architectures under place-and-route and power constraints," *IEEE/ACM Design Automation Conf.*, 2000, pp. 4332-437.

[11] R. Chou, K. Saluja, and V. Agrawal, "Scheduling tests for VLSI systems under power constraints," *IEEE Trans. VLSI Systems*, 5(2):175-184, June 1997.

[12] 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 Conf.*, 2006, pp. 291-296.

[13] P. Rosinger, B. M. Al-Hashimi, and K. Chakrabarty, "Thermal-safe test scheduling for core-based system-on-chip integrated circuits," *IEEE Trans. CAD of ICs and Systems*, Vol. 25, No. 11, pp. 2502-2512, Nov. 2006.

[14] T. Yu, T. Yoneda, K. Chakrabarty, and H. Fujiwara, "Thermal-safe test access mechanism and wrapper co-optimization for system-on-chip," *IEEE Asian Test Symp.*, 2007, pp. 187-192.

[15] Z. He, Z. Peng, P. Eles, P. Rosinger, and B. M. Al-Hashimi, "Thermal-aware SoC test scheduling with test set partitioning and interleaving," *J. Electronic Testing; Theory and Applications*, Vol. 24, No. 1-3, pp. 247-257, June 2008.

[16] Z. He, Z. Peng, and P. Eles, "A Heuristic for Thermal-Safe SoC Test Scheduling," *IEEE Int. Test Conf.*, 2007, pp. 1-10.

[17] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," *Int. Symp. Computer Architecture*, 2003, pp. 2-13.

[18] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," *IEEE/ACM Design Automation Conf.*, 2004. pp. 878-883.

[19] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, "ISAC: Integrated Space and Time Adaptive Chip-Package Thermal Analysis," *IEEE Trans. CAD of ICs and Systems*. Vol. 26, No. 1, pp. 86-99, Jan. 2007.

[20] S. K. Goel and E. J. Marinissen, "Control-aware test architecture design for modular SoC testing," *European Test Workshop*, 2003. pp. 57-62.

[21] S. Samii, E. Larsson, K. Chakrabarty, and Z. Peng, "Cycle-accurate test power modeling and its application to SoC test scheduling," *IEEE Int. Test Conf.*, 2006, pp. 1-10.