Temperature-Aware SoC Test Scheduling Considering Inter-Chip Process Variation

Nima Aghaee, Zhiyuan He, Zebo Peng, and Petru Eles
Embedded Systems Laboratory (ESLAB), Linkoping University, Sweden
{g-nimgh, zhihe, zebpe, petel}@ida.liu.se

***Abstract*—Systems on Chip implemented with deep submicron technologies suffer from two undesirable effects, high power density, thus high temperature, and high process variation, which must be addressed in the test process. This paper presents two temperature-aware scheduling approaches to maximize the test throughput in the presence of inter-chip process variation. The first approach, an off-line technique, improves the test throughput by extending the traditional scheduling method. The second approach, a hybrid one, improves further the test throughput with a chip classification scheme at test time based on the reading of a temperature sensor. Experimental results have demonstrated the efficiency of the proposed methods.**

I. Introduction

Deep submicron integration comes with high power density and Process Variation (PV) [1]. The power consumption for a System on Chip (SoC) is substantially higher during test compared with normal operation mode [2]. Very high power density at test time will lead to very high temperature and eventually overheating the chip under test. Therefore, the chip temperature has to be taken into account while designing the test process [3, 4]. Traditionally, temperature-aware SoC test scheduling is usually done off-line with the help of thermal simulation in order to minimize the Test Application Time (TAT) while avoid overheating [5-7].

The difference between the expected temperature, which is estimated by simulation, and the actual temperature has various sources including errors induced by the PV. In order to avoid overheating the chips because of these errors, traditional design methodologies use a Safety Margin (SM). The SM is usually defined based on the smallest error probability which is considered as significant. When the expected overall error is small, the SM will be small. With the traditional technologies, the overall impact of the SM on test scheduling objective is negligible and, therefore, this aspect is usually not addressed explicitly [5-7]. But, with the deep submicron technologies, the Temperature Error induced by Process Variation (TEPV) is not negligible anymore. TEPV is a component of the overall temperature error and its increase results in larger SMs and consequently a considerable negative impact on the test objectives.

In this paper we consider the issue of inter-chip variations. This means that we account for the fact that different chips will produce different temperature errors relative to the expected temperature. Intra-chip variations are not considered here, which means that we assume that the variation of the temperature error over the same chip is negligible.

Traditionally, test scheduling aims at the minimization of the Test Application Time (TAT) [7], which does not take into account the problem of chips that are overheated and may be damaged during test. In this paper we introduce the concept of Test Throughput (TT) as the design objective, which addresses the problem of damaged chips during test. When a chip's temperature exceeds the highest safe temperature, it may fail the test or it may be damaged. In this paper, we consider the overheated chip as damaged. The quantity of the damaged chips under test due to overheating is described by the number of damaged chips per number of chips entering the test facility and is represented by the Test Damage Probability (TDP). We define TT as the number of chips passed through the test facility without being damaged per time unit. The test scheduling objective in this paper is to maximize TT which means to minimize TAT and TDP, simultaneously. The TT, in chips per second, is defined as follows:

$$TT = \frac{1-TDP}{TAT}. \tag{1}$$

In this paper we will show that in the presence of inter-chip PV, a traditional test scheduling methodology will result in a small TT. Thus, the traditional method is extended to an off-line method in order to increase TT. The off-line method is vulnerable to TEPV since the TEPV value is not known a priori and therefore the temperatures that are simulated off-line can be very different from the actual temperatures of the chip. However, because of the accessibility of the actual temperatures during the test, an on-line scheduling alternative seems promising at the beginning. But, there are serious drawbacks for on-line test scheduling, mainly due to the implementation overhead. As a compromise, a hybrid approach is proposed to take advantage of both off-line and on-line methods. The basic concept of the hybrid method is to prepare a number of different test schedules in an off-line phase for different TEPV conditions. In the on-line phase, the test schedule corresponding to the measured TEPV is applied.

The rest of the paper is organized as follows. Section II discusses the traditional test scheduling approach. In section III a method to reduce the problem size and the computational effort is suggested. An off-line variation-aware test scheduling approach is introduced and is evaluated in section IV. The hybrid test scheduling approach is proposed in section V. The experimental results are given in section VI. Conclusions are presented in section VII.

II. Background and Motivation

Traditionally, a thermal-safe test schedule is usually generated to minimize the TAT by taking into account a SM. The SM together with the highest safe temperature of the chip are used to determine the temperature limit used to guide an off-line SoC test scheduling algorithm, such as the one proposed in [7]. For example, if the highest safe temperature for the chip is 93^{o}C, and the SM is 3^{o}C, then the temperature limit for the scheduler will be 90^{o}C. The SM is usually determined based on the error probability statistics. For example, assuming a normal distribution for the overall temperature errors, the SM is determined as a multiple of the standard deviation to cover a certain percentage of the chips.

The well-known 68-95-99.7 rule which describes that 68, 95, and 99.7 percent of the chips will be located within 1, 2, and 3 standard deviations around the mean [8] is used for the normal distribution. Since the negative errors cause no harm, 84, 97.5, and 99.85 of the chips are safe, respectively. For example, assuming that 0.15 is an insignificant percentage of the chips (tolerable damage), the SM will be 3 times the standard deviation. Assuming that the temperature error distribution is $\mathcal{N}(0,1)$, the SM is 3^{o}C. Then the temperature limit of the off-line scheduler will be set 3^{o}C lower than the highest safe temperature for the chip.

This research is partially supported by the Swedish Research Council, under contract 621-2007-6587.

In the presence of large PV, the TEPV component in the overall temperature error is enlarged. In order to address the TEPV explicitly, the overall error is divided into two components, the TEPV and the residual error. The residual error is expected to be very similar to the overall error in the absence of large variations, and therefore it is handled like discussed above by assuming a SM. The methods that are discussed in the rest of this paper address the TEPV component, while the residual error has its own SM. A traditional method handles the enlarged TEPV component by following the same method used for residual temperature errors. Following the above example and assuming that TEPV distribution is $\mathcal{N}(0,36)$, the SM required for TEPV will be equal to 18°C (temperature limit = 93 – 3 – 18 = 72°C). Such a large SM will drastically increase the TAT. This negative effect is demonstrated in the following experiment.

To do the experiment, TAT and TDP must be computed for the given SM and TEPV statistics. In order to obtain the TAT, a thermal-safe scheduling algorithm is required to schedule a set of test sequences for a certain SoC. The scheduling algorithm which we use is based on the heuristic proposed in [7]. The thermal-simulator which is used to simulate the core temperatures is ISAC [9]. The samples are named SoC4, SoC8, and SoC16, having 4, 8, and 16 cores, respectively. The cores are selected from the ISCAS'89 benchmarks. The test clock frequency is 100 MHz and the available test bus widths are 20, 23, and 27 bits, respectively. The power values are computed using a cycle accurate method introduced in [10]. The test patterns are generated using Turbo Tester [11]. The thermal-safe test scheduling is guided by core temperatures at each test cycle in addition to the test bus utilization.

For the experiments, consider the above example and assume that the significant TEPV errors range from -20°C to +20°C having a resolution equal to 0.1°C. Therefore, the size of the data set, M, is 400 TEPV values. We will evaluate the traditional method for the following cases:

1. There is no error at all (no SM, denoted by "Ideal");
2. There is error but no TEPV (temperature error ~ $\mathcal{N}(0,1)$), thus, $SM_R = 3\times1 = 3^oC$, denoted by "No PV"); and
3. There is inter-chip TEPV (TEPV ~ $\mathcal{N}(0,36)$) thus, the additional $SM_{TEPV} = 3\times6 = 18^oC$, denoted by "PV").

It is assumed that the TDP is 0, since the SM is set in a thermal-safe manner. The TATs, TTs, and TT change percentages relative to the "Ideal" case (ΔTT%), are given in Table I. Looking into these percentages reveals that the degradation of the TT is unacceptably high. For example, for the SoC16, the TT degradation is 84.64% when there is a large inter-chip TEPV. Therefore, it is necessary to develop variation-aware test scheduling methods directly targeting the optimization of the TT.

III. TEPV Statistics and Error Clustering

As opposed to the traditional method, we proposed to explore different SMs to find the largest TT. In order to compute the TT values, the TEPV distribution has to be known in advance. The statistics that show the probability of the TEPV is a collection of TEPV values and their corresponding probabilities. The range and the resolution of the given statistics determine the size of the TEPV statistics data set (M). M is large and the TAT optimization time is long. Therefore, it is not feasible to compute the TAT for all possible SMs. For example, assuming M = 400 and an average TAT optimization time of 1000s, the total run time will be 4.5 days.

It is only possible to compute the exact TAT and TT for a limited number of SMs. To determine these points, the TEPV is clustered into a number of *error clusters* (L) and the exact TAT is computed on the *error cluster* borders. A straight forward uniform clustering will result in an accurate solution at the expense of a considerable computational effort when L is large, and it will result in a fast solution at the expense of a poor accuracy when L is small. In order to have both accurate and fast solution, a non-uniform clustering approach is used.

TABLE I. THE EVALUATION OF THE TRADITIONAL METHOD

Error Mode		Ideal	No PV	PV
Upper Bound Temperature (°C)		93	90	72
SoC4	***TAT(us)***	265	423	1435
	TT(×10 Chip/s)	377	236	70
	ΔTT%	0	-37.33	-81.51
SoC8	***TAT(us)***	297	453	1835
	TT(×10 Chip/s)	336	221	55
	ΔTT%	0	-34.39	-83.80
SoC16	***TAT(us)***	357	502	2327
	TT(×10 Chip/s)	280	199	43
	ΔTT%	0	-28.80	-84.64

The non-uniform clustering produces more clusters in the regions which accommodate a higher number of chips or in the regions which are more likely to accommodate a final solution. The number of chips is characterized by the error probability while the likeliness of a solution being located in a certain region is given by the TT. Since initially the only available data is the error probability, it is used as an initialization heuristic. The initial *error clustering* is followed by a refinement step which takes the now available TT into account. Based on the available computational power and time, the initial number of *error clusters* (Q) is decided. The initialization heuristic is applied in order to cluster the M error values into Q equally probable initial *error clusters*. Then the TAT is computed for the borders of the initial Q clusters. Having the error probabilities and TATs, the *error clustering* is refined by breaking down some of the initial *error clusters*, resulting in L total *error clusters*.

For the set of experiments given in this paper, Q is 12. The resulted initial cluster borders and corresponding TATs are given in Table II (columns 1 to 4). The refinement, results in a number of new *error clusters* which their borders and corresponding TATs are also given in Table II (columns 5 to 10). The L is equal to 26, 29, and 28 for SoC4, SoC8, and SoC16, respectively. These *error clusters* and TATs are used for the experiments in the rest of this paper.

IV. Off-line Approach

In order to optimize the TT, instead of introducing a SM beyond which the number of damaged chips is negligible, one has to target the actual parameters TAT and TDP on which the TT depends. This will lead us to a tradeoff between the schedule length and the percentage of the damaged chips. What has to be done is to select that particular SM which produces the best balance of TAT and TDP such that the TT is optimized. Selecting a small SM will result in a short TAT, but a large number of chips will be damaged by overheating. A larger SM will result in a long TAT, but will reduce the damage. Given a certain SM we obtain the corresponding TAT using the test scheduling algorithm introduced in [7]. The corresponding TDP is calculated by summing the error probabilities from the SM to the positive TEPV extreme. Our off-line approach computes TT for all TEPV values and selects the largest TT and its corresponding SM.

The designer must find the optimum SM in order to obtain the maximum TT. In practice TAT is computed for a limited number of TEPV values (The *error cluster* borders.), because of its relatively long execution time. But TDP values could be obtained for all TEPV values using the given TEPV statistics. The TAT values are interpolated and the TT is computed for all TEPV

TABLE II. THE ERROR CLUSTER BORDERS AND TATS

EB	Initial			Refined					
	SoC4	*SoC8*	*SoC16*	*SoC4*		*SoC8*		*SoC16*	
	TAT	*TAT*	*TAT*	*EB*	*TAT*	*EB*	*TAT*	*EB*	*TAT*
-20	125	222	265	-2.0	273	-3.4	294	-3.4	356
-8.4	235	261	338	-0.8	282	-2.0	306	-0.7	481
-5.9	247	275	346	0.6	433	-0.8	440	0.6	510
-4.2	257	287	354	3.3	480	0.6	464	1.9	530
-2.7	267	300	359	4.6	509	1.9	488	3.3	561
-1.4	277	312	470	5.2	524	3.3	517	4.6	590
-0.1	422	452	500	6.7	565	4.6	547	5.2	605
+1.2	443	475	518	7.6	596	5.2	563	6.5	643
+2.5	466	500	542	9.4	668	6.5	601	7.1	663
+4.0	494	532	576	10.5	723	7.1	621	7.8	687
+5.8	538	579	621	11.8	799	7.8	647	9.2	739
+8.4	626	671	710	13.3	905	9.3	711	10.1	780
+20	2407	3098	3873	15.2	1090	10.3	764	11.3	848
--	--	--	--	17.4	1324	11.5	840	12.8	979
--	--	--	--	--	--	13.0	961	14.7	1248
--	--	--	--	--	--	14.9	1173	17.1	2012
--	--	--	--	--	--	17.2	1593	--	--

EB is Error Cluster Border TEPV is in (ºC) TAT is in (us)

values. To find the optimum SM value all TEPV values are explored.

To evaluate the off-line approach, the same experimental settings as introduced in section II are used here. The values for TAT, TDP, and TT are calculated for all SoCs, and the values for SoC16 are presented in Fig.1. The TT and the CPU times for the solutions are given in Table III. The average TT improvement achieved by the off-line approach with respect to the traditional method is 170%. The optimum SM for SoC16, as could be seen in Fig.1, is equal to 5.9^{o}C which results in a TT equal to 1351.5, TDP equal to 16.04%, and TAT equal to 0.621ms. The price of the damaged chips has to be considered separately and if the chips are expensive, a larger SM should be chosen to reduce the TDP. However, the obtained optimal TTs are more than twice the TTs obtained with the traditional method (See Table III); and there is still room for improvement. Incorporating a temperature sensing phase into a hybrid test scheduling approach in order to increase the TT even more is of our interest and is addressed in the next section.

V. HYBRID SCHEDULING APPROACH

The off-line approach applies one single schedule to all chips. However, each chip is characterized by its own TEPV. Considering that specific TEPV, a customized test schedule could be generated for that particular chip. Having such customized schedules for each value of TEPV, and applying them to the corresponding chip would, of course, produce the best possible global TT. Unfortunately, this is not possible for two reasons. First, generating a potentially extremely large number of schedules is not possible due to the excessively long optimization time. Second, there are a limited number of test schedules which can be stored in the ATE memory.

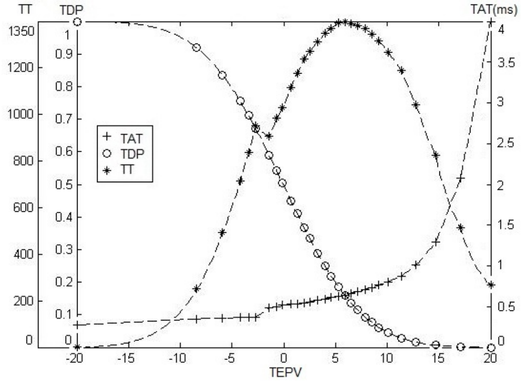

Figure 1. TAT, TDP, and TT versus TEPV for SoC16.

In order to solve the previous two complexity issues, we do not generate schedules for each individual possible TEPV but we produce one single schedule for a range of TEPV values. All the chips with their TEPV value within such a range will be tested using that particular test schedule. We call such a TEPV value range, a *chip cluster*. The schedule corresponding to a *chip cluster* is obtained using the off-line approach presented in section IV. At test time, after application of a warm up sequence, the actual temperature is read using a sensor. As the expected temperature is known in advance, the error value will be known and the chip will be classified into one of the *chip clusters*. After this, the ATE will load the corresponding test schedule and will perform the test accordingly. The number of *chip clusters* and schedules is decided depending on the ATE capabilities and the available optimization time.

Now the problem is reduced to finding the proper *chip clusters* while their number (N) is given. The search space is the set of the possible combinations of TEPV values, as *chip cluster* borders. Each solution has its corresponding expected TT as the objective of the optimization. This solution space is explored by a search algorithm to find the optimum *chip clustering* which results in the maximized TT. The TAT for the i-th error point, TAT(i), is obtained by running the scheduler or by interpolation. The *error cluster* borders are represented by EB(j). The *chip cluster* borders are represented by CB(k).

$$
\begin{aligned}
TAT &= \{TAT(i), i = 0,1, \dots, M-1\}, \\
EB &= \{EB(j), j = 0,1, \dots, L\}, \\
CB &= \{CB(k), k = 0,1, \dots, N\}.
\end{aligned}
\tag{2}
$$

The Computed Test Application Times (CTAT), by running the scheduler, are used to estimate the unknown values of TAT. The CTAT also includes the two extreme points.

$$CTAT = \{CTAT(l), l = 0,1, \dots, L\} \tag{3}$$

$$TAT(i) = \begin{cases} CTAT(j); when\ i = EB(j) \\ IP(i,j);\ where\ EB(j-1) < i < EB(j) \end{cases}. \tag{4}$$

The operator IP(i,j) gives the interpolated value for point *i* which is located in the j-th *error cluster*. The off-line approach gives the optimum Safety Margins (SM) for each *chip clusters* (5). The *Chip Cluster* Probability (CP) is the sum of the TEPV Probabilities (P) within a certain *chip cluster* (6).

$$SM = \{SM(k), k = 0,1, \dots, N-1\} \tag{5}$$

$$CP(k) = \sum_{i=CB(k)+1}^{CB(k+1)} P(i)\ , \kappa = 0,1, \dots, N\text{–}1 \tag{6}$$

Since for testing each individual chip, one of the pre-calculated schedules will be used corresponding to the chip cluster which a certain chip belongs to, there is no longer one single TAT for all chips. Instead of the TAT, the Expected Test Application Time (ETAT) is used to compute the TT. The ETAT is the sum of the optimum TAT within the *chip clusters* (found by the off-line approach) scaled by the *chip cluster* probabilities, over all chip clusters. The ETAT can be written as follows:

TABLE III. TT AND CPU TIME FOR DIFFERENT SCHEDULING METHODS

Design	Test Size (MB)	Traditional		Off-line		Hybrid (N=2)		Hybrid (N=3)		Hybrid (N=4)		Hybrid (N=5)	
		TT	*CPU*	*TT*	*CPU*	*TT*	*CPU*	*TT*	*CPU*	*TT*	*CPU*	*TT*	*CPU*
SoC4	10.72	70	0:53	159	5:59	196	5:59	213	5:59	220	6:01	224	8:59
SoC8	21.43	55	1:11	145	7:24	178	7:24	193	7:24	199	7:26	203	10:40
SoC16	42.86	43	1:32	135	8:38	157	8:38	171	8:38	179	8:41	182	11:48
Average TT relative to the traditional				170%		224%		253%		266%		272%	

TT is in (×10 Chip/s) CPU is in (Hour : Minute) format

$$ETAT = \sum_k (CP(k) \times TAT(SM(k)))\,, k = 0,\ldots, N-1. \qquad (7)$$

Similar to the concept of expected TAT, the Expected Test Damage Probability (ETDP) is the sum of the damage probabilities of clusters over all *chip clusters*. The damage probability of a single *chip cluster* is obtained by summing the TEPV probabilities from the corresponding optimum SM to the right side *chip cluster* border, as follows:

$$ETDP = \sum_k \sum_{i=SM(k)}^{CB(k+1)} P(i)\,, k = 0,\ldots, N-1. \qquad (8)$$

The TT calculation follows Equation (1), however, the expected values, ETAT and ETDP, are used instead of the single values, TAT and TDP, respectively. The search algorithm finds the optimum N-1 unknown *chip cluster* borders in addition to N optimum SMs. The proposed hybrid scheduling method could be summarized as follows:

1. Do the Error clustering based on Section III and compute CTAT on *error cluster* borders.
2. Compute the interpolated TAT values for all TEPV values.
3. Consider a certain chip clustering alternative. For each *chip cluster* apply the off-line approach introduced in section IV. Calculate the TT corresponding to the current chip clustering alternative.
4. Repeat step 3 for different alternative chip clustering and chose the solution with the best TT.

Once a *chip clustering* has been produced as result of the exploration outlined above, the representative schedule for each *chip cluster* is generated according to the off-line approach (see Section IV). These schedules, one for each *chip cluster*, will be stored in the ATE. At test time, after the chip has been classified based on the sensor reading, as discussed earlier, the corresponding test schedule is applied.

VI. Experimental Results

We have done experiments using the settings introduced in section II and considering the *error cluster* borders and the CTATs given in Table II. The resulted TT for the traditional approach, the off-line approach, and for the hybrid approach having 2, 3, 4, and 5 *chip clusters* are given in Table III for all sample SoCs. The *chip cluster* borders and optimum SMs are given in Table IV. The experiments are done for different number of *chip clusters* since the number of *chip clusters* affects the TT and the CPU time of the hybrid approach. The comparison of the different approaches, given in Table III, reveals that the traditional method will result in a very low TT but needs a very short CPU time. The off-line scheduling method has higher TT and considerably higher CPU time. The hybrid approach has higher TT than the off-line approach, while the relative increase in CPU time is small.

The average distance between the solutions points, given in Table IV, and the nearest suggested non-uniform *error cluster* border is 0.115 while the average distance of these points to the nearest uniform *error cluster* border is 0.283 which is more than twice. Therefore, the suggested non-uniform method has improved the *error clustering*. The TT and the CPU times for the hybrid method are shown in Fig. 2. The increase in the TT reduces with higher number of chip clusters. This implies that a good improvement is achieved even by having a minimal number of clusters. The CPU time starts growing fast, for higher number of the chip clusters, and this is another reason to limit the number of *chip clusters*, in this experiment.

VII. Conclusion

Process variation creates a challenging situation for SoC test scheduling as it reduces the test throughput and it increases the damage to the chips. We have proposed two approaches for addressing this problem in the context of inter-chip variations. Our off-line approach improves the test throughput without requiring any measurements using temperature sensors. The test throughput can be further improved by the proposed hybrid approach which requires a chip classification using a temperature sensor. Experiments confirm the effectiveness of the proposed approaches.

TABLE IV. Chip Cluster Borders and Safety Margines

Number of Chip Clusters		***2***		***3***			***4***				***5***				
Chip Cluster Borders	***SoC4***	-0.8		-0.8	3.7		-0.8	3.3	7.0		-0.8	2.7	5.9	9.0	
	SoC8	-1.4		-1.4	4.9		-1.4	2.8	6.5		-1.4	2.0	5.4	8.7	
	SoC16	-2.7		-2.7	3.6		-2.7	2.0	6.3		-2.7	1.3	4.1	6.8	
Optimum Safety Margins	***SoC4***	-0.8	8.3	-0.8	3.7	9.7	-0.8	3.3	7.0	11.9	-0.8	2.7	5.9	9.0	13.3
	SoC8	-1.4	7.9	-1.4	4.9	10.4	-1.4	2.8	6.5	11.5	-1.4	2.0	5.4	8.7	13.0
	SoC16	-2.7	7.9	-2.7	3.6	10.2	-2.7	2.0	6.3	11.4	-2.7	1.3	4.1	6.8	11.4

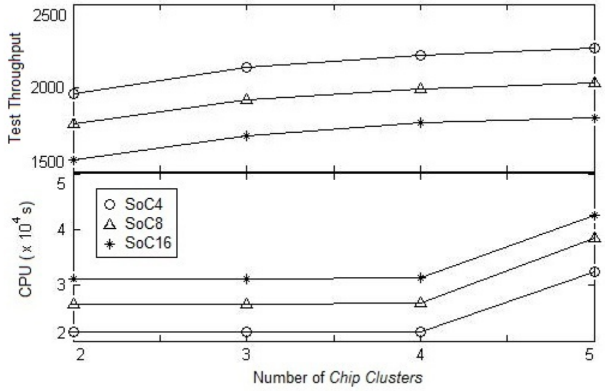

Figure 2. Test Throughput and CPU time for hybrid approach.

References

[1] K.-T. Cheng, S. Dey, M. Rodgers, and K. Roy, “Test challenges for deep sub-micron technologies,” DAC 2000, pp. 142–149.

[2] Y. A. Zorian. “Distributed BIST Control Scheme for Complex VLSI Devices.” VLSI Test Symposium 1993, pp. 4-9.

[3] T. E. Yu, T. Yoneda, K. Chakrabarty, and H. Fujiwara. “Test infrastructure design for core-based system-on-chip under cycle-accurate thermal constraints.” ASP-DAC 2009, pp. 793-798.

[4] U. Chandran and D. Zhao. “Thermal Driven Test Access Routing in Hyper-interconnected Three-Dimensional System-on-Chip.” IEEE DFT 2009, pp. 410 - 418.

[5] 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, pp. 247-257, Vol. 24, No.1-3, 2008.

[6] C. Yao, K. K. Saluja, P. Ramanathan, “Partition Based SoC Test Scheduling with Thermal and Power Constraints under Deep Submicron Technologies,” ATS 2009, pp. 281-286.

[7] Z. He, Z. Peng, P. Eles, "Simulation-Driven Thermal-Safe Test Time Minimization for System-on-Chip," ATS 2008, pp. 283-288.

[8] R. Larson and E. Farber, Elementary Statistics: Picturing the World, 2nd ed., Prentice Hall College Div., 2004, PP.76.

[9] 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., Vol. 26, No. 1, pp. 86-99, 2007.

[10] S. Samii, M. Selkala, E. Larsson, K. Chakrabarty, Z. Peng. "Cycle-Accurate Test Power Modeling and its Application to SoC Test Architecture Design and Scheduling", IEEE Trans. CAD., Vol. 27, No. 5, pp. 973-977, 2008.

[11] G. Jervan, A. Markus, P. Paomets, J. Raik, R.Ubar. “Turbo Tester: A CAD System for Teaching Digital Test,” Microelectronics Education., pp.287-290, 1998.