PEAKASO: Peak-Temperature Aware Scan-Vector Optimization

Minsik Cho and David Z. Pan

Dept. of ECE, The University of Texas at Austin, Austin, TX 78712

thyeros@cerc.utexas.edu, dpan@ece.utexas.edu

Abstract— In this paper, an algorithm for scan vector ordering, PEAKASO, is proposed to minimize the peak temperature during scan testing. Given a circuit with scan and the scan vectors, hotspot is predicted by window-based power analysis. The peak temperature on the hotspot is minimized by global scan vector ordering which expedites heat dissipation to ambient air through large thermal gradient. Further peak temperature reduction is achieved by local scan vector reordering based on overheat precompensation. As an output, PEAKASO provides a scan vector order with lower peak temperature. Note that the scan vectors themselves are not changed at all (only the order is changed), and thus there is no impact on fault coverage and no design overhead. Experimental results on benchmark circuits show that 4.3 - 9.7% peak temperature reduction can be achieved, compared with a scan vector order that is optimized only for average power consumption. Such temperature reduction can be very significant in terms of test time reduction (by 40-60%) under the same peak temperature constraint.

I. INTRODUCTION

Due to substantially higher switching activity of a circuit under test (CUT), power consumption during test can be significantly higher than that during functional mode, e.g., more than 2x higher average power consumption [1] and 30x higher peak power consumption [2]. The power issue becomes more acute with scan chain architectures as each scan vector generates a large number of scan register shifting operations with high circuit activity. Such excessive power consumption not only increases overall chip temperature, but also creates localized overheating, so called *hotspot*. Peak temperature incurred on such hotspot may cause permanent damage to silicon, high cooling cost and reliability failure [3]–[6]. As a result, peak temperature becomes a bottleneck for fast and safe scan based testing.

To address the above power-temperature issue in scan testing, there has been a significant amount of work on power optimization for scan chain, which can be classified into three techniques. First, scan cell reordering is proposed in [7]–[10] to reduce average and peak power consumption. By changing the routing order of scan cells, the number of shifting operations can be minimized at a cost of wire length. Second, scan vector modification is researched in [11] [12] to minimize power consumption. The unspecified values (don't cares) in a scan vector can be filled in a way that the number of shifting operations can be reduced. Last, scan vector ordering is an effective method to reduce both average and peak power consumption [13]–[16].

Although minimizing power consumption can reduce the temperature of CUT, it does not necessarily provide the best solution for peak temperature minimization due to two reasons. First, peak temperature is a *localized event* by nature. Second, peak temperature depends on both *heat generation* from power consumption and *heat dissipation* to ambient air.

In this paper, we propose a scan vector optimization algorithm, PEAKASO to minimize peak temperature of a CUT for scan chains. Essentially, PEAKASO properly orders scan vectors to **i)** maximize the effectiveness of heat dissipation on hotspot by enlarging thermal gradient between ambient air and a die, and **ii)** minimize heat generation (power consumption) on hotspot by reducing the number of transitions. To our best knowledge, this is the first time that peak temperature is directly addressed with scan-vector optimization. Our major contributions are as follows:

- We show that scan vector order which is well known for influence on heat generation, also has a significant impact on *heat dissipation* of the CUT.
- We propose an efficient hotspot prediction algorithm by window-based power analysis.
- We propose an efficient scan vector ordering algorithm to minimize the peak temperature by maximizing the effectiveness of heat dissipation, minimizing heat generation and pre-compensating the overheat.
- The proposed algorithm complements nicely any existing low power techniques [7]–[12], without any overhead.

The remainder of the paper is organized as follows. In Section II, preliminaries are described. A motivating example is presented in Section III to explain the key idea of PEAKASO. The PEAKASO algorithm overview and details are explained in Section IV. Experimental results are discussed in Section V. Section VI concludes this paper.

II. PRELIMINARIES

A. Temperature Model

Inside the chip, the temperature distribution is governed by the following heat conduction equation [17] [18]:

$$\rho c_p \frac{\partial T(x,y,z,t)}{\partial t} = \kappa \bigtriangledown^2 T(x,y,z,t) + g(x,y,z,t) \quad (1)$$

where T is the time varying temperature (K), ρ is the density of the material (kg/m^3), κ is the thermal conductivity (W/mK), g is the time dependent volume power density (W/m^3), and c_p is the specific heat (J/kgK). Discretizing the

N_V	total number of scan vectors for a circuit
N_B	total number of thermal blocks in a circuit
$N_C(b)$	the number of scan cells in thermal block b
V_i	$i-th$ scan vector
P_{clk}	clock power consumption of one scan vector
$P(i)$	total power consumption of scan-in and scan-out of V_i
$P(b)$	total power consumption during test in thermal block b
$P(b,i)$	power consumption of scan-in and scan-out of V_i in thermal block b
$W(b,i)$	weighted transitions of scan-in and scan-out of V_i in thermal block b
α	average power consumption due to one transition
$\beta_{i,j}$	if the last bit of scan-out of V_i the same as the first bit of scan-in of V_j is, then 1. otherwise 0

(a) randomly ordered power slots (b) increasingly ordered power slots

(c) symmetrically ordered power slots (d) decreasingly ordered power slots

Fig. 1.   Thermal simulation results of four different power plots

continuous time domain and space domain, this partial differential equation can be solved by finite difference method [4], [17]. A unit of discretized time domain is the time taken to apply one scan vector, and a discretized space domain is defined as *Thermal Block* throughout this paper.

Inferring from Eqn. (1), if there is temperature difference across two geometric points, it creates thermal gradient which causes heat transfer from higher temperature point to lower temperature point. This heat transfer is proportional to thermal gradient. In VLSI circuits, heat flows from a die to ambient air which has near constant temperature [17]. Thus, one insight is that more heat can be dissipated to ambient air with higher average temperature on a die (larger thermal gradient between a die and ambient air).

B. Notations

Table I lists the notations used throughout this paper.

C. Power Model for Scan-based Testing

Larger weighted transitions consume more power in CUT [19]. Thus, power consumption can be represented as a function of weighted transitions. Then, power consumed in a thermal block b due to V_i can be shown:

$$P(b,i) = \alpha[W(b,i) + \beta_{i-1,i}N_C(b)] + \frac{P_{clk}}{N_B} \qquad (2)$$

Total power consumed in a thermal block b can be shown:

$$P(b) = \sum_{i=1,N_V} P(b,i) \qquad (3)$$

It is assumed that each transition consumes the same amount of power, α for simplicity. However, each transition can be characterized with different power consumptions for more accurate power estimation.

III. HEAT DISSIPATION AND PEAK TEMPERATURE

If the amount of power to be consumed in a die is fixed, only way to minimize peak temperature is to maximize heat dissipation to ambient air which is the primary heat flow path [4]. In this section, motivating thermal simulation results are presented to describe how to maximize heat dissipation for peak temperature reduction.
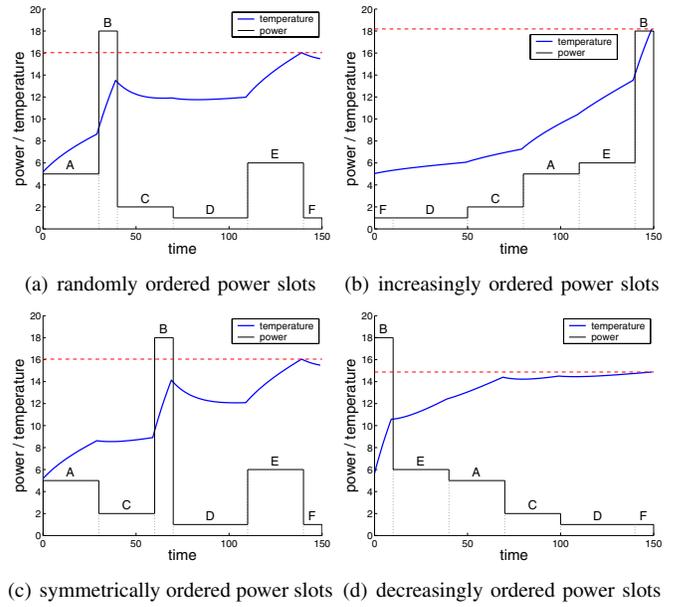
Consider Fig. 1 where fully optimized six power slots (A, B, C, D, E and F) are ordered in different manners. According to the way power slots are ordered, corresponding temperature plots and peak temperatures (dotted lines) are shown. Note that every power slot order has the equal total power consumption.

It can be seen that average temperature and peak temperature heavily depend on how power slots are ordered. While Fig. 1(b) shows the lowest average temperature and the highest peak temperature, Fig. 1(d) shows the highest average temperature but the lowest peak temperature. This is because for Fig. 1(d), high average temperature of a die creates large thermal gradient between a die and ambient air, which in turn expedites heat dissipation as discussed in Section II-A. All simulations show that with fixed amount of heat generation, as high power consuming slots are ordered early, the peak temperature is lower due to more efficient heat dissipation.

In the context of scan vector ordering (SVO), since total amount of heat generated (power consumed) by scan vectors is bounded, if more heat (power) is dissipated to air, peak temperature of CUT can be reduced. As a result, if high power consuming scan vectors are ordered early, peak temperature can be lowered. However, SVO for peak temperature minimization is not computationally trivial in three aspects.

1) Peak temperature is a geographically localized event so that location of hotspot (which produces peak temperature) should be known for peak temperature reduction.
2) SVO is a NP-hard which can be translated into asymmetric Travelling Salesman Problem (TSP) [13].
3) While generic SVO is only to minimize average power consumption, SVO for peak temperature minimization is to both maximize heat dissipation and minimize average power consumption (heat generation).

To efficiently overcome the above challenges, PEAKASO algorithm is proposed in Section IV.

IV. PEAKASO Algorithm

In this section, we present a peak temperature aware scan vector optimization algorithm, PEAKASO. The key idea of PEAKASO is that by ordering scan vectors, both heat dissipation to ambient air and heat generation from power consumption on hotspot, can be optimized for peak temperature reduction. The three steps of PEAKASO includes:

- **Hotspot Prediction:** Since peak temperature occurs on a localized hotspot, minimizing the peak temperature of CUT can be reduced to minimizing that of the hotspot. Thus, we propose an algorithm for hotspot prediction based on window-based power analysis.
- **Global Scan Vector Ordering:** To minimize peak temperature, heat dissipation and generation should be taken care of together. To achieve this, we use Prime algorithm-based greedy approach to find an optimized scan vector order for peak temperature reduction on hotspot.
- **Local Scan Vector Reordering:** After a certain number of scan vectors are applied, temperature may monotonically decrease. Thus, locally reordering scan vectors in temperature-decreasing region can further reduce the peak temperature on hotspot.

Algorithm 1 shows the overall flow of PEAKASO, which is explained in the rest of this section. Section IV-A discuss hotspot prediction. Section IV-B illustrates global scan vector ordering. Section IV-C explains local scan vector reordering.

Algorithm 1 PEAKASO

Input: Scan Cell Order O_s, Thermal Block List T_b
1: Scan Vector Order $O_v = \phi$
2: Hotspot H = Hotspot_Prediction(O_s,T_b)
3: **for** each hotspot h in H **do**
4:     O_v = Global_ScanVector_Ordering(O_s,h)
5:     O_v = Local_ScanVector_Reordering(O_v,O_s,h)
6:     Update the best O_v
7: **end for**
Output: O_v with the reduced peak temperature

A. Hotspot Prediction

The peak temperature of a CUT occurs on hotspot. Thus, if hotspot can be identified, computational complexity can be reduced, as optimization can be concentrated on hotspot. However, hotspot identification may not be trivial, as scan vector order and hotspot are interdependent on each other. An exact scan vector order is required to estimate hotspot accurately, while hotspot is essential to find a scan vector order for minimal peak temperature. In this subsection, we propose an efficient algorithm for hotspot prediction based on window-based power analysis.

The temperature of a thermal block b is mainly proportional to $P(b)$. But, power consumed by neighbor blocks should be also considered, as heat flows along thermal gradient between thermal blocks. Fig. 2(a) and (b) show a power profile and temperature profile of a circuit respectively. If heat from

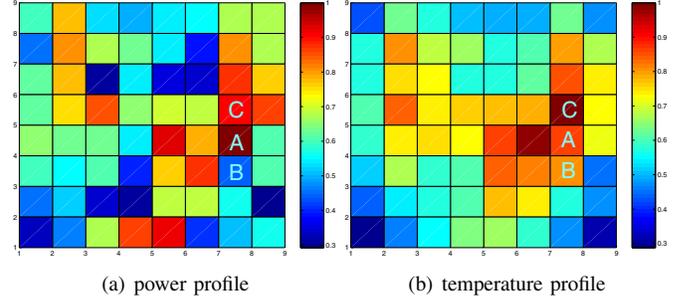(a) power profile   (b) temperature profile

Fig. 2. Power profile and temperature profile of s38415 circuit

neighbor blocks is ignored, a thermal block A will be predicted as hotspot, because A is the most power consuming block as in Fig. 2(a). But, the thermal profile in Fig. 2(b) indicates that actual hotspot is not A but C. This misprediction happens, because a thermal block B (abutting to A) consumes relatively small amount of power. Thus, heat generated on A flows to B along thermal gradient, cooling down A. Thus, a metric of thermal block b to predict hotspot should be window-based to take neighbor thermal blocks into account:

$$M(b) = P(b) + \sum_{n \in \omega} \gamma_n P(n) \qquad (4)$$

where γ_n is a parameter which is inversely proportional to the distance between a thermal block b and n, and ω is neighbor thermal blocks around b. Note that γ_n and ω depend on process technology and granularity of thermal simulation. Since scan vector order is not determined yet, $M(b)$ has the lower bound and upper bound as shown:

$$M(b)^L \mid_{\beta=0} \leq M(b) \leq M(b)^U \mid_{\beta=1} \qquad (5)$$

As each thermal block has the upper bound and lower bound, it is hard to decide which thermal block is hotspot. For example, in case that $M(A)^L \leq M(B)^U \leq M(A)^U$ w.r.t two thermal blocks A and B, either can be hotspot depending on the final scan vector order (β values between two successive vectors).

Consider Fig. 3 where a die is only composed of four thermal blocks. In Fig. 3(a), each thermal block b has $P(b)^L \mid_{\beta=0}$ and $P(b)^U \mid_{\beta=1}$. Assuming $\gamma = 0.5$ and ω is a set of abutting blocks, $M(b)^L$ and $M(b)^U$ can be computed for every block by Eqn. (4) as shown in Fig. 3(b). For example,

- $M(2)^U = P(2)^U + 0.5[P(1)^U + P(3)^U] = 36$.
- $M(3)^L = P(3)^L + 0.5[P(2)^L + P(4)^L] = 29$.

Then, the block 3 becomes the first hotspot, as $M(3)^U$ is the biggest (line 2 Algorithm 2). Additionally, the block 2 can be a

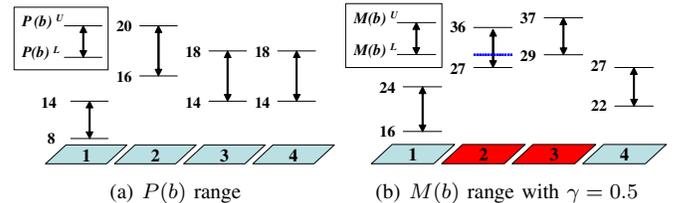(a) $P(b)$ range   (b) $M(b)$ range with $\gamma = 0.5$

Fig. 3. Example of hotspot prediction

hotspot as $M(3)^L \leq M(2)^U$ (line 3-7 Algorithm 2). Note that this prediction does not need expensive thermal simulation, and works for all the circuits in Table III. Overall hotspot prediction is presented in Algorithm 2.

Algorithm 2 Hotspot_Prediction

Input: Scan Cell Order O_s, Thermal Block List T_b
1: Hotspot $H = \phi$
2: Find a block B with the biggest $M(B)^U$
3: **for** each block b in T_b **do**
4:    **if** $M(B)^L \leq M(b)^U$ **then**
5:       Add b into H
6:    **end if**
7: **end for**
Output: H

B. Global Scan Vector Ordering

As scan vector ordering (SVO) is a NP-hard, a SVO can be converted into a complete graph where each scan vector becomes a vertex, and an edge cost from V_i to V_j on hotspot h can be computed as:

$$C(\overrightarrow{i,j}) = \beta_{i,j} F_b(h) + W_b(h, j) \qquad (6)$$

Then, we need to find a scan vector order satisfying the following three conditions for peak temperature minimization.

Φ Each scan vector should appear once in the final scan vector order, to keep the same fault coverage and the same number of scan vectors.

Ψ High power consuming scan vectors should be ordered earlier to reduce the peak temperature (**maximize heat dissipation with larger thermal gradient**).

Ω The summation of the edge cost should be minimized to reduce the average power consumption (**minimize heat generation**).

Satisfying Ψ and Ω together may not be possible. Consider Fig. 4 where three scan vector V_a, V_b and V_c are ordered in two different ways, assuming V_a is already chosen. If an order of Fig. 4(a) is selected, it satisfies Ψ condition, but it has higher power consumption (violating Ω), which potentially increases peak temperature. If another order of Fig. 4(b) is selected, it has lower power consumption (satisfying Ω), but peak temperature may increase due to the violation of Ψ condition. Thus, it is hard to evaluate which order provides the lower peak temperature.
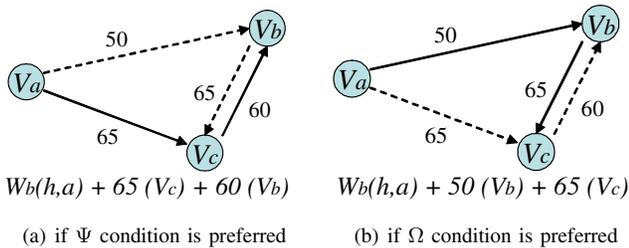

Fig. 4. Trade off between Ψ and Ω conditions

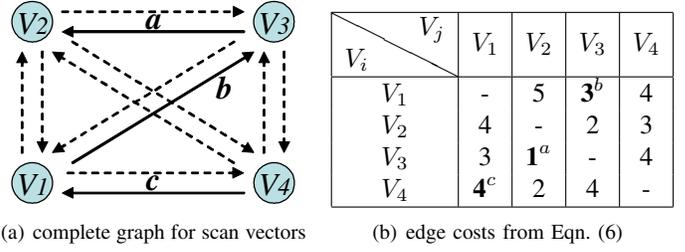(a) complete graph for scan vectors   (b) edge costs from Eqn. (6)

Fig. 5. Example of global scan vector ordering

We adopt a greedy heuristic based on Prime algorithm to find a decent trade off between Ψ and Ω conditions. Basically, an incoming edge with the smallest cost to each vertex is chosen, while keeping Φ condition. In Fig. 5, a complete graph of four scan vectors (V_1, V_2, V_3 and V_4), and all edge costs are shown. The final vector order is constructed by following:

(a)   The edge with the smallest cost ($V_3 \rightarrow V_2$) is chosen. Put V_2 into the last empty slot of scan vector order.

(b)   Since V_2 is already in the order, the next smallest incoming edge ($V_1 \rightarrow V_3$) is selected. Put V_3 into the last empty slot of the order.

(c)   As only one vector is left, put V_1 and V_4 into the last empty slots of the order. Then, final scan vector order is $V_4 \rightarrow V_1 \rightarrow V_3 \rightarrow V_2$.

This approach is faithful to Ψ condition, because an edge with smaller cost tends to be picked earlier, and an edge with larger cost tends to be picked later. As the scan vector order is built from the end, high power consuming scan vector tends to be ordered earlier. And, each time the next scan vector is picked, it greedily seeks for an edge with the smallest cost, observing Ω condition. Overall global scan vector ordering is presented in Algorithm 3.

Algorithm 3 Global_ScanVector_Ordering

Input: Scan Cell order O_s, Hotspot h
1: Scan Vector Order $O_v = \phi$
2: Calculate $C(i, j)$ for any V_i and V_j, $i \neq j$ w.r.t h
3: Find an edge with the smallest cost
4: **repeat**
5:    Find the smallest valid incoming edge, and update O_v
6: **until** all scan vectors are selected once
Output: O_v

C. Local Scan Vector Reordering

A global scan vector order from Algorithm 3 may show monotonic temperature decrease after a certain number of scan vectors are applied, because low power consuming scan vectors are ordered later due to global scan vector ordering. (heat generation from scan vectors is getting smaller than heat dissipation to air). Thus, reordering scan vectors locally after when heat generation becomes smaller than heat dissipation can further improve the scan vector order to reduce the peak temperature.

(a) temperature plot (b) power plot

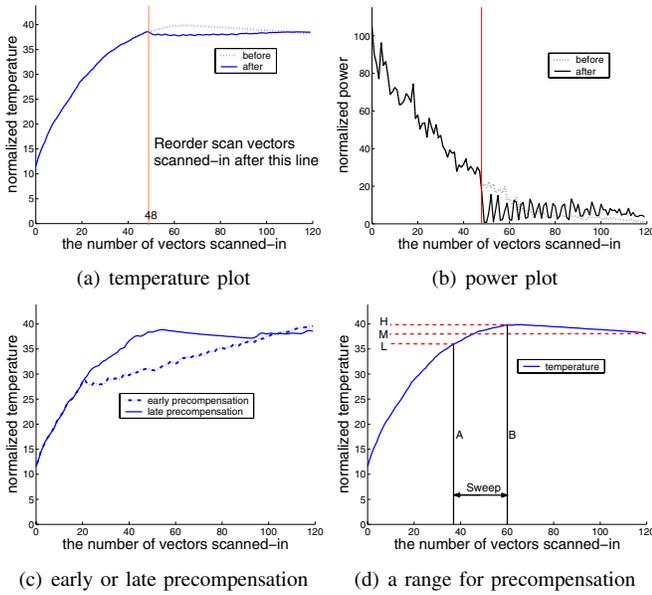(c) early or late precompensation (d) a range for precompensation

Fig. 6. Local scan vector reordering with s15850 circuit

Consider Fig. 6(a) where the temperature plot of the scan vector order by Algorithm 3 is shown in dotted, and the temperature plot after local reordering is shown solid line. It can be observed that local scan vector reordering reduces the peak temperature further. The reason for this reduction is described in Fig. 6(b) where the power plots, before and after local scan vector reordering, are shown in dotted line and solid line respectively. While dotted line shows near-monotonic decrease, solid line shows vertical swings. Those swings happen, because low power consuming scan vectors are moved in front of high power consuming vectors. This alternative ordering makes low power consuming scan vectors precompensate the heat from high power consuming vectors.

The proposed local scan vector reordering is based on overheat precompensation where overheat is defined as excessive power consumption above average power consumption during a certain period. Table II shows how a scan vector order from Section IV-B can be locally reordered.

1) The average weight transition after vector 3 is 10. Thus, if scan vector 13 is immediately applied after scan vector 3, the overheat becomes $35 - 10 = 25$ as in step 1.
2) To compensate the overheat from vector 13, partial scan vector order $2 \rightarrow 7 \rightarrow 6 \rightarrow 9$ is inserted in front of 13. Then, the overheat after vector 13 reduces to -2 as in step2.
3) The same procedure is repeated for scan vector 53 and 52. Finally, there is no positive overheat in step 3.

The result of local scan vector reordering depends on the first precompensated vector (like vector 13 in Table II). Figure 6(c) shows when precompensation starts early or late. If it starts too early, the rapid temperature increase can be mitigated, but it ends up with higher peak temperature. If it starts too late, it also ends up with higher peak temperature without taking full advantage of precompensation. Since it is hard to pick an optimal precompensation starting vector , we

TABLE II

EXAMPLE OF LOCAL SCAN VECTOR REORDERING

step 1 : compensation of 13												
i	3	**13**	53	52	8	5	1	4	**2**	**7**	**6**	**9**
w	..	35	21	14	9	7	6	5	4	3	4	2
o	..	25	11	4	-1	-3	-4	-5	-6	-7	-6	-8
a	..	**25**	36	40	39	36	32	27	21	14	8	0
step 2 : compensation of 53												
i	3	2	7	6	9	**13**	*53*	52	8	5	1	4
w	..	4	3	4	2	35	21	14	9	7	6	5
o	..	-6	-7	-6	-8	25	11	4	-1	-3	-4	-5
a	..	-6	-14	-20	-27	**-2**	*9*	13	12	10	6	0
step 3 : completed												
i	3	2	7	6	9	**13**	1	4	**53**	8	5	**52**
w	..	4	3	4	2	35	6	5	21	9	7	14
o	..	-6	-7	-6	-8	25	-4	-5	11	-1	-3	4
a	..	-6	-14	-20	-27	**-2**	-6	-11	**0**	-1	-4	**0**

i: scan vector w: $W(h,i)|_{\beta=1}$
o: overheat = w - average of w a: accumulated overheat

propose to sweep a range, which yields good results for all the circuits in Table III. The way to find such range is depicted in Figure 6(d). With a thermal simulation of globally ordered scan vectors from Algorithm 3, the temperature plot can be obtained. Then, the peak temperature on a line H and the final temperature on a line M can be found. Another line L such that temperature gap between H and M is the same as the gap between M and L can be found. Finally, the range between two vertical lines, A and B in Figure 6(d) can be swept to find the best precompensation starting vector for the lowest peak temperature. Local scan vector reordering is presented in Algorithm 4.

V. EXPERIMENTAL RESULTS

We implement our algorithm in C++ and run the program on a 1.5GHz Pentium-4 PC. The ISCAS89 circuits [20] are synthesized (placement, routing and scan routing) by Silicon Ensemble [21] with 0.16 μm CMOS standard library. An ATPG tool, ATLANTA [22] is used to generate the deterministic scan vectors for each circuit with maximum compaction. Minimum-Transition Fill [23] is performed on all the scan vectors to reduce the number of scan shiftings. Power dissipations are estimated based on [24] with 50MHz test frequency, and Hotspot [4] is used for thermal simulations. All the physical constants like thermal conductivity and specific heat are as in Hotspot package.

Algorithm 4 Local_ScanVector_Reordering

Input: Scan Vector Order O_v, Scan Cell order O_s, Hotspot h
1: R = Find a range to sweep
2: **for** each scan vector r in R **do**
3:     Calculate average weighted transition after r
4:     Precompensate overheats after r
5:     Update the best O_v by thermal simulation
6: **end for**
Output: The best O_v

TABLE III
THERMAL SIMULATION RESULT FOR ISCAS89 CIRCUITS.

circuit	the num of scan cells	the num of scan vectors	normalized peak temperature			imprv over(%)		freq trans
			rand[a]	G.A.[b]	PEAKASO	rand	G.A.	G.A.(MHz)
s13207	699	239	27.4	26.4	23.9	13.0	9.7	80 (60%)
s15850	597	120	31.6	30.3	28.6	9.6	5.7	73 (46%)
s38417	1636	95	68.2	67.4	64.3	5.8	4.6	70 (40%)
s38584	1452	131	79.8	76.0	72.8	8.8	4.3	70 (40%)

(a) Random Ordering - average of 100 random scan vector orders   (b) Greedy_ATSP in [13]

Table III shows thermal simulation results for the selected benchmark circuits. The scan vectors for each circuit are ordered by three different algorithms, Random ordering, Greedy_ATSP [13] and PEAKASO, and all scan vector orders are simulated to obtain peak temperatures. Note that Greedy_ATSP orders scan vectors for average power (heat generation) reduction, while PEAKASO orders scan vectors for peak temperature reduction by maximizing heat dissipation and minimizing heat generation.

The result shows that PEAKASO provides the scan vector orders of the lowest peak temperature for all the tested circuits. PEAKASO reduces the peak temperature by 5.8 - 13.0% over Random ordering, and 4.3 - 9.7% over Greedy_ATSP, keeping the same fault coverage (scan vectors are never modified). Note that scan vector orders by PEAKASO have 1.8% more weighted transitions than those by Greedy_ATSP on average. It is obvious that for peak temperature reduction, both heat dissipation and heat generation should be taken into account as done by PEAKASO. One observation is that the improvement is proportional to the number of scan vectors, implying that PEAKASO can be more efficient to industrial scan vectors.

This peak temperature reduction can be translated into increased test speed, because the amount of reduced temperature can be used as temperature budget. With the scan vector orders by PEAKASO, we increase test clock frequency from 50MHz gradually, until the peak temperature reaches that by Greedy_ATSP (for s13207, until the peak temperature reaches 26.4). It shows that the peak temperature reduction from PEAKASO enables to increase testing speed from 40% to 60%. Finally, Note that PEAKASO does not involve any overhead in terms of hardware, fault coverage and testing time.

VI. CONCLUSION

Peak temperature is one of the bottlenecks in current VLSI testing. In order to minimize peak temperature during scan-based testing, we present an efficient, peak temperature aware scan vector optimization algorithm for scan chains, PEAKASO which consists of three steps, hotspot prediction, global scan vector ordering and local scan vector reordering. Experimental results show that our approach reduces the peak temperature of CUT significantly (up to 10%) without any overhead. Such peak temperature reduction can be translated into faster test time (up to 60%) under the same peak temperature constraint.

VII. ACKNOWLEDGEMENT

The authors would like to thank Ilsoo Lee and Jinkyu Lee for useful discussion, and Prof. Nur Touba for valuable advice.

REFERENCES

[1] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," in *Proc. VLSI Test Symp.*, 1993, pp. 4–9.
[2] C. Shi and R. Kapur, "How power-aware test improves reliability and yield," in *EEDesign.com (http://www.eetimes.com/news/design/features/showArticle.jhtml?articleId=47208594&kc=4235)*, Sep 2004.
[3] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Test Power: a Big Issue in Large SOC Designs," in *IEEE Int. Workshop on Electronic Design, Test and Applications*, 2002, pp. 447–449.
[4] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact Thermal Modeling for Temperature-Aware Design," in *Proc. Design Automation Conf.*, June 2004.
[5] P. Girard, "Survey of low-power testing of VLSI circuits," *IEEE Design & Test of Computers*, vol. 19, pp. 80–90, May 2002.
[6] ——, "Low Power Testing of VLSI Circuits: Problems and Solutions," in *Proc. Int. Symp. on Quality Electronic Design*, 2000.
[7] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Power Driven Chaining of Flip-flops in Scan Architectures," in *Int. Testing Conf.*, 2002, pp. 796–803.
[8] ——, "Efficient Scan Chain Design for Power Minimization During Scan Testing Under Routing Constraint," in *Int. Testing Conf.*, 2003.
[9] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, and A. Virazel, "Design of Routing-Constrained Lower Power Scan Chains," in *Proc. Design, Automation and Test in Eurpoe*, 2004.
[10] S. Ghosh, S. Basu, and N. A. Touba, "Joint Minimization of Power and Area in Scan Testing by Scan Cell Reordering," in *Proc. IEEE Annual Symp. on VLSI*, 2003, pp. 246–249.
[11] S. Kajihara, K. Ishida, and K. Miyase, "Test Vector Modification for Power Reduction during Scan Testing," in *Proc. VLSI Test Symp.*, 2002.
[12] R. Sankaralingam and N. A. Touba, "Controlling Peak Power During Scan Testing," in *Proc. VLSI Test Symp.*, 2002, pp. 160–165.
[13] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, pp. 1325–1333, Dec 1998.
[14] P. Girard, L. Guiller, S. Pravossoudovitch, and D. Severac, "Reducing power consumption during test application by test vector ordering," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 1998, pp. 296–299.
[15] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A Test Vector Ordering Technique for Switching Activity Reduction During Test Operation," in *Proc. Great Lakes Sympo. on VLSI*, 1999.
[16] M. Bello, D. Bakalis, D. Nikolos, and X. Kavousianos, "Low Power Testing by Test Vector Ordering with Vector Repetition," in *Proc. Int. Symp. on Quality Electronic Design*, 2004, pp. 205–210.
[17] T. Wang and C. C. Chen, "3D Thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 12, Dec 2002.
[18] M. N. Ozisik, "Boundary Value Problems of Heat Conduction," *Oxford University Press, Oxford*, 1968.
[19] R. Sankaralingam and N. A. Touba, "Reducing Power Dissipation During Test Using Scan Chain Disable," in *Proc. VLSI Test Symp.*, 2001.
[20] F. Brglez, D. Bryant, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 1989, pp. 1929–1934.
[21] "Silicon Ensemble User Guide," in *Cadence Design System*, 2000.
[22] http://www.ee.vt.edu/~ha/cadtools/.
[23] R. O. R. Sankaralingam and N. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," in *Proc. VLSI Test Symp.*, 2000, pp. 35–40.
[24] V. Saxena, F. N. Najm, and I. N. Hajj, "Monte-Carlo Approach for Power Estimation in Sequential Circuits," in *Proc. European Design and Test Conf.*, 1997, pp. 416–420.