Power-Aware SoC Test Optimization through Dynamic Voltage and Frequency Scaling

Vijay Sheshadri, Vishwani D. Agrawal and Prathima Agrawal
Department of Electrical and Computer Engineering
Auburn University
Auburn, AL 36849, USA
Email: {vijay, agrawvd, agrawpr}@auburn.edu

***Abstract*—Reducing test cost by minimizing the overall test time remains one of the main goals of System-on-Chip (SoC) testing. Power-aware strategies optimize the overall test time of a SoC for a global peak power budget. Test time and test power can be regulated by V_{DD} and test clock frequency to optimize SoC test schedules for a given power budget. Dynamic voltage and frequency scaling (DVFS) techniques have been used in the past to optimize energy efficiency in SoCs. In this paper, we extend the concept of DVFS to optimize the test scheduling of SoC. We adopt a sessionless test scheduling strategy and provide a simple heuristic approach for its optimization. The proposed idea is implemented on several ITC02 benchmarks. Results show significant test time reduction over sessionless reference test schedules for which V_{DD} and clock frequency are fixed at nominal values.**

I. Introduction

A system-on-chip (SoC) may contain an entire system integrated onto a single chip. Such an SoC is often implemented by embedding reusable blocks called cores. The cores can be a variety of intellectual property blocks, such as digital logic, processor, memory, and analog or mixed signal circuit. Due to technology scaling, it is now possible to accommodate a larger number of cores on an SoC device [1]. The resulting increase in the complexity of SoC devices has led to high volumes of test data and long test times. Minimizing test time is, therefore, one of the major concerns in SoC testing research. Some of the approaches taken towards achieving this objective involve test architecture design and optimization, test scheduling optimization etc.

Test scheduling can be defined as a process of scheduling the tests associated with various cores of the SoC such that resource and power requirements are satisfied. Test schedules can be optimized for better resource and power management and a quicker overall test time. Existing test scheduling strategies can be broadly classified into:

- Session-based (non-partitioned) test scheduling, where no new test is allowed to start until all tests of a previous session are completed. A test session refers of a set of tests initiated simultaneously and run concurrently.
- Sessionless (partitioned) test scheduling, where test session boundaries are ignored and a test may be scheduled to start as soon as possible. The sessionless or partitioned test scheduling can be further divided into preemptive and non preemptive scheduling. In the preemptive strategy tests can be interrupted or restarted at any time. The non preemptive strategy does not allow such interruptions, i.e., a test once initiated must complete.

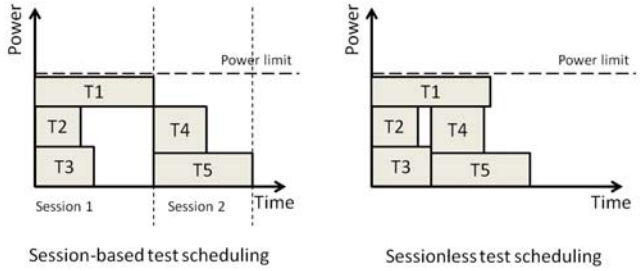

Fig. 1. Two test scheduling strategies.

Figure 1 shows how a session-based strategy introduces idle time gaps in the test schedule. Sessionless test scheduling provides a better alternative in terms of test time reduction at the cost of test hardware and scheduling complexities. It overcomes the drawback of the session-based testing by scheduling new tests as soon as a running test completes and hence reduces the idle time gaps.

The power consumption of a circuit during test mode is often higher than the functional mode. Simultaneously testing multiple cores can reduce the test time significantly, but such concurrent execution is limited by high power consumption on the SoC. Therefore, power-aware test strategies are needed for efficient test power management. Recently, reduced voltage testing has been shown to significantly reduce test time, under specified power constraints [28], [30]. By scaling the supply voltage during testing, the test power can be lowered drastically, thereby, allowing increase in the test clock rate without violating the power limit. The concept of varying V_{DD} and clock frequency for SoCs has been proposed in the past. Termed as dynamic voltage and frequency scaling (DVFS) techniques, they have been adopted to optimize energy efficiency, leakage power management, etc., in multi-core SoCs [3], [29]. In this paper, we propose heuristic approaches for sessionless power-constrained test scheduling that make use of DVFS techniques to minimize the overall test time of an SoC. We compare the results of our method with a reference case where the clock and the supply voltage are fixed for the entire test schedule at pre-specified nominal values. The test schedules for the

978-1-4799-0524-9/13/$31.00 ©2013 IEEE

reference case are created by adopting the best-fit decreasing (BFD) heuristic. The remainder of this article is organized as follows: Section II outlines the related work in the area of SoC testing. The heuristic methods are presented in Section III. Section IV discusses the results and finally Section V concludes the paper.

II. Related Work

In the past, the test schedule optimization problem has been tackled by Integer Linear Program (ILP) and mixed-ILP (MILP) solvers [4], [16], [27]. However, ILP methods are NP-hard and, in general, computationally expensive for large SoCs. The CPU time required to obtain an optimal solution increases exponentially as the number of cores and the complexity of the SoC increases. Heuristic algorithms, often employing greedy approaches, perform much better in terms of CPU time as compared to exact methods such as ILP. While a heuristic method does not guarantee an optimal solution, a good algorithm can produce near-optimal values, consistently. Some of the heuristics that have been proposed for SoC test scheduling include tree-growing algorithm [24], bin packing [13], [15] and simulated annealing [12], [36].

Power-constrained test scheduling, where a test power budget is defined for the SoC, has been studied in the past [7], [10], [14], [20], [26], [34]. Some of these papers discuss test scheduling optimization through optimal allocation of test access mechanism (TAM) resources of the SoC to core tests. Several papers [15], [16], [18], [25], [34] have considered test time reduction through optimizing TAM and wrapper chains of the core. In our work, we make use of the test clock frequency to reduce test time and it is assumed that the design for testability (DFT) infrastructure is already in place for the SOC and that the TAM assignment and the wrapper design have been optimized.

Preemptive testing, where a test can be interrupted and restarted, has been considered by several authors [14] and [19], [20]. While preemption can help in reducing the test time, not all tests can be preempted (e.g., memory BIST). Besides, this technique introduces additional complexity in the scheduling process and the test control circuitry.

The idea of dynamically scaling voltage and frequency has been prevalent in the field of microprocessors and SoCs. In [29], a locally placed configurable dynamic voltage and frequency scaling (DVFS) controller enables a large number of on-chip processors to switch V_{DD} by selecting from two power grids and also independently control their clock rates, in order to improve the energy efficiency of the multi-processor SoC (MPSoC).

DVFS techniques have also been employed in testing of SoCs. While reducing the voltage for power reduction in scan testing, the authors of a recent paper [8] suggest dynamic control of voltage and frequency for reduction of test time. In another paper [17] scheduling with multiple voltage islands and testing of cores at multiple voltages has been considered. Those authors schedule core tests at multiple voltage levels and clock domains and reduce the clock frequency during low voltage testing to enable a time division multiplexing scheme for concurrent testing of cores.

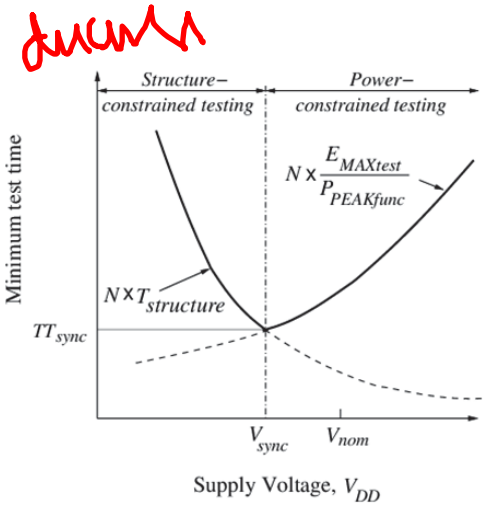

Fig. 2. Test time as a function of V_{DD} [31], [32]. The nominal and the optimal V_{DD} are denoted by V_{nom} and V_{sync}, respectively.

The focus of the present work is in the reduction of test application time (TAT) for core tests. Also, the current work considers power aware testing whereas in [17], the power constraint during test scheduling is not considered. Venkataramani *et al.* [30]–[33] discuss two aspects of testing, namely, power constrained testing where the test clock speed is limited by the circuit's rated power and structure constrained testing where the test clock speed is limited by the critical path or other timing constraints of the circuit. The supply voltage is used for the purpose of balancing these two constraints to allow higher test clock rates in order to achieve test time reduction. Since test power is two to four times higher than the functional power, test clock is often power constrained, i.e., any increase in the clock would cause the power to exceed the device's rated maximum. The power consumption can be reduced by lowering the operating voltage. As a result, the clock rate can be increased without exceeding the power constraint of the core. However, reducing the voltage causes the delay of a circuit to increase, hence, elongating the critical path of the device. Thus, as we reduce V_{DD}, on one hand, the lowered power consumption allows higher clock rates thereby shrinking the test time but, on the other hand, the increased circuit delay requires slower clock rate and a longer test time. As Figure 2 [31], [32] shows there exists an optimal point where the two constraints are satisfied and at the same time test time is significantly reduced. Experiments on ISCAS benchmark circuits by those authors show test time reductions of up to 62% at optimal values of V_{DD} [30].

In [28], the authors use voltage and frequency scaling to minimize the SoC test time in a session-based test schedule. In our work, we adopt a sessionless test scheduling strategy, which is known to achieve lower test times. Also, in [28], an ILP method is used to optimize the test time for a given power constraint. The drawback of the ILP is that as the SoC size grows, the solution becomes intractable. Hence, we develop a heuristic approach towards optimization which can be easily applied to large SoCs.

BFD Heuristic
list1 = list of core tests to be scheduled {initially contains all core tests}
list2 = list of core tests currently executed {initially empty}
$t_{sch} = 0$ {overall test time of the test schedule}
list1.sort(key = power, reverse = True)
while list1 is **not** empty **or** list2 is **not** empty **do**
  for each test i in list1 **do**
    if $P < P_{max}$ **then**
      insert test i into list2
      delete test i from list1
      $P = \Sigma P_i, \forall i \in list2$
    else
      remove recently added test from list2
    end if
  end for
  $t_{sch} = t_{sch} + min(t_i, \forall i \in list2)$
  delete the test with smallest test length from list2
  for all remaining tests in list2 **do**
    update test length
  end for
end while

Fig. 3. Best-fit decreasing (BFD) algorithm for sessionless test scheduling. Test schedules obtained from this algorithm are used as reference cases in this paper where voltage and frequency are fixed at their nominal values for the entire schedule.

III. Heuristic Methods

The new heuristic approaches given in this paper employ DVFS for optimizing SoC test schedules. A simple heuristic algorithms are proposed for both preemptive and non-preemptive sessionless test scheduling. The frequency scaling is performed in the main procedure of the algorithm whereas voltage scaling is done by calling a function.

For comparison with DVFS schedules, we create an algorithm to generate reference sessionless schedules with voltage and frequency fixed at nominal values. The test scheduling process is modeled as bin packing. An individual core test is treated as a block with test power as height and test time as width. A best-fit decreasing (BFD) heuristic then solves the bin packing problem. The tests are sorted in decreasing order of their power consumption and stacked together in such a way that at any given time in the test schedule the total power does not exceed a specified P_{max}.

The algorithm for the reference case is provided in Figure 3. The procedure first sorts the list of unscheduled core tests in the decreasing order of their test power. Next, each test from this list is 'scheduled' by relocating it to a new list. This new list contains tests that are currently running. This step is repeated until the total test power is as close to the power limit as possible. After the completion of a test, a new test is added to the schedule from the sorted list. This whole process is repeated until all core tests are scheduled. The end time of the final test is the total test time of the test schedule.

Since scaling voltage and frequency alters the test time and power of a core test, test scheduling with DVFS cannot be modeled as a bin packing problem. Hence, a simple heuristic is formulated, which inserts tests into the test schedule until the test power reaches P_{max}. The clock frequency and the voltage are adjusted as the tests are being inserted into the schedule.

Heuristic 1
list1 = list of core tests to be scheduled {initially contains all core tests}
list2 = list of core tests currently executed {initially empty}
$t_{sch} = 0$ {overall test time of the test schedule}
while list1 is **not** empty **do**
  list2 = empty list
  while $P' < P_{max}$ **do**
    insert random test i into list2
    delete test i from list1
    $F = min(f_i, \forall i \in list2)$
    $P = \Sigma P_i, \forall i \in list2$
    $P' = P \times F$
  end while
  if $P' > P_{max}$ **then**
    if $P > P_{max}$ **then**
      remove recently added test from list2
    else
      $F = \frac{P_{max}}{P}$
    end if
  end if
  $t_{sch} = t_{sch} + \frac{min(t_i, \forall i \in list2)}{F}$
  delete the test with smallest test length
  for all remaining tests **do**
    preempt the test and add it to list1 with updated test length
  end for
end while

Fig. 4. Preemptive (Heuristic 1) algorithm for sessionless test scheduling.

In the first heuristic approach (Figure 4), we assume that a test can be suspended (preempted) and then resumed at any time. This allows us to partition a single test into multiple tests, each with a smaller test length. We also assume that these partitioned tests can execute independently. Each test has an equal probability of being scheduled and tests are chosen to be scheduled at random, as long as their test power does not cause a violation of the power limit. The test clock rate is scaled by a factor F, which reduces the test time but increases the power consumption proportionately. Hence, P_{max} limits the scaling factor F. The scaling factor is also restricted by the maximum frequency limit of individual cores (f_i), decided by the cores' critical path or rated power limit. Since cores tested concurrently share the same test clock, the clock rate is decided by the slowest core among them. Hence $F = min\{min(f_i), \frac{P_{max}}{\sum(P_i)}\}$, where i belongs to the set of core tests that are currently being executed.

If the session power exceeds P_{max}, depending on what caused it a power correction step is performed. If the cause is addition of an extra test into the session, the extra test is put back from $list2$ to $list1$, else if the excess is due to high clock rate of the session, the clock rate is lowered accordingly. On the completion of a test, the remaining tests in $list2$ are preempted. A preemption implies that the tests are suspended and the remainder of the tests are treated as new tests to be scheduled later. These 'new tests' are added to the $list1$. This process is repeated until $list1$ becomes empty, indicating that all cores of the SoC have been included in the test schedule.

Preemption increases the complexity of the scheduling algorithm and in some cases it may be undesirable to preempt a

Heuristic 2
list1 = list of core tests to be scheduled {initially contains all core tests}
list2 = list of core tests currently executed {initially empty}
$t_{sch} = 0$ {overall test time of the test schedule}
while list1 is **not** empty **or** list2 is **not** empty **do**
  while $P' < P_{max}$ **do**
    insert random test i into list2
    delete test i from list1
    $F = min(f_i, \forall i \in list2)$
    $P = \Sigma P_i, \forall i \in list2$
    $P' = P \times F$
  end while
  if $P' > P_{max}$ **then**
    if $P > P_{max}$ **then**
      remove recently added test from list2
    else
      $F = \frac{P_{max}}{P'}$
    end if
  end if
  $t_{sch} = t_{sch} + \frac{min(t_i, \forall i \in list2)}{F}$
  delete the test with smallest test length from list2
  for all remaining tests **do**
    retain the test in list2
    update test length
  end for
end while

Fig. 5. Non-preemptive (Heuristic 2) algorithm for sessionless test scheduling.

test. Hence, we propose another heuristic that, on completion of a test, randomly adds a new test to the schedule (Figure 5) while continuing the tests already executing to run to completion. In other words, the continuing tests are not preempted.

These heuristic procedures only perform frequency scaling. For voltage scaling, we define a function $VScale()$ (Figure 6). As the supply voltage is scaled, it not only affects the test power but also the maximum frequency limits of each core test. As mentioned earlier, the clock rate each core is limited by is its rated maximum power (power constraint) and its critical path (structural constraint). The dependence of these limits on V_{DD} is given in [28]. The voltage is decreased from its nominal value, in small steps. For each step, a function call is made to $VScale()$ before computing the test time (t_{sch}). In the function, $VScale()$, the new power and structure constraints along with the test power are calculated. The scaling factor F is updated based on these changed values. The new test time, due to the scaled voltage, is then calculated. If the voltage reduction led to a reduction in test time, the process is repeated until V_{DD} reaches a pre-defined lower limit. Else, V_{DD} is incremented by one step and no more scaling takes place.

IV. Results

The proposed heuristic algorithms was applied to several ITC'02 benchmarks [2]. Vector by vector power consumption at nominal voltage (assumed 1V) and clock frequency for ITC'02 benchmark circuits was obtained from Millican and Saluja [23]. Their data provided the peak power P_i and test time for each core i at nominal operating conditions. The core of an SoC with highest test power and longest test time was

VScale()
Calculate power constraint limit (f_{p_i})
Calculate structure constraint limit (f_{s_i})
Calculate new test power (P_i')
Update clock scaling factor,
$F = min\{min(f_{p_i}, f_{s_i}), P_{max}/\sum(P_i')\}$
return F

Fig. 6. Function to perform dynamic voltage scaling.

regarded as the slowest and the rest of the cores in the SoC were normalized with respect to that core. To set limits on the frequency of individual cores, higher peak clock rate was assigned to a core based on how much lower its power P_i was. The experiments were preformed on a Dell workstation with a 3.4GHz Intel Pentium processor and 2GB memory.

Test times obtained for the benchmarks are given in Table I. Reference cases (column 4) are the fixed nominal voltage and clock frequency schedules obtained from the algorithm of Figure 3. These schedules are sessionless and have test time that may already be shorter than those for the conventional session-based test schedules.

The next two cases in Table I are DVFS schedules. For each benchmark, the proposed heuristic algorithms were iterated until the best solution obtained did not improve for ten thousand consecutive runs. Percent reductions are with respect to the reference cases of column 4. We notice that by scaling the voltage and frequency dynamically the test time can be shortened by 45-60%. We also observe that the preemptive and non-preemptive strategies yield almost identical solutions. However, the preemptive strategy introduces extra complexity in the scheduling process by adding the preempted tests as new tests to the list of unscheduled core tests after the completion of each test. This phenomenon is reflected in the CPU times. With more tests being added to the scheduling list due to preemption, the number of while loops executed in the heuristic increases as do the calls to the voltage scaling function. The combined effect leads to a longer CPU time for the preemptive algorithm.

The heuristic algorithms were also applied to ASIC Z SoC. ASIC Z was introduced by Zorian [35] and consists of RAM, ROM and other blocks (Figure 7). It has been used as benchmark in the past for implementing various test scheduling ideas. Chou *et al.* [7] reported a test time of 331 units whereas Larsson and Peng [21] obtained a test time of 300 units through their optimization methods. In Table II, we compare our results with recently published test times.

Harmanani *et al.* [12] adopt a sessionless test schedule and use a simulated annealing algorithm for optimization whereas [27] and [28] adopt a session-based test strategy and use ILP for optimization. They use session-wise frequency scaling at a nominal voltage [27] as well as at an optimally selected voltage [28]. In either case, the voltage V_{DD} remains fixed for the entire test schedule. [28] also employs voltage and frequency scaling to reduce SoC test time but their optimal test time is an ILP solution for session-based test schedule. The heuristic solution for the sessionless test schedule provides an improvement of at least 5.24% over the ILP based optimal

TABLE I
TEST TIMES (IN ARBITRARY UNITS) FOR SESSION-LESS TEST SCHEDULING WITH DVFS.

Benchmarks	No. of cores	P_{max}	Fixed voltage/frequecy sessionless test time	Preemptive DVFS scheduling			Non-preemptive DVFS scheduling		
				Test time	% Reduction	CPU time	Test time	% Reduction	CPU time
a586710	7	800mW	14090716	7572316.31	46.26	1.99sec	7582339.33	46.19	2.7sec
h953	8	800mW	122636	60805.62	50.42	2.33sec	60805.62	50.42	1.67sec
d695	10	400mW	13301	5264.61	60.42	2.96sec	5210.1465	60.83	2.22sec
g1023	14	400mW	18084	8952.53	50.49	5.76sec	8898.817641	50.79	3.15sec
p34392	19	400mW	701684	340527.94	51.47	6.12sec	340508.9511	51.47	4.56sec
t512505	31	400mW	5344747	2953786.9	44.73	24.44sec	2940986.25	44.97	8.45sec
p93791	32	400mW	139008	74582.87	46.35	37.93sec	73681.67	46.99	13.94sec

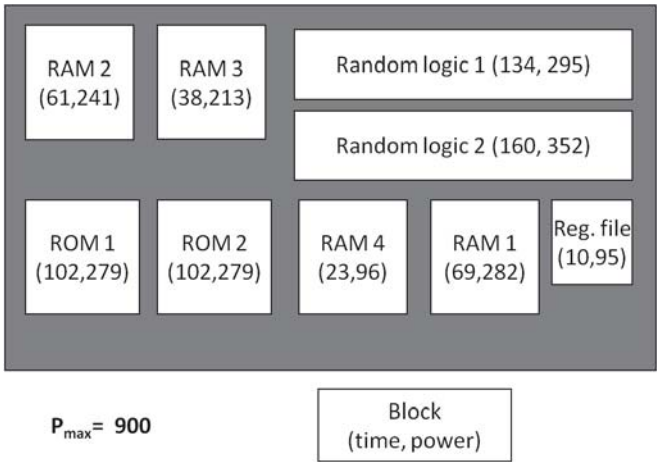

Fig. 7. The components of ASIC Z, and their test time (in arbitrary units) and test power (in mW) [35].

TABLE II
TEST TIMES (IN ARBITRARY UNITS) FOR ASIC Z SOC.

Source	Test time	% Difference**
Preemptive*	147.38	–
Non preemptive*	149.37	−1.35
[12]	262	−77.77
[27]	268.274	−82.03
[28]	155.1	−5.24

*this work
**Difference calculated with respect to lowest test time

solution. As mentioned earlier, heuristic algorithms perform much better in terms of CPU time as compared to exact methods such as ILP. Hence, the proposed heuristic solution would be applicable to a much larger range of SoCs than the ILP based optimization method.

V. CONCLUSION

We have presented heuristic methods for session-less SoC test scheduling. These methods employ dynamic scaling of supply voltage and test clock frequency to enhance the test schedule optimization and further reduce the overall test time. We have provided two heuristic approaches for the session-less scheduling, one preemptive and the other non-preemptive. In the preemptive algorithm, it is assumed that a test can be suspended and resumed at will whereas in the non-preemptive strategy, the tests run uninterrupted until completion. The heuristics were implemented on several ITC'02 benchmarks. Both methods achieve test time reductions of 45-60% in these benchmark SoCs. However, the preemptive algorithm requires more computation than the non-preemptive method since the preempted tests are added as new tests to the list of unscheduled tests.

The DVFS schemes are intended only for the reduction of test time and should not interfere with the fault coverage of the test. It has been shown that while V_{DD} does not affect stuck-open defects, it may affect the behavior of resistive opens [9], [22]. This does not, however, invalidate the proposed method but only restricts the available voltage range for the DVFS scheme. Hence, the contribution of this paper is a method with enough flexibility that user can select the range of voltages based on the defect coverage requirement. Most of the previously reported work is on "very low" voltage testing [5], [6], [9], [11], [22].

The heuristics presented in this work are simple in nature. We randomly pick and examine solutions from the solution space similar to a Monte-Carlo method. While the heuristics guarantee a feasible solution at every iteration, finding the optimal solution may require sampling many points from the solution space. The solution space also grows with the number of cores, thus requiring a larger number of simulations of the algorithm to find a near-optimal solution. A more efficient way would be to start at a random point in the solution space and proceed by picking better solutions among the neighboring points following a directed search. Ongoing research is focused on developing such a directed search heuristics for SoC test schedule optimization.

ACKNOWLEDGMENTS

This research is supported in parts by the National Science Foundation Grants CCF-1116213 and IIP-0738088. The authors would like to thank S. K. Millican and Professor K. K. Saluja, from University of Wisconsin, Madison for providing the power profiles for ITC'02 benchmarks.

REFERENCES

[1] "International Technology Roadmap for Semiconductors 2008 Update Overview." http://www.itrs.net/Links/2008ITRS/Update/.

[2] *ITC 2002 SOC Benchmarking Initiative*. http://www.extra.research.philips.com/itc02socbenchm.

[3] E. Beigné, F. Clermidy, S. Miermont, and P. Vivet, "Dynamic Voltage and Frequency Scaling Architecture for Units Integration Within a GALS NoC," in *Proc. Second ACM/IEEE International Symp. Networks-on-Chip*, 2008, pp. 129–138.

[4] K. Chakrabarty, "Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 10, pp. 1163–1174, Oct. 2000.

[5] J. T. Y. Chang and E. J. McCluskey, "Detecting Delay Flaws by Very-Low-Voltage Testing," in *Proc. International Test Conf.*, Oct. 1996, pp. 367–376.

[6] J. T. Y. Chang and E. J. McCluskey, "Quantitative Analysis of Very-Low-Voltage Testing," in *Proc. 14th IEEE VLSI Test Symp.*, 1996, pp. 332–337.

[7] R. M. Chou, K. K. Saluja, and V. D. Agrawal, "Scheduling Tests for VLSI Systems Under Power Constraints," *IEEE Trans. VLSI Systems*, vol. 5, no. 2, pp. 175–185, June 1997.

[8] V. R. Devanathan, C. P. Ravikumar, R. Mehrotra, and V. Kamakoti, "PMScan: A Power-Managed Scan for Simultaneous Reduction of Dynamic and Leakage Power During Scan Test," in *Proc. IEEE International Test Conf.*, Oct. 2007. Paper 13.3.

[9] P. Engelke, I. Polian, M. Renovell, B. Seshadri, and B. Becker, "The Pros and Cons of Very-Low-Voltage Testing: An Analysis based on Resistive Bridging Faults," in *Proc. 22nd IEEE VLSI Test Symp.*, 2004, pp. 171–178.

[10] P. Girard, N. Nicolici, and X. Wen, *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer, 2010.

[11] H. Hao and E. J. McCluskey, "Very-Low-Voltage Testing for Weak CMOS Logic ICs," in *Proc. International Test Conf.*, Oct. 1993, pp. 275–284.

[12] H. M. Harmanani and H. A. Salamy, "A Simulated Annealing Algorithm for System-on-Chip Test Scheduling With Power and Precedence Constraints," *International J. Computational Intelligence and Applications*, vol. 6, no. 4, pp. 511–530, 2006.

[13] Y. Huang, S. M. Reddy, W. T. Cheng, P. Reuter, N. Mukherjee, C. C. Tsai, O. Samman, and Y. Zaidan, "Optimal Core Wrapper Width Selection and SOC Test Scheduling Based on 3-D Bin Packing Algorithm," in *Proc. International Test Conf.*, 2002, pp. 74–82.

[14] V. Iyengar and K. Chakrabarty, "Precedence-Based, Preemptive and Power Constrained Test Scheduling for System-on-Chip," in *Proc. 19th IEEE VLSI Test Symp.*, 2001, pp. 368–374.

[15] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "On Using Rectangle Packing for SOC Wrapper/TAM Co-Optimization," in *Proc. 20th IEEE VLSI Test Symp.*, 2002, pp. 253–258.

[16] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-optimization for System-on-Chip," *J. Electronic Testing: Theory and Applications*, vol. 18, pp. 211–228, Mar. 2002.

[17] X. Kavousianos, K. Chakrabarty, A. Jain, and R. Parekhji, "Time-Division Multiplexing for Testing SoCs with DVS and Multiple Voltage Islands," in *Proc. European Test Symp.*, May 2012, pp. 1–6.

[18] S. Koranne, "Design of Reconfigurable Access Wrappers for Embedded Core Based SoC Test," *IEEE Trans. VLSI Systems*, vol. 11, no. 5, pp. 955–960, Oct. 2003.

[19] E. Larsson, *Introduction to Advanced System-on-Chip Test Design and Optimization*. Springer, 2005.

[20] E. Larsson and H. Fujiwara, "Power Constrained Preemptive TAM Scheduling," in *Proc. 7th IEEE European Test Workshop*, 2002, pp. 119–126.

[21] E. Larsson and Z. Peng, "An Integrated Framework for the Design and Optimization of SOC Test Solutions," *J. Electronic Testing: Theory and Applications*, vol. 18, pp. 385–400, 2002.

[22] J. C. M. Li, C. W. Tseng, and E. J. McCluskey, "Testing for Resistive Opens and Stuck Opens," in *Proc. International Test Conf.*, 2001, pp. 1049–1058.

[23] S. K. Millican and K. K. Saluja, "Linear Programming Formulations for Thermal-Aware Test Scheduling of 3D-Stacked Integrated Circuits," in *Proc. 21st Asian Test Symp.*, Nov. 2012, pp. 37–42.

[24] V. Muresan, X. Wang, V. Muresan, and M. Vladutiu, "A Comparison of Classical Scheduling Approaches in Power-Constrained Block-Test Scheduling," in *Proc. International Test Conf.*, 2000, pp. 882–891.

[25] J. Pouget, E. Larsson, Z. Peng, M. L. Flottes, and B. Rouzeyre, "An Efficient Approach to SoC Wrapper Design, TAM Configuration and Test Scheduling," in *Proc. IEEE European Test Workshop*, 2003, pp. 51–56.

[26] A. Sehgal, S. Bahukudumbi, and K. Chakrabarty, "Power-Aware SoC Test Planning for Effective Utilization of Port Scalable Testers," *ACM Trans. Design Automation of Electronic Systems*, vol. 13, no. 3, pp. 53–71, July 2008.

[27] V. Sheshadri, V. D. Agrawal, and P. Agrawal, "Optimal Power-Constrained SoC Test Schedules with Customizable Clock Rates," in *Proc. 25th IEEE System-on-Chip Conf.*, Sept. 2012, pp. 271–276.

[28] V. Sheshadri, V. D. Agrawal, and P. Agrawal, "Optimum Test Schedule for SoC with Specified Clock Frequencies and Supply Voltages," in *Proc. 26th International Conf. VLSI Design*, Jan. 2013, pp. 267–272.

[29] D. Truong, W. H. Cheng, T. Mohsenin, Z. Yu, A. T. Jacobson, G. Landge, M. J. Meeuwsen, C. Watnik, A. T. Tran, Z. Xiao, E. W. Work, J. W. Webb, P. V. Mejia, and B. M. Baas, "A 167-Processor Computational Platform in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1130–1144, 2009.

[30] P. Venkataramani and V. D. Agrawal, "Reducing Test Time of Power Constrained Test by Optimal Selection of Supply Voltage," in *Proc. 26th International Conf. VLSI Design*, Jan. 2013, pp. 273–278.

[31] P. Venkataramani and V. D. Agrawal, "Test Time Reduction Using Aynchronous Clocking," in *Proc. International Test Conf.*, Sept. 2013.

[32] P. Venkataramani, S. Sindia, and V. D. Agrawal, "A Test Time Theorem and Its Applications," in *Proc. 14th IEEE Latin-American Test Workshop*, Apr. 2013.

[33] P. Venkataramani, S. Sindia, and V. D. Agrawal, "Finding Best Voltage and Frequency to Shorten Power-Constrained Test Time," in *Proc. 31st IEEE VLSI Test Symp.*, Apr. 2013, pp. 19–24.

[34] D. Zhao and S. Upadhyaya, "Power Constrained Test Scheduling with Dynamically Varied TAM," in *Proc. 21st IEEE VLSI Test Symp.*, 2003, pp. 273–278.

[35] Y. Zorian, "A Distributed Control Scheme for Complex VLSI Devices," in *Proc. 11th IEEE VLSI Test Symp.*, Apr. 1993, pp. 4–9.

[36] W. Zou, S. M. Reddy, I. Pomeranz, and Y. Huang, "SOC Test Scheduling Using Simulated Annealing," in *Proc. 21st IEEE VLSI Test Symp.*, 2003, pp. 325–330.