

Test-Length and TAM Optimization for Wafer-Level Reduced Pin-Count Testing of Core-Based SoCs

Sudarshan Bahukudumbi, *Student Member, IEEE*, and Krishnendu Chakrabarty, *Fellow, IEEE*

Abstract—Wafer-level testing (wafer sort) is used in the semiconductor industry to reduce packaging and test cost. However, a large number of wafer-probe contacts lead to higher yield loss. Therefore, it is desirable that the number of chip pins contacted by tester channels during wafer sort be kept small to reduce the yield loss resulting from improper contacts. Since test time and the number of contacted chip pins are major practical constraints for wafer sort, not all scan-based digital tests can be applied to the die under test. We propose an optimization framework based on mathematical programming (integer linear programming, nonlinear programming, and geometric programming) and fast heuristic methods. This framework addresses test-access mechanism (TAM) optimization and test-length selection for wafer-level testing of core-based digital system-on-chips (SoCs). The objective here is to design a TAM architecture and determine test lengths for the embedded cores such that the overall SoC defect-screening probability at wafer sort is maximized. Defect probabilities for the embedded cores, obtained using statistical yield modeling, are incorporated in the optimization framework. Simulation results are presented for five of the ITC'02 SoC Test benchmarks.

Index Terms—Defect screening, reduced pin-count testing (RPCT), system-on-chip (SoC) test, wafer sort.

I. INTRODUCTION

THE CONSUMER electronics market is characterized by low product cost and decreasing profit margins. Market realities, as well as rapid advances in process technology and design tools, have led to system-on-chip (SoC) integrated circuits (ICs) that reduce the design cycle time and design cost. Short design cycle times are achieved by integrating a number of predesigned and preverified embedded cores into an SoC. While the testing of such core-based SoCs continues to be a major concern in the semiconductor industry [1], [2], a number of efficient solutions have recently been proposed for test-access mechanism (TAM) optimization and test scheduling [3]–[10]. The design of efficient TAM architectures and SoC test schedules are important problems that need to be addressed during system integration.

Test and packaging costs are major contributors to the overall product cost for an SoC [11]. Wafer-level testing, also referred to as wafer sort, is used to screen defective dies prior to

packaging, thereby reducing packaging cost and test time for the packaged ICs [12]–[14]. However, a large number of wafer-probe contacts lead to higher yield loss. Therefore, the number of chip pins contacted by tester channels during wafer sort is deliberately kept small to reduce the yield loss that results from improper contacts [15]. Since test time and the number of contacted chip pins are major practical constraints for wafer sort, not all scan-based digital tests can be applied to the die under test.

Reduced pin-count testing (RPCT) has been advocated as a design-for-test technique, particularly for use at wafer sort, to reduce the number of IC pins that needs to be contacted by the tester [15]–[19]. RPCT reduces the cost of test by enabling the reuse of old testers with limited channel availability. It also reduces the number of probe points required during wafer test; this translates to lower test cost as well as less yield loss issues arising from contact problems with the wafer probe.

The test cost for SoCs can also be reduced by using estimated defect probabilities for the embedded cores to guide test scheduling [20], [21]. These defect probabilities can be used to determine the order in which the embedded cores in the SoC are tested, as well as to identify the subsets of cores that are tested concurrently. The defect probabilities for the cores were assumed in [20] to be either known *a priori* or obtained by binning the failure information for each individual core over the product cycle [21]. In practice, however, short product cycles make defect estimation based on failure binning difficult. Moreover, defect probabilities for a given technology node are not necessarily the same for the next (smaller) technology node. Recently, a wafer-level defect-screening technique for core-based SoCs was presented in [22] and [23]. This approach is based on a combination of statistical yield modeling, to determine the defect probability for each core in the SoC, and integer linear programming (ILP) for optimization. It does not consider the problem of RPCT for wafer sort.

In this paper, we present an optimization framework that addresses TAM optimization and test-length selection for RPCT-based wafer-level testing of core-based digital SoCs. The objective here is to design a TAM architecture for wafer sort that utilizes a predesigned underlying TAM architecture for package test and also to determine test lengths for the embedded cores such that the overall SoC defect-screening probability at wafer sort is maximized. Defect probabilities for the embedded cores, obtained using statistical yield modeling, are incorporated in the optimization framework. The proposed method reduces packaging cost and the subsequent test time for the IC lot while efficiently utilizing available tester bandwidth at wafer sort. While an optimal test access architecture and test

Manuscript received May 7, 2008; revised July 23, 2008. Current version published December 17, 2008. This work was supported in part by the Semiconductor Research Corporation under Contract 2004-TJ1174 and in part by an equipment grant from Intel Corporation. An earlier version of this paper appeared in Proceedings of IEEE VLSI Design Conference, 2007, pp. 459–464. This paper was recommended by Associate Editor R. D. Blanton.

S. Bahukudumbi is with Intel Corporation, Hillsboro, OR 97124 (e-mail: spb@ee.duke.edu).

K. Chakrabarty is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: krish@ee.duke.edu).

Digital Object Identifier 10.1109/TCAD.2008.2009150

schedule can also be developed for wafer sort, we assume that these test planning problems are best tackled for package test, simply because the package test time is higher.

We present three techniques for test-length selection and TAM optimization. The first technique is based on the formulation of a nonlinear integer programming model, which can be subsequently linearized and solved using standard ILP tools. While this approach leads to a thorough understanding of the optimization problem, it does not appear to be scalable for large SoCs. We therefore describe a second method that enumerates all possible valid TAM partitions, and then uses the ILP model presented in [22] to derive test lengths to maximum defect screening at wafer sort. This enumerative procedure allows an efficient search of a large solution space, thereby resulting in significantly lower computation time than that needed for the first method. The third method relies on a heuristic based on geometric programming (GP). Simulation results on TAM optimization and test-length selection are presented for five of the ITC'02 SoC Test benchmarks [24].

The remainder of this paper is organized as follows. In Section II, we describe how the defect-screening probabilities of the cores and the SoC are determined using the approach presented in [22]. Section III formulates the problem of TAM optimization and test-length selection. An integer programming model is presented to solve this problem. Simulation results for two ITC'02 SoC Test benchmarks are presented in Section III. Section IV presents the second optimization method based on the enumeration of TAM partitions. Simulation results for five of the ITC'02 SoC Test benchmarks are presented. Section V presents the GP-based heuristic and compares the defect-screening probabilities obtained using the different techniques. In Section VI, a nonlinear programming (NLP) solver is used to determine the error introduced by the linearization technique used in Section III. Finally, Section VII concludes this paper.

II. DEFECT-SCREENING PROBABILITY

In this section, we briefly describe how the defect-screening probability for the SoC can be determined using the method presented in [22]. The defect probabilities for the embedded cores are obtained using the yield modeling technique presented in [22]. Let us now define the following statistical events for core i .

- 1) A_i : the event that the core has a fault. The probability associated with this event is determined from the statistical yield model described in [22].
- 2) B_i : the event that the tests applied to core i do not produce an incorrect response. \bar{A}_i and \bar{B}_i represent events that are complementary to events A_i and B_i , respectively.

Two important conditional probabilities associated with the aforementioned events are yield loss and test escape, denoted by $\mathcal{P}(\bar{B}_i | \bar{A}_i)$ and $\mathcal{P}(B_i | A_i)$, respectively. By using a basic identity of probability theory, we can derive the probability that the test applied to core i detects a defect

$$\mathcal{P}(\bar{B}_i) = \mathcal{P}(\bar{B}_i | A_i) \cdot \mathcal{P}(A_i) + \mathcal{P}(\bar{B}_i | \bar{A}_i) \cdot \mathcal{P}(\bar{A}_i). \quad (1)$$

Due to SoC test time and TAM width constraints during wafer-level testing, only a subset of the pattern set can be

applied to any core i , i.e., if the complete test suite for the SoC contains p_i scan patterns for core i , only $p_i^* \leq p_i$ patterns can be actually applied to it during wafer sort. Let $f_{c_i}(p_i^*)$ be the fault coverage for core i with p_i^* test patterns. Let us now assume that the yield loss is γ_i , the test escape is β_i , and the probability that core i has a defect is θ_i . By using these variables, we can rewrite (1) as

$$\mathcal{P}(\bar{B}_i) = f_{c_i}(p_i^*) \cdot \theta_i + \gamma_i \cdot (1 - \theta_i). \quad (2)$$

Similarly, we can rewrite $\mathcal{P}(B_i)$ as follows:

$$\mathcal{P}(B_i) = 1 - \mathcal{P}(\bar{B}_i) = \theta_i \cdot \beta_i + (1 - \gamma_i) \cdot (1 - \theta_i). \quad (3)$$

The defect-screening probability \mathcal{P}_S for an SoC with N embedded cores is given by $\mathcal{P}_S = 1 - \prod_{i=1}^N \mathcal{P}(\bar{B}_i)$.

III. REDUCED PIN-COUNT TEST-LENGTH AND TAM OPTIMIZATION PROBLEM

In practice, the TAM bitwidth used at wafer sort is considerably less than the maximum available TAM bitwidth for package test. This is because of the likelihood of yield loss at wafer sort due to improper touchdowns/probe-pin contacts by the wafer probe [12]–[14]. RPCT methods are therefore desirable for wafer sort. Reconfigurable TAM architectures such as in [25] and [26] can be useful for wafer sort; these techniques impose considerable area and performance overhead due to the need for wrapper multiplexing. In [22], it is assumed that the TAM architecture is fixed and optimized for package test; the same architecture is used for wafer-level test optimization. Therefore, the authors in [22] make the unrealistic assumption that the TAM bitwidth at wafer sort is the same as that for package test. We next formulate the test-length selection and TAM optimization problems for RPCT.

A. Test Data Serialization

Suppose that core i is accessed from the SoC pins for package test using a TAM of width w_i (bits). Let us assume that for RPCT-based wafer sort, the TAM width for core i is constrained to be w_i^* bits, where $w_i^* < w_i$. In order to access core i using only w_i^* bits for wafer sort, the predesigned TAM architecture for package test needs to be appropriately modified.

Fig. 1(a) shows a wrapped core that is connected to a 4-b-wide TAM width ($w_i = 4$). For the same wrapped core, Fig. 1(b) shows a modified test access design that allows an RPCT-based wafer-level test with $w_i^* = 2$. For wafer sort in this example, the lines TAM_{out}[0] and TAM_{out}[2] are not used. In order to ensure efficient test access architecture for wafer sort, serial-to-parallel conversion of the test data stream is necessary at the wrapper inputs of the core. A similar parallel-to-serial conversion is necessary at the wrapper outputs of the cores. Boundary input cells $BIC[0], \dots, BIC[3]$ and boundary output cells $BOC[0], \dots, BOC[3]$, which can operate in both a parallel load and a serial shift mode, are added at the I/Os of the wrapped core. Multiplexers are added on the input side of the core to enable the use of a smaller number of TAM lines for wafer sort. A global select signal \overline{PT}/WS is used to choose

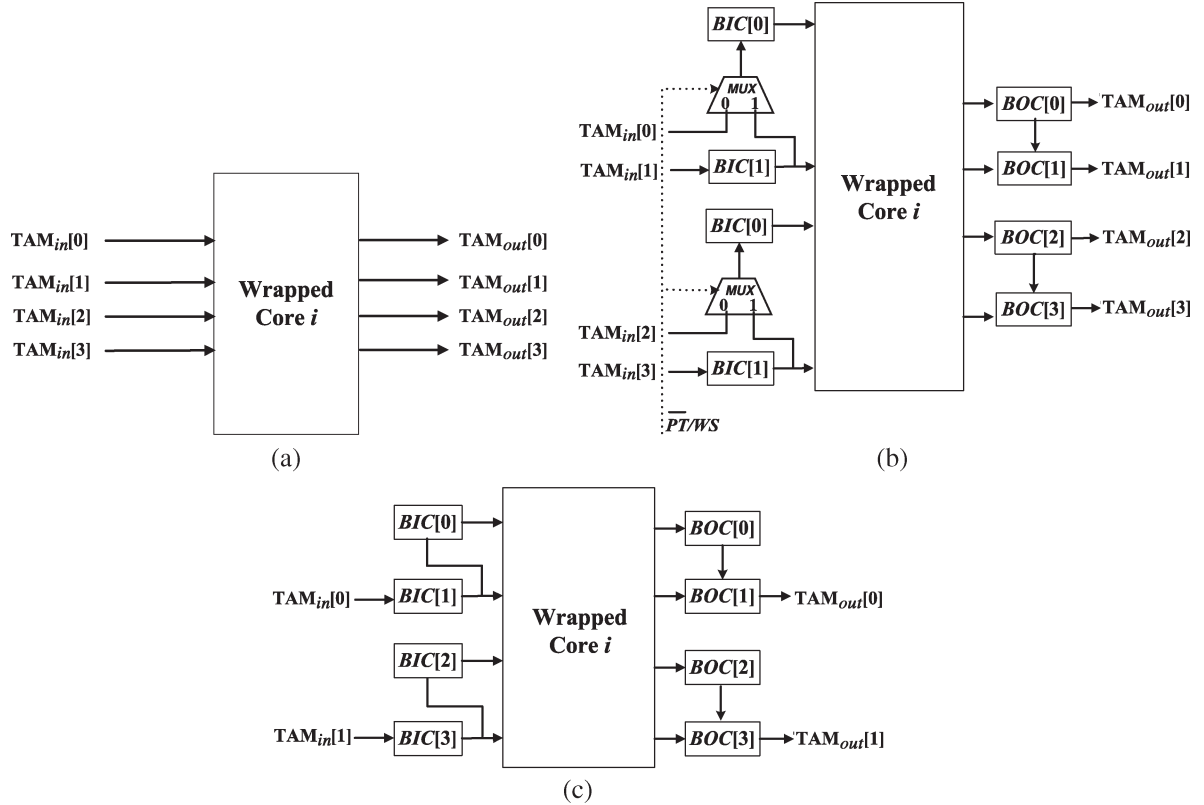


Fig. 1. (a) Accessing a wrapped core for package test only. (b) TAM design that allows RPCT-based wafer sort using a predesigned wrapper/TAM architecture. (c) Wrapper use for wafer test.

either the package test mode ($\overline{PT}/WS = 0$) or the wafer-sort mode ($\overline{PT}/WS = 1$). For the output side, the multiplexers are not needed; the test response can be serially shifted out to the TAM, while the next pattern is serially shifted into the boundary input cells. Fig. 1(c) shows the test data path without the multiplexers during the wafer-sort test mode. Note that the aforementioned design is fully compliant with the IEEE 1500 standard [27] because no modifications are made to the standard wrapper cells.

We next explain how the test time for core i is affected by the serialization process. Let $T_i(j)$ be the total testing time (in clock cycles) for core i if it is placed on TAM partition j of the SoC. Let $w_i(j)$ be the width of TAM partition j in the predesigned TAM architecture. At the wafer level, if only w_i^* bits are available for TAM partition j , we assume, as in [28] for hierarchical SoC testing, that the w_i lines are distributed equally into w_i^* parts. Thus, the wafer-level testing time for core i on TAM partition j equals $\lceil (w_i(j)/w_i^*(j)) \rceil \cdot T_i(j)$ clock cycles. In the example of Fig. 1(b), the test time for core i due to serialization is $T_i^*(j) = T_i(j) \cdot (4/2)$. Note that other TAM serialization methods can also be used for wafer sort. While TAM serialization can be integrated in an overall optimization problem, it is not considered here for the sake of simplicity.

B. Test-Length and TAM Optimization Problem: \mathcal{P}_{TLTWS}

Let the upper limit on the test time for an SoC at wafer sort be T_{\max} (clock cycles). This upper limit on the scan test time at wafer sort is expected to be a fraction of the scan test time

T_{SoC} (clock cycles) for package test, as determined by the TAM architecture and test schedule. We assume a fixed-width TAM architecture as in [5], where the top-level TAM is divided into several TAM partitions. This architecture implies that the total test time on each TAM partition must not exceed T_{\max} .

If the internal details of the embedded cores are available to the system integrator, fault simulation can be used to determine the fault coverage for various values of p_i^* , i.e., the number of patterns applied to the cores during wafer sort. Otherwise, we model the relationship between fault coverage and the number of patterns with an exponential function. It is well known in the testing literature that the fault coverage for stuck-at faults increases rapidly initially as the pattern count increases, but it flattens out when more patterns are applied to the circuit under test [29], [30]. In our work, without loss of generality, we use the normalized function $fc_i(p_i^*) = (\log_{10}(p_i^* + 1) / \log_{10} p_i^*)$ to represent this relationship. A similar relationship was used in [30]. We have verified that this empirical relationship matches the “fault coverage curve” for the ISCAS benchmark circuits.

Let $\epsilon_i(P_i^*)$ be the defect-escape probability for core i when p_i^* patterns are applied to it. This probability can be obtained using (3) as a function of the test escape β_i and the probability θ_i that the core is faulty. The value of θ_i for each core in the SoC is obtained using the procedure described in [22]. Let us now consider an SoC with a top-level TAM width of W bits and suppose that it has B TAM partitions of widths w_1, w_2, \dots, w_B . For a given value of the maximum wafer-level TAM width W^* , we need to determine appropriate TAM partitions of widths $w_1^*, w_2^*, \dots, w_B^*$ such that $w_i^* \leq w_i, 1 \leq i \leq B$,

and $w_1^* + w_2^* + \dots + w_B^* = W^*$. The optimization problem $\mathcal{P}_{\text{TLTWS}}$ can now be formally stated as follows.

Problem $\mathcal{P}_{\text{TLTWS}}$: Given a predesigned TAM architecture for a core-based SoC, the defect probabilities for each core in the SoC, maximum available test bandwidth at wafer sort W^* , and the upper limit on the test time for the SoC at wafer sort T_{MAX} , determine the following: 1) the total number of test patterns to be applied to each core and 2) the (reduced) TAM width for each partition, such that the overall testing time on each TAM partition does not exceed the upper bound T_{max} and the defect-screening probability $P(\bar{B}_i)$ for the SoC is maximized.

The objective function for the optimization problem is as follows:

$$\text{Maximize } Y = \prod_{i=1}^N 1 - \mathcal{P}(B_i)$$

where the number of cores in the SoC is N . We next introduce the indicator binary variable δ_{ij} , $1 \leq i \leq N$, $0 \leq j \leq p_i$, which ensures that exactly one test length is selected for each core. It is defined as follows:

$$\delta_{ij} = \begin{cases} 1, & \text{if } p_i^* = j \\ 0, & \text{otherwise} \end{cases}$$

where $\sum_{j=1}^{p_i} \delta_{ij} = 1$. The defect-escape probability ϵ_i^* for core i is given by $\epsilon_i^* = \sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j)$. We next reformulate the objective function to make it more amenable for further analysis. Let $\mathcal{F} = \ln(Y)$. We therefore get

$$\begin{aligned} \mathcal{F} &= \ln(Y) \\ &= \ln \left(\prod_{i=1}^N 1 - \mathcal{P}(B_i) \right) \\ &= \sum_{i=1}^N \ln \left(1 - \sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right). \end{aligned}$$

We next use the Taylor series expansion $\ln(1-x) = -(x + x^2/2 + x^3/3 + \dots)$ and ignore the second-order and higher order terms [31]. This approximation is justified if the defect-escape probability for core i is much smaller than one. While this is usually the case, occasionally, the defect-escape probability is large; in such cases, the optimality claim is valid only in a limited sense. The simplified objective function is given by

$$\text{Maximize } \mathcal{F} = \sum_{i=1}^N \left(\sum_{j=1}^{p_i} -(\delta_{ij} \epsilon_i(j)) \right).$$

In other words, the objective function can be stated as

$$\text{Minimize } \mathcal{F} = \sum_{i=1}^N \left(\sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right). \quad (4)$$

The constraint on the overall test time at wafer sort is given by T_{max} which is a fraction of the overall test time of the SoC

Minimize $\mathcal{F} = \sum_{i=1}^N \left(\sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right)$, subject to :

- 1) $\sum_{i=1}^{n_x} \left(\sum_{j=1}^{p_i} \sum_{k=1}^{w_i} \delta_{ij} T_i(j) \lambda_{jk} \lceil w_i \cdot k \rceil \right) \leq T_{\text{max}}; \forall x, 1 \leq x \leq B$
- 2) $\sum_{j=1}^{p_i} \delta_{ij} = 1; \forall i, 1 \leq i \leq N$
- 3) $\sum_{k=1}^{w_i} \lambda_{ik} = 1; \forall i, 1 \leq i \leq N$
- 4) $\sum_{k=1}^{w_i} k \cdot \lambda_{ik} = W_x^*; \forall \text{Core } i \in A_j$
- 5) $\sum_{x=1}^B W_x^* \leq W^*$
- 6) $\delta_{ij} + \lambda_{ik} - 1 \leq u_{ijk}; \forall i, j, k$
- 7) $\delta_{ij} + \lambda_{ik} \geq 2 \cdot u_{ijk}; \forall i, j, k$

/* Constants : $\epsilon_i(j), T_{\text{max}}$ */

/* Variables : $\delta_{ij}, \lambda_{ik}, u_{ijk}; 1 \leq i \leq N, 0 \leq j \leq p_i$ */

Fig. 2. ILP model for $\mathcal{P}_{\text{TLTWS}}$.

(T_{SoC}). Due to serialization, the testing time for core i on TAM partition j is given by $\lceil w_i(j)/w_i^*(j) \rceil T_i(j)$ [28]. Therefore, the test time of core i when it is tested with a reduced bitwidth of w_i^* is given by

$$T_i^* = \sum_{j=1}^{p_i} \delta_{ij} T_i(j) \left\lceil \frac{w_i(j)}{w_i^*(j)} \right\rceil. \quad (5)$$

Let us now define a second binary indicator variable λ_{ik} to ensure that every core in the SoC is tested using a single TAM width; this variable can be defined as follows:

$$\lambda_{ik} = \begin{cases} 1, & \text{if } w_i^* = 1/k \\ 0, & \text{otherwise.} \end{cases}$$

It can be inferred from the aforementioned definition that $\sum_{k=1}^{w_i} \lambda_{ik} = 1$, and (5) can now be represented as $T_i^*(j) = \sum_{j=1}^{p_i} \sum_{k=1}^{w_i} \delta_{ij} T_i(j) \lambda_{jk} \lceil w_i \cdot k \rceil$. The nonlinear term in the constraint $\delta_{ij} \cdot \lambda_{ik}$ can be replaced with a new binary variable u_{ijk} by introducing two additional constraints

$$\delta_{ij} + \lambda_{ik} \leq u_{ijk} + 1 \quad (6)$$

$$\delta_{ij} + \lambda_{ik} \geq 2 \cdot u_{ijk}. \quad (7)$$

A constraint to ensure that every core in a TAM partition is tested with the same TAM width W_x^* is also necessary and can be represented as in (8). The variable A_j denotes the set of cores that are assigned to TAM partition j . The constraint must be satisfied for every core in A_j .

$$\sum_{k=1}^{w_i} k \cdot \lambda_{ik} = W_x^*. \quad (8)$$

The complete ILP model is shown in Fig. 2. The number of variables and constraints in the ILP model determines the complexity of the problem. The number of variables in our ILP model is $\sum_{i=1}^N (p_i + w_i + p_i \cdot w_i)$, and the number of constraints is $2 \cdot N + 2 \sum_{i=1}^N (p_i \cdot w_i) + B + 1$.

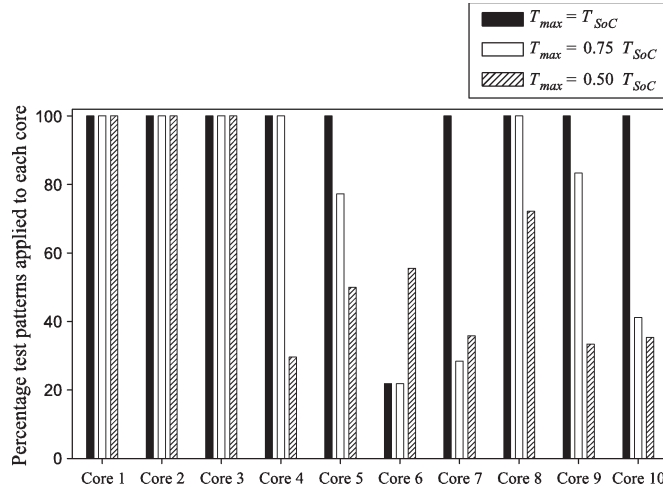


Fig. 3. Percentage of test patterns applied to each core in d695 when $W^* = 16$ and $W = 32$.

C. Experimental Results: \mathcal{P}_{TLTWS}

- 1) Given values of W^* and T_{\max} relative to T_{SoC} , the percentage of test patterns that must be applied for each individual core to maximize the defect-screening probability for the SoC.
- 2) The values of TAM partition widths $w_1^*, w_1^*, \dots, w_B^*$ such that $w_1^* + w_1^* + \dots + w_B^* = W^*$.
- 3) The relative defect-screening probability \mathcal{P}_S^r for each core in an SoC, where $\mathcal{P}_S^r = \mathcal{P}_S / \mathcal{P}_S^{100}$ and \mathcal{P}_S^{100} is the defect-screening probability if all 100% of the patterns are applied per core.
- 4) The relative defect-screening probability for the SoC obtained using the ILP model.

We first present results on the number of patterns determined for the cores. The fault coverage data for the d695 benchmark circuit are obtained using a commercial automatic test-pattern generation tool. The results for the d695 benchmark SoC are shown in Fig. 3 for three values of T_{\max} : T_{SoC} , $0.75T_{SoC}$, and $0.5T_{SoC}$. The fraction of test patterns applied per core is found to be different in each case to maximize the defect-screening probability. Results are reported only for $W^* = 16$ and $W = 32$; similar plots are obtained for different values of W^* and W . Fig. 4 shows the defect-screening probabilities for the cores in the d695 benchmark for the aforementioned test case.

We summarize the results for two benchmark SoCs in Table I for three different values of W^* and $W = 32$. The relative defect-screening probabilities \mathcal{P}_S and TAM partition widths to be used at wafer sort, obtained using \mathcal{P}_{TLTWS} , are enumerated for both benchmark SoCs. The ILP-based technique takes up to 3 h of CPU time on a 2.4-GHz AMD Opteron processor with 4 GB of memory for d695, when $W^* = 16$ and $W = 32$. The results show that a significant portion of the faulty dies can be screened at wafer sort using the proposed technique.

IV. ENUMERATION-BASED TAM WIDTH AND TEST-LENGTH SELECTION: $\mathcal{P}_{e-TLTWS}$

The ILP-based approach in Section III is efficient only for small SoCs. However, due to its large size, it may not scale well

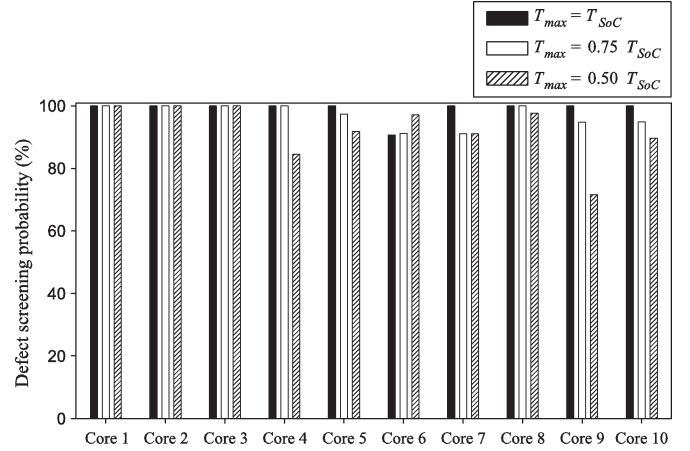


Fig. 4. Relative defect-screening probabilities for the individual cores in d695 when $W^* = 16$ and $W = 32$.

for SoCs with a large number of cores. It is therefore necessary to develop an alternative technique that can handle larger SoC designs. We next propose an efficient approach based on a combination TAM partition-width enumeration and ILP.

Our enumeration approach is based on the ‘‘odometer’’ principle used in a car odometer. Each digit of a car odometer here corresponds to a TAM partition width at wafer sort. Each digit can take values between 1 and the upper limit fixed by the TAM architecture designed for package test. We first increase the least significant digit if possible and, next, roll the digit over to one and increase the next least significant digit. The implementation of the enumeration approach for determining the optimal TAM partition widths and test lengths can be done using the following sequence of procedures.

- 1) Given the number of TAM partitions B and an upper limit on the maximum TAM width W^* , we first enumerate all possible TAM partition combinations. This enumeration can be done following the principle of a B -bit odometer, where each bit corresponds to the width of each TAM partition. The odometer resets to one as opposed to zero in the case of a conventional odometer (the maximum value that the i th bit can take before a reset is w_i). At every increment in the odometer, we check whether $\sum_{i=1}^B w_i^* = W^*$. All possible TAM partitions that meet the aforementioned condition are recorded as a valid partition. We illustrate the aforementioned enumeration procedure with a small example. Let us consider an SoC whose TAM architecture is fixed and designed for 5 b and partitioned into three TAM partitions of widths 2, 3, and 1, respectively, for package test. All possible TAM enumerations for the aforementioned partitions are $\{\langle 1, 1, 1 \rangle, \langle 1, 2, 1 \rangle, \langle 1, 3, 1 \rangle, \langle 2, 1, 1 \rangle, \langle 2, 2, 1 \rangle, \langle 2, 3, 1 \rangle\}$. These partitions are explicitly enumerated in the proposed method. If we consider W^* to be 4, then the valid TAM partitions are $\{\langle 1, 2, 1 \rangle, \langle 2, 1, 1 \rangle\}$.
- 2) For each valid TAM partition calculated in Step 1), we apply the test-length selection procedure \mathcal{P}_{TLS} from [22]. We calculate the defect-screening probability for the SoC from the results obtained using \mathcal{P}_{TLS} .

TABLE I
RELATIVE DEFECT-SCREENING PROBABILITIES OBTAINED USING $\mathcal{P}_{\text{TLTWS}}(W = 32)$

SoC	W^*	$T_{\max} = T_{\text{SoC}}$		$T_{\max} = 0.75T_{\text{SoC}}$		$T_{\max} = 0.5T_{\text{SoC}}$	
		Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability
d695	8	(5,1,2)	0.3982	(5,1,2)	0.2907	(4,1,3)	0.1058
	12	(5,1,6)	0.4426	(5,1,6)	0.3272	(5,3,4)	0.2631
	16	(10,3,3)	0.9064	(10,3,3)	0.7279	(10,3,3)	0.4306
a586710	8	(1,4,3)	0.7294	(1,4,3)	0.6142	(1,4,3)	0.4623
	12	(1,7,4)	0.7519	(1,7,4)	0.6682	(1,7,4)	0.5191
	16	(1,8,7)	0.7621	(1,8,7)	0.6682	(1,8,7)	0.5191

Minimize $\mathcal{F} = \sum_{i=1}^N \left(\sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right)$ subject to:

1. $\sum_{i=1}^{p_i} \delta_{ij} = 1, 1 \leq i \leq N, 0 \leq j \leq p_i$
2. $\sum_{\text{Core}_i \in A_j} T_i^* \leq T_{\max}, 1 \leq j \leq B$
3. $\delta_{ij} = 0$ or $1, 1 \leq i \leq N, 0 \leq j \leq p_i$

/* Constants : $\epsilon_i(j), T_i(j)$ */

/* Variables : $\delta_{ij}, T_i^*, 1 \leq i \leq N, 0 \leq j \leq p_i$ */

Fig. 5. ILP model for \mathcal{P}_{TLS} [22].

The objective function for \mathcal{P}_{TLS} is the same as (4). Let $T_i(j)$ be the test time for core i when j patterns are applied to it. For a given core i on a TAM partition of width w_B , we use the *design wrapper* technique from [5] to determine the longest scan in (out) chains of length $s_i(s_o)$ of the core on that TAM partition. The value of $T_i(j)$ can be determined using the formula $T_i(j) = (1 + \max\{s_i, s_o\} \cdot j + \min\{s_i, s_o\})$ [5]. The test time T_i^* for core i is therefore given by $T_i^* = \sum_{j=1}^{p_i} \delta_{ij} T_i(j)$. Let A_j denote the set of cores that are assigned to TAM partition j . We must ensure that $\sum_{\text{Core}_i \in A_j} T_i^* \leq T_{\max}, 1 \leq j \leq B$. The complete ILP model is shown in Fig. 5

- 3) If the defect-screening probability of the new partition is greater than the previous partition, we store it as the new defect-screening probability and store this partition as the current optimal partition.
- 4) We repeat this procedure until all possible TAM partitions are enumerated.

The experimental results obtained using the $\mathcal{P}_{\text{e-TLTWS}}$ procedure are summarized in Table II. The results are represented in a similar fashion as in Table I. The values of the defect-screening probabilities \mathcal{P}_S for five benchmark circuits [24], as well as the recommended TAM partition widths for wafer sort, are shown in the table. The number of patterns determined using $\mathcal{P}_{\text{e-TLTWS}}$ for the p34392 SoC is shown in Fig. 6. The results are shown for three values of T_{\max} : T_{SoC} , $0.75T_{\text{SoC}}$, and $0.5T_{\text{SoC}}$. Results are reported only for $W^* = 16$ and $W = 32$; similar plots are obtained for a range of values of W^* and W . Fig. 7 shows the relative defect-screening probabilities for the cores in the p34392 benchmark for the aforementioned test case. The heuristic method results in lower defect-screening probability for most cases compared with the ILP-based method; for higher values of W^* , the difference in defect-

screening probability between the two methods decreases. The computation time for the largest benchmark SoC p93791 was only 4 min; hence, this approach is suitable for large designs.

V. TAM WIDTH AND TEST-LENGTH SELECTION BASED ON GP

GP problems are convex optimization problems that are similar to linear programming problems [32]. A GP is a mathematical problem of the form

$$\text{Minimize } f_0(x)$$

$$\text{subject to } f_i(x) \leq 1, \quad i = 1, \dots, m$$

$$g_i(x) = 1, \quad i = 1, \dots, p$$

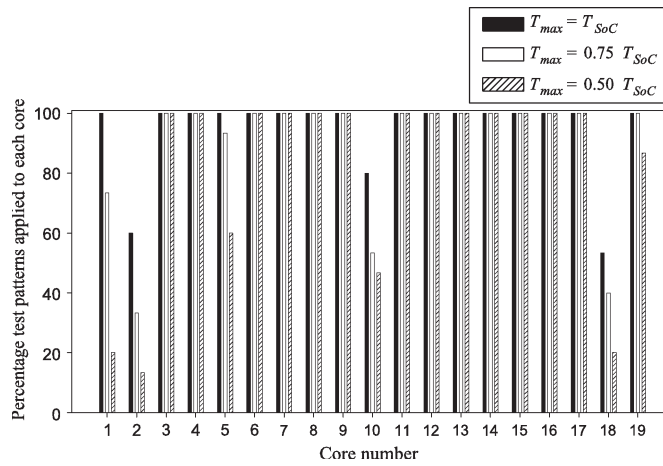
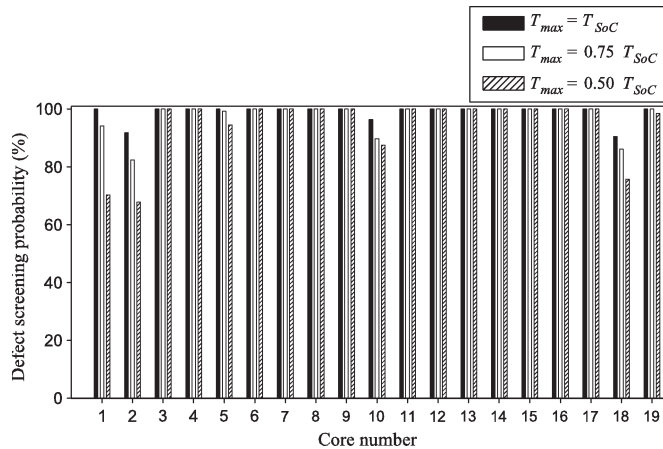
where f_i 's are posynomial functions, g_i 's are monomials, and x_i 's are the optimization variables; it is implicitly assumed that the optimization variables are positive, i.e., $x_i > 0$ [32]. Mixed-integer GPs (MIGPs) are a class of problems that are hard to solve [32]. The problem $\mathcal{P}_{\text{TLTWS}}$ can be modeled as an MIGP problem. We employ a heuristic method to solve the MIGP problem $\mathcal{P}_{\text{gp-TLTWS}}$ for test-length and TAM width selection. By using heuristic methods, approximate solutions can be found in a reasonable amount of time; the optimality of the solution, however, cannot be guaranteed. Before we describe the GP-based heuristic method, we need to modify the objective function to make it amenable for further analysis. The objective of $\mathcal{P}_{\text{TLTWS}}$ is to maximize the defect-screening probability, $\mathcal{P}_S = 1 - \prod_{i=1}^N \mathcal{P}(B_i)$. This is equivalent to the following minimization-based objective function:

$$\text{Minimize } \mathcal{G} = \prod_{i=1}^N \left(\sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right).$$

The constraints for the optimization problem described in Section III can be easily modified for use in the MIGP problem. The complete MIGP problem for $\mathcal{P}_{\text{TLTWS}}$ is shown in Fig. 8. We use GP relaxation to transform the MIGP problem to a general GP problem that can be solved using commercial tools [33]. To obtain an approximate solution of the MIGP problem, the MIGP is relaxed to a GP and solved using commercial tools [33]; the result obtained in this way is a lower bound on the optimal value of the objective function for MIGP. The values of the variables obtained after relaxation are then simply rounded toward the nearest integer. The heuristic then iteratively reassigns the values of the variables such that the constraints are satisfied while maximizing the defect-screening probability for

TABLE II
 RELATIVE DEFECT-SCREENING PROBABILITIES OBTAINED USING $\mathcal{P}_{e-}TLTWS$

SoC	W^*	$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.5T_{SoC}$	
		Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability
a586710	8	(1,5,2)	0.5732	(1,5,2)	0.5341	(1,5,2)	0.3319
	12	(2,6,4)	0.7014	(2,6,4)	0.5789	(2,6,4)	0.4449
	16	(2,9,5)	0.7118	(2,9,5)	0.5837	(2,9,5)	0.4580
d695	8	(5,1,2)	0.5392	(4,2,2)	0.5471	(5,1,2)	0.3102
	12	(7,2,3)	0.8139	(7,3,2)	0.5542	(7,2,3)	0.4116
	16	(9,2,5)	0.8543	(9,3,4)	0.7022	(8,2,6)	0.5231
p34392	8	(3,3,2)	0.3385	(3,2,3)	0.2275	(3,2,3)	0.1110
	12	(4,5,3)	0.6382	(4,4,4)	0.4360	(4,4,4)	0.2180
	16	(6,7,3)	0.8010	(5,7,4)	0.5968	(4,7,5)	0.2948
p22810	8	(3,3,2)	0.1331	(3,3,2)	0.0580	(3,3,2)	0.0098
	12	(4,6,2)	0.1891	(4,5,3)	0.1800	(3,6,3)	0.0333
	16	(5,6,5)	0.6186	(6,6,4)	0.3841	(6,6,4)	0.1495
p93791	8	(2,4,2)	0.0606	(2,4,2)	0.0165	(2,4,2)	0.0050
	12	(3,6,3)	0.2228	(3,7,2)	0.0949	(3,7,2)	0.0189
	16	(4,8,4)	0.5018	(4,8,4)	0.2201	(4,8,4)	0.0615


 Fig. 6. Percentage of test patterns applied to each core in p34392 when $W^* = 16$ and $W = 32$.

 Fig. 7. Relative defect-screening probabilities for the individual cores in p34392 when $W^* = 16$ and $W = 32$.

the SoC. The heuristic used to solve $\mathcal{P}_{gp-}TLTWS$ consists of the following steps.

- 1) In the first step of this procedure, we relax the MIGP for \mathcal{P}_{TLTWS} to a GP problem. The relaxation essentially

Minimize $\mathcal{G} = \prod_{i=1}^N \left(\sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right)$, subject to :

- 1) $\frac{\sum_{i=1}^{n_x} \left(\sum_{j=1}^{p_i} \sum_{k=1}^{w_i} \delta_{ij} T_i(j) \lambda_{jk} [w_i \cdot k] \right)}{T_{max}} \leq 1; \forall x, 1 \leq x \leq B$
- 2) $\sum_{j=1}^{p_i} \delta_{ij} = 1; \forall i, 1 \leq i \leq N$
- 3) $\sum_{k=1}^{w_i} \lambda_{ik} = 1; \forall i, 1 \leq i \leq N$
- 4) $\frac{\sum_{k=1}^{w_i} k \cdot \lambda_{ik}}{W^*} = 1; \forall Core_i \in A_j$
- 5) $\frac{\sum_{x=1}^B W_x}{W^*} = 1$

/* Constants : $\epsilon_i(j), T_{max}, W^*$ */

/* Variables : $\delta_{ij}, \lambda_{ik}; 1 \leq i \leq N, 0 \leq j \leq p_i$ */

 Fig. 8. GP model for \mathcal{P}_{TLTWS} .

means that the binary indicator variables used in the optimization problem can take noninteger values.

- 2) We then use commercial tools [33] to solve the relaxed MIGP problem. The resulting values of the indicator variables δ_{ij} are sorted for each core i . The highest value of δ_{ij} for each core is rounded to unity, while the remaining variables are rounded down to zero.
- 3) The procedure then assigns the smallest value of TAM width to each core in the SoC, i.e., $\lambda_{i1} = 1, \forall i$. For the smallest value of TAM widths assigned to the cores, the test time for each TAM partition is calculated.
- 4) The procedure then iteratively assigns additional TAM width to the TAM partition with the maximum test time. This is repeated until $\sum_{i=1}^B w_i^* = W^*$.
- 5) Once the TAM widths for RPCT are determined, we check to determine if $\sum_{Core_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$. If violations in test time constraints are observed, we identify a core in each TAM partition for which a reduction in the number of applied patterns results in a minimal decrease in the overall defect-screening probability. For each TAM partition $j, 1 \leq j \leq B$, where a violation in test time is observed, we chose a particular $Core_i \in A_j$ such that a decrease in the number of applied patterns Δp_i^* results in a minimal decrease in $\epsilon_i(p_i^*)$. We consider different values of Δp_i^* in the range $1 \leq \Delta p_i^* \leq 5$ in our

TABLE III
RELATIVE DEFECT-SCREENING PROBABILITIES OBTAINED USING THE GP-BASED HEURISTIC METHOD

SoC	W^*	$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.5T_{SoC}$	
		Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability
a586710	8	(4,2,2)	0.7226	(4,1,3)	0.5833	(4,1,3)	0.4594
	12	(6,3,3)	0.7446	(6,3,3)	0.6391	(6,3,3)	0.5138
	16	(9,3,4)	0.7582	(8,3,5)	0.6435	(8,3,5)	0.5120
d695	8	(4,1,3)	0.5027	(4,1,3)	0.5288	(4,1,3)	0.3014
	12	(6,1,5)	0.7962	(6,2,4)	0.5532	(6,2,4)	0.3961
	16	(8,2,6)	0.8420	(8,2,6)	0.6931	(8,2,6)	0.5090
p34392	8	(3,4,1)	0.3440	(3,3,2)	0.2330	(3,3,2)	0.1150
	12	(3,4,4)	0.6473	(3,4,4)	0.4455	(3,4,4)	0.2251
	16	(6,6,4)	0.8072	(6,5,5)	0.6081	(6,5,5)	0.3002
p22810	8	(3,2,3)	0.1346	(3,1,4)	0.0598	(3,1,4)	0.0100
	12	(4,5,3)	0.1911	(4,6,2)	0.1848	(4,6,2)	0.0322
	16	(4,5,7)	0.6246	(5,4,7)	0.3892	(5,4,7)	0.1508
p93791	8	(4,3,1)	0.0620	(4,3,1)	0.0170	(4,2,2)	0.0051
	12	(2,6,4)	0.2285	(2,6,4)	0.0977	(2,6,4)	0.0193
	16	(3,7,6)	0.5119	(3,8,5)	0.2216	(3,8,5)	0.0619

experiments and choose the value that results in maximum defect-screening \mathcal{P}_S^r for the SoC. This procedure searches for a core in each TAM partition, which yields a maximum value for $\theta_i \cdot (fc_i(p_i^*) - (fc_i(p_i^*) - \Delta p_i^*))$. This is repeated until the time constraints on all TAM partitions are satisfied.

The experimental results obtained using the GP-based heuristic procedure are summarized in Table III. The results are represented in a similar fashion as in Table II. The relative defect-screening probability obtained using the GP-based heuristic is greater than that obtained using the enumerative heuristic technique and less than that obtained using the ILP method. The computation time ranges from 6 min for the a586710 SoC to 51 min for the p93791 SoC.

VI. APPROXIMATION ERROR IN \mathcal{P}_S^r

A Taylor's series expansion of $\delta_i(j)\epsilon_i(j)$, without the higher order terms, was used in Section III to obtain a linear objective function for \mathcal{P}_{TLTWS} . If the defect-escape probability for core i is much smaller than unity, this assumption can be justified. To study the effect of this approximation, we evaluated the approximation error for the benchmark circuits. We used a commercial NLP solver [34] to incorporate higher order terms in our objective function. The NLP solver [34] uses the generalized reduced gradient method to solve large-scale nonlinear problems [35].

The nonlinear objective function that we use in our experiments is shown as

Minimize \mathcal{F}

$$= \sum_{i=1}^N \left(\sum_{j=1}^{p_i} \delta_{ij}\epsilon_i(j) + \frac{(\delta_{ij}\epsilon_i(j))^2}{2} + \frac{(\delta_{ij}\epsilon_i(j))^3}{3} \right). \quad (9)$$

The relative magnitudes of the quadratic and cubic terms are negligible compared to the leading order term when the defect-escape probability of the core is negligible. We determine the approximation error as a measure to quantify the effect of these

higher order terms on \mathcal{P}_S^r . The approximation error is given by $((\mathcal{P}_{S-ILP}^r - \mathcal{P}_{S-NLP}^r)/\mathcal{P}_{S-ILP}^r) \times 100\%$.

We present experimental results on the approximation error in \mathcal{P}_S^r when ILP and heuristic methods are used to solve \mathcal{P}_{TLTWS} versus when NLP- and GP-based methods are used. We use a commercial solver [33] for the GP-based heuristic method. The relative defect-screening probability was determined for a nonlinear objective function using a commercial solver [34], where the quadratic and cubic terms are considered in addition to the leading order term; this procedure is described in Section II.

Let \mathcal{P}_{S-ILP}^r , $\mathcal{P}_{S-e-TLTWS}^r$, \mathcal{P}_{S-NLP}^r , and \mathcal{P}_{S-GP}^r denote the relative defect-screening probability of the SoC obtained using a linear objective function, the defect-screening probability using the enumerative heuristic, the relative defect-screening probability of the SoC using a nonlinear objective function, and the relative defect-screening probability using the GP-based heuristic method, respectively. We determine the approximation error as a measure to quantify the effect of these higher order terms on \mathcal{P}_S^r . The approximation errors obtained using the ILP method are determined as $\delta_{ILP} = ((\mathcal{P}_{SILP}^r - \mathcal{P}_{SNLP}^r)/\mathcal{P}_{SNLP}^r) \times 100\%$. The approximation errors obtained using the heuristic and GP are similarly determined as $\delta_{Heur} = ((\mathcal{P}_{SHeur}^r - \mathcal{P}_{SNLP}^r)/\mathcal{P}_{SNLP}^r) \times 100\%$ and $\delta_{GP} = ((\mathcal{P}_{SGP}^r - \mathcal{P}_{SNLP}^r)/\mathcal{P}_{SNLP}^r) \times 100\%$, respectively.

As it is evident from the aforementioned equations, the results obtained using the NLP solver are used as a baseline case. This is because the results obtained using GP-based heuristic are only bounds (upper bounds on the relative defect-screening probability), and the results obtained using ILP and the enumerative heuristic method are not optimal. The "p" benchmarks do not consider solutions obtained using ILP because of the lack of a suitable solver to solve problems of this size. The approximation errors for the benchmark circuits are presented in Tables IV and V. The time needed by the NLP solver [34] to solve \mathcal{P}_{TLTWS} with the nonlinear objective function ranges from 6 min for the d695 SoC to 4 h for the "p" SoCs from Philips. This clearly indicates that the nonlinear version of \mathcal{P}_{TLTWS} is not scalable for large SoCs. The time to solve the

TABLE IV
APPROXIMATION ERROR IN RELATIVE DEFECT-SCREENING PROBABILITY FOR d695 AND a586710

SoC	W^*	$T_{max} = T_{SoC}$			$T_{max} = 0.75T_{SoC}$			$T_{max} = 0.5T_{SoC}$		
		δ_{Heur}	δ_{GP}	δ_{ILP}	δ_{Heur}	δ_{GP}	δ_{ILP}	δ_{Heur}	δ_{GP}	δ_{ILP}
d695	8	4.36	1.59	0.17	7.77	4.82	0.37	15.09	11.81	3.31
	12	3.83	1.58	0.15	7.48	7.48	0.58	12.73	10.45	2.94
	16	3.23	1.74	0.25	7.10	5.72	1.04	11.31	9.61	2.36
a586710	8	2.47	1.52	0.12	3.77	1.46	0.54	5.40	4.76	0.94
	12	2.18	1.46	0.32	3.27	1.22	0.41	4.87	3.79	0.79
	16	2.05	1.53	0.28	3.03	0.78	0.46	4.36	2.93	1.03

TABLE V
APPROXIMATION ERROR IN RELATIVE DEFECT-SCREENING PROBABILITY FOR THE “p” SoCs

SoC	W^*	$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.25T_{SoC}$	
		δ_{Heur}	δ_{GP}	δ_{Heur}	δ_{GP}	δ_{Heur}	δ_{GP}
p22810	8	2.91	4.09	4.36	7.68	5.40	7.18
	12	1.82	2.92	4.29	7.09	5.31	8.19
	16	1.28	2.26	3.06	4.43	3.62	4.54
p34392	8	0.97	2.62	3.48	5.99	6.54	10.33
	12	1.12	2.56	3.25	5.49	7.55	11.07
	16	1.85	2.64	2.84	4.78	5.67	7.62
p93791	8	4.30	6.62	5.75	8.97	6.48	8.92
	12	6.16	8.87	7.10	10.28	8.21	10.72
	16	4.90	7.01	6.53	7.23	8.35	9.15

GP-based heuristic ranges from 2 min for the d695 SoC to 45 min for the “p” SoCs. The GP-based heuristic can therefore be used to quickly determine bounds on \mathcal{P}_S^c .

VII. CONCLUSION

We have formulated a test-length and a TAM width selection problem for wafer-level testing of core-based digital SoCs. To the best of our knowledge, this is the first attempt to incorporate TAM-width selection in an RPCT-based wafer-level SoC test flow. We have also incorporated core defect probabilities into the modeling and optimization framework. The optimization problem has been tackled using ILP, NLP, GP, and heuristic methods. The experimental results for the ITC’02 SoC test benchmarks show that the proposed approach can contribute to effective defect screening at wafer sort.

REFERENCES

- [1] Y. Zorian, E. Marinissen, and S. Dey, “Testing embedded core-based system chips,” *Computer*, vol. 32, no. 6, pp. 52–60, Jun. 1999.
- [2] *International Technology Roadmap for Semiconductors: Assembly and Packaging*, 2005. [Online]. Available: <http://www.itrs.net/Links/2005ITRS/AP2005.pdf>
- [3] C. P. Su and C. W. Wu, “A graph-based approach to power-constrained SoC test scheduling,” *J. Electron. Test.: Theory Appl.*, vol. 20, no. 1, pp. 45–60, Feb. 2004.
- [4] V. Iyengar, K. Chakrabarty, and E. Marinissen, “Test access mechanism optimization, test scheduling, and tester data volume reduction for system-on-chip,” *IEEE Trans. Comput.*, vol. 52, no. 12, pp. 1619–1632, Dec. 2003.
- [5] V. Iyengar, K. Chakrabarty, and E. Marinissen, “Test wrapper and test access mechanism co-optimization for system-on-chip,” *J. Electron. Test.: Theory Appl.*, vol. 18, no. 2, pp. 213–230, Apr. 2002.
- [6] E. Larsson, K. Arvidsson, H. Fujiwara, and Z. Peng, “Efficient test solutions for core-based designs,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 758–775, May 2004.
- [7] Q. Xu and N. Nicolici, “Modular SoC testing with reduced wrapper count,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1894–1908, Dec. 2005.
- [8] G. Chen, S. M. Reddy, and I. Pomeranz, “Procedures for identifying untestable and redundant transition faults in synchronous sequential circuits,” in *Proc. Int. Conf. Comput. Des.*, 2003, pp. 36–41.
- [9] S. K. Goel and E. J. Marinissen, “Effective and efficient test architecture design for SOCs,” in *Proc. Int. Test Conf.*, 2002, pp. 529–538.
- [10] C. Liu, K. Chakrabarty, and W.-B. Jone, “System/network-on-chip test architectures,” in *System-on-Chip Test Architectures: Nanometer Design for Testability*, L.-T. Wang, C. Stround, and N. A. Toubia, Eds. Amsterdam, The Netherlands: Elsevier, 2007.
- [11] A. B. Kahng, “The road ahead: The significance of packaging,” *IEEE Des. Test. Comput.*, vol. 19, no. 6, pp. 104–105, Nov. 2002.
- [12] W. R. Mann, F. L. Taber, P. W. Seitzer, and J. J. Broz, “The leading edge of production wafer probe test technology,” in *Proc. Int. Test Conf.*, 2004, pp. 1168–1195.
- [13] G. Breed, “On-wafer testing verifies IC performance and process yield,” *High Freq. Electron.*, vol. 5, no. 5, pp. 66–70, May 2006.
- [14] S. K. Goel and E. J. Marinissen, “On-chip test infrastructure design for optimal multi-site testing of system chips,” in *Proc. Des. Autom. Test Eur. Conf.*, 2005, pp. 44–49.
- [15] J. Jahangiri, N. Mukherjee, W. T. Cheng, S. Mahadevan, and R. Press, “Achieving high test quality with reduced pin count testing,” in *Proc. Asian Test Symp.*, 2005, pp. 312–317.
- [16] R. W. Bassett, B. J. Butkus, S. L. Dingle, M. R. Faucher, P. S. Gillis, J. H. Panner, J. G. Petrovick, and D. L. Wheeler, “Low-cost testing of high-density logic components,” in *Proc. Int. Test Conf.*, 1989, pp. 550–557.
- [17] J. Darringer, E. Davidson, D. J. Hathaway, B. Koenemann, M. Lavin, J. K. Morrell, K. Rahmat, W. Roesner, E. Schanzenbach, G. Tellez, and L. Trevillyan, “EDA in IBM: Past, present, and future,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 12, pp. 1476–1497, Dec. 2000.
- [18] H. Vranken, T. Waayers, H. Fleury, and D. Lelouvier, “Enhanced reduced pin-count test for full-scan design,” in *Proc. Int. Test Conf.*, 2001, pp. 738–747.
- [19] T. G. Foote, D. E. Hoffman, W. V. Huott, T. J. Koprowski, M. P. Kusko, and B. J. Robbins, “Testing the 500-MHz IBM S/390 microprocessor,” *IEEE Des. Test Comput.*, vol. 15, no. 3, pp. 83–89, Jul.–Sep. 1998.
- [20] E. Larsson, J. Pouget, and Z. Peng, “Defect-aware SoC test scheduling,” in *Proc. VLSI Test Symp.*, 2004, pp. 228–233.
- [21] U. Ingelsson, S. K. Goel, E. Larsson, and E. J. Marinissen, “Test scheduling for modular SOCs in an abort-on-fail environment,” in *Proc. Eur. Test Symp.*, 2005, pp. 8–13.
- [22] S. Bahukudumbi and K. Chakrabarty, “Defect-oriented and time-constrained wafer-level test-length selection for core-based digital SOCs,” in *Proc. Int. Test Conf.*, 2006, pp. 1–10.
- [23] S. Bahukudumbi and K. Chakrabarty, “Wafer-level modular testing of core-based SOCs,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1144–1154, Oct. 2007.
- [24] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, “A set of benchmarks for modular testing of SOCs,” in *Proc. Int. Test Conf.*, 2002, pp. 519–528.

- [25] S. Koranne, "A novel reconfigurable wrapper for testing of embedded core-based SOCs and its associated scheduling algorithm," *J. Electron. Test.: Theory Appl.*, vol. 18, no. 4/5, pp. 415–434, Aug. 2002.
- [26] E. Larsson and H. Fujiwara, "System-on-chip test scheduling with reconfigurable core wrappers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 3, pp. 305–309, Mar. 2006.
- [27] *IEEE Standard Testability Method for Embedded Core-Based Integrated Circuits*, IEEE Std. 1500, 2005.
- [28] K. Chakrabarty, V. Iyengar, and M. D. Krasniewski, "Test planning for modular testing of hierarchical SOCs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 3, pp. 435–448, Mar. 2005.
- [29] M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Norwell, MA: Kluwer, 2000.
- [30] E. Larsson and H. Fujiwara, "Optimal system-on-chip test scheduling," in *Proc. Asian Test Symp.*, 2004, pp. 306–311.
- [31] E. Kreyszig, *Advanced Engineering Mathematics*, 8th ed. Hoboken, NJ: Wiley, 1998.
- [32] S. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optim. Eng.*, vol. 8, no. 1, pp. 67–127, Apr. 2007.
- [33] *TOMLAB Optimization: TOMLAB/GP*. [Online]. Available: <http://www.tomopt.com>
- [34] *Premium Solver Platform*, Frontline Syst. Inc., Incline Village, NV, 2007. [Online]. Available: <http://www.solver.com/xlspplatform.html>
- [35] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. Hoboken, NJ: Wiley, 2006.



Krishnendu Chakrabarty (S'92–M'96–SM'00–F'08) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Kharagpur, India, in 1990, and the M.S.E. and Ph.D. degrees in computer science and engineering from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively.

He is currently a Professor of electrical and computer engineering with Duke University, Durham, NC. He has contributed 15 invited chapters to book volumes, published 290 papers in archival journals and refereed conference proceedings, and delivered over 110 keynote, plenary, and invited talks. He is the holder of a U.S. patent in built-in self-test and is a coinventor of a pending U.S. patent on sensor networks. He has authored five books—*Microelectrofluidic Systems: Modeling and Simulation* (CRC Press, 2002), *Test Resource Partitioning for System-on-a-Chip* (Kluwer, 2002), *Scalable Infrastructure for Distributed Sensor Networks* (Springer, 2005), *Digital Microfluidics Biochips: Synthesis, Testing, and Reconfiguration Techniques* (CRC Press, 2006), and *Adaptive Cooling of Integrated Circuits using Digital Microfluidics* (Artech House, April 2007)—and edited the book volumes *SOC (System-on-a-Chip) Testing for Plug and Play Test Automation* (Kluwer, 2002) and *Design Automation Methods and Tools for Microfluidics-Based Biochips* (Springer, 2006). His current research projects include testing and design for testability of system-on-chip integrated circuits, digital microfluidic biochips, nanotechnology circuits and systems based on DNA self-assembly, and delay-tolerant wireless networks.

Prof. Chakrabarty is a Distinguished Engineer of ACM and a member of Sigma Xi. He served as a Distinguished Visitor of the IEEE Computer Society for 2005–2007 and as a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2006–2007. Since 2008, he has been serving as an ACM Distinguished Speaker. He is an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON VLSI SYSTEMS, IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, and *ACM Journal on Emerging Technologies in Computing Systems*. He is also an Editor of the *IEEE Design & Test of Computers* and *Journal of Electronic Testing: Theory and Applications*. He recently completed his term as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM I (2006–2007). He received a Meritorious Service Award from the IEEE Computer Society in 2008. He is a recipient of Duke University's 2008 Dean's Award for Excellence in Mentoring. He is also a recipient of the National Science Foundation Early Faculty (CAREER) award (1999) and the Office of Naval Research Young Investigator award (2001) and a recipient of best paper awards at the 2007 IEEE International Conference on VLSI Design, the 2005 *IEEE International Conference on Computer Design*, and the 2001 *IEEE Design, Automation and Test in Europe (DATE) Conference*. He is also a recipient of the Humboldt Research Fellowship, awarded by the Alexander von Humboldt Foundation, Germany, in 2004, and the Mercator Visiting Professorship for 2000–2002, awarded by the Deutsche Forschungsgemeinschaft, Germany.



Sudarshan Bahukudumbi (S'06) received the B.E. degree in electrical and electronics engineering from the Sri Venkateswara College of Engineering, University of Madras, Chennai, India, in 2003, the M.S. degree in electrical and computer engineering from New Mexico State University, Las Cruces, in 2005, and the Ph.D. degree in electrical and computer engineering from Duke University, Durham, NC, in 2008.

He is currently with Intel Corporation, Hillsboro, OR, as a Quality and Reliability Test Engineer. His research interests include test planning and test resource optimization for wafer-level testing of integrated circuits.