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

Sudarshan Bahukudumbi and Krishnendu Chakrabarty

Department of Electrical and Computer Engineering

Duke University, Durham, NC 27708, USA

Email: {spb, krish}@ee.duke.edu

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 leads 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 that addresses test access mechanism (TAM) optimization and test-length selection for wafer-level testing of core-based digital 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.

I. INTRODUCTION

The consumer electronics market is characterized by low product cost and decreasing profit margins. Rapid advances in process technology and design tools have led to system-on-chip (SoC) integrated circuits that reduce the design cycle time. Short design cycle times are achieved by integrating a number of pre-designed and pre-verified 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]–[9]. 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 [10]. 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 integrated circuits (ICs) [11]–[13]. However, a large number of wafer probe contacts leads 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 [14]. Since test time and the number of contacted chip pins are major practical constraints for wafer sort, all scan-based digital tests cannot always be applied to the die-under-test.

Reduced-pin count testing (RPCT) has been advocated as a design-for test technique, especially for use at wafer sort, to reduce the number of IC pins that needs to be contacted by the tester [14]–[17]. 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.

Test cost for SoCs can also be reduced by using estimated defect probabilities for the embedded cores to guide test scheduling [18], [19]. These defect probabilities 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. Short product cycles in practice make the defect estimation process extremely difficult. The defect probabilities for the cores were assumed in [18] to be either known *a priori* or obtained by binning the failure information for each individual core over the product cycle [19]. 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 SOC's was presented in [20]. 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 for optimization.

In this paper, we present an optimization framework that addresses TAM optimization and test-length selection for wafer-level testing of core-based digital SoCs. The objective here is to design a TAM architecture for wafer sort that utilizes a pre-designed underlying TAM architecture for package test, 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. 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 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 two techniques for test-length selection and TAM optimization. The first technique is based on the formulation of a non-linear integer programming model, which can be subsequently linearized and solved using standard integer linear programming (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 [20] to derive test-lengths to maximum defect screening at wafer sort. This enumerative procedure allows an efficient search of a large solution space it results in significantly lower computation time than that needed for the first method. Simulation results on TAM optimization and test-length selection are presented for five of the ITC'02 SoC Test benchmarks [21].

The remainder of the paper is organized as follows. In Section 2, we briefly describe how the defect screening probabilities of the cores and the SoC are determined using the approach presented in [20].

*This research was supported in part by the Semiconductor Research Corporation under contract no. 2004-TJ1174.

Section 3 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 3. Section 4 presents the second optimization method based on the enumeration of TAM partitions. Simulation results for four of the ITC'02 SoC Test benchmarks are presented. Finally, Section 5 concludes the paper and outlines directions for future work.

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 [20]. The defect probabilities for the embedded cores are obtained using the yield modeling technique presented in [20]. Let us now define the following statistical events for Core i :

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 [20].

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 above 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. 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 . Using these variables, we can rewrite Equation (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 [11]–[13]. RPCT methods are therefore desirable for wafer sort. In [20], 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, [20] makes 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 problem for RPCT.

A. Test data serialization

Suppose 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 pre-designed TAM architecture for package test needs to be appropriately modified.

Fig. 1(a) shows a wrapped core that is connected to a 4-bit wide TAM width ($w_i = 4$). For the same wrapped core, Fig. 1(b) outlines a modified test access design that allows 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 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 in to the boundary input cells. Note the above design is fully compliant with the IEEE 1500 standard [22] 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 pre-designed TAM architecture. At the wafer level, if only w_i^* bits are available for TAM partition j , we assume, as in [23] 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 $\lceil \frac{w_i(j)}{w_i^*} \rceil \cdot T_i(j)$ clock cycles. In the example of Fig. 1(b), the test time for core i due to serialization for 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 [24], [25]. In our work, without loss of generality, we use the normalized function $f_{c_i}(p_i^*) = \frac{\log_{10}(p_i^* + 1)}{\log_{10} p_i}$ to represent this relationship. A similar relationship was used in [25].

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

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 Equation (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 [20]. Let us now consider an SoC with a top-level TAM width of W bits and suppose it has $\mathbf{B}$ TAM partitions of widths $[w_1, w_2, \dots, w_B]$, respectively. For a given value of wafer level (maximum) TAM width W^* , we need to determine appropriate TAM sub-partitions of widths $w_1^*, w_2^*, \dots, w_B^*$ such that $w_i^* \leq w_i$, $1 \leq i \leq \mathbf{B}$, and $w_1^* + w_2^* + \dots + w_B^* = W^*$. The optimization problem $\mathcal{P}_{TLTWS}$ can now be formally stated as follows:

Problem $\mathcal{P}_{TLTWS}$: Given a pre-designed 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 (i) the total number of test patterns to be applied to each core and (ii) reduction in TAM width for each partition such that: (a) the overall testing time on each TAM partition does not exceed the upper bound T_{max} and (b) 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 ensure 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 $\epsilon_i^* = \sum_{j=1}^{q_i} \delta_{ij} \epsilon_i(j)$. 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- and higher-order terms [26]. 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 (T_{SoC}). Due to serialization, the testing time for core i on TAM partition j , is given by $\lceil (w_i(j)/w_i^*(j)) T_i(j) \rceil$ [23]. Therefore the test time of core i when it is tested with a reduced bitwidth of w_i^* is given by Equation (5).

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

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 above definition that $\sum_{k=1}^{w_i} \lambda_{ik} = 1$ and Equation (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 shown in Equation (8). A_j denotes the set of cores that are assigned to TAM partition j . The complete ILP model is shown in Fig. 5.

$$\sum_{k=1}^{w_i} k \cdot \lambda_{ik} = W_x^*; \forall \text{Core } i \in A_j \quad (8)$$

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

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

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

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

Fig. 2. Integer linear programming model for $\mathcal{P}_{TLTWS}$.

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 $2 \cdot N + 2 \sum_{i=1}^N (p_i \cdot w_i) + B + 1$.

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

We now present the experimental results for two SoCs from the ITC'02 SoC test benchmark suite [21]. We use the public domain ILP solver *lpsolve* for our experiments [27]. Since the objectives of our experiment are to select the number of test patterns in a time- and bitwidth-constrained wafer sort environment, and at the same time maximize the defect screening probability, we present the following results:

- 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.
- The values of TAM partition widths $w_1^*, w_1^*, \dots, w_B^*$ such that $w_1^* + w_1^* + \dots + w_B^* = W^*$.
- 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.
- The relative defect-screening probability for each SoC obtained using the ILP model.

We first present results on the number of patterns determined for the cores. The results for the d695 benchmark SoC is presented 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 illustrates the defect-screening probabilities for the cores in the d695 benchmark for the above-mentioned 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 hours 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.

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

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

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

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

- (i) 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

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

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
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

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 above condition are stored as a valid partition. We illustrate the above enumeration procedure with a small example. Let us consider an SoC whose TAM architecture is fixed and designed for 5 bits, and partitioned into three TAM partitions of widths 2, 3, and 1 respectively. The possible TAM enumerations for the above partitions are {111, 121, 131, 211, 221, 231}. If we consider W^* to be 4, then the valid TAM partitions are {121, 211}.

(ii) For each valid TAM partition calculated in Step (i), we apply the test-length selection procedure $\mathcal{P}_{TLS}$ from [20]. We calculate the defect screening probability for the SoC from the results obtained using $\mathcal{P}_{TLS}$.

The objective function for $\mathcal{P}_{TLS}$ is the same as Equation (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 $\sum_{Core_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$. The complete ILP model is shown as Fig. 5

(iii) 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.

(iv) We repeat this procedure until all possible TAM partitions are enumerated.

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

$$1. \sum_{i=1}^{p_i} \delta_{ij} = 1, 1 \leq i \leq N, 0 \leq j \leq p_i$$

$$2. \sum_{Core_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$$

$$3. \delta_{ij} = 0 \text{ or } 1, 1 \leq i \leq N, 0 \leq j \leq p_i$$

/* 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. Integer linear programming model for $\mathcal{P}_{TLS}$ [20].

Experimental results obtained using the $\mathcal{P}_{e-TLTWS}$ 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 four benchmark circuits [21], as well as the recommended TAM partition widths for wafer-sort are shown in the table. The number of patterns determined using $\mathcal{P}_{e-TLTWS}$ for the p34392 SoC is illustrated in Fig. 6. The results are shown for three values of T_{max} : T_{SoC} , $0.75T_{SoC}$ and $0.5T_{SoC}$. Results are reported only

for $W^* = 16$ and $W = 32$; similar plots are obtained for different values of W^* and W . Fig. 7 illustrates the relative defect-screening probabilities for the cores in the p34392 benchmark for the above-mentioned test case. The computation time for the largest benchmark SoC p93791 was only 4 minutes, hence this approach is suitable for large designs.

V. CONCLUSIONS

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 the wafer-level SoC test flow. We have also incorporated core defect probabilities into the modeling and optimization framework. 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," *IEEE Computer*, vol. 32, pp. 52–60, Jun. 1999.
- [2] International Technology Roadmap for Semiconductors: Assembly and Packaging, <http://www.itrs.net/Links/2005ITRS/AP2005.pdf> 2005.
- [3] C. P. Su and C. W. Wu, "A graph-based approach to power-constrained SoC test scheduling," *JETTA*, vol. 19, 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. Computers*, vol. 52, pp. 1619–1632, Dec. 2003.
- [5] —, "Test wrapper and test access mechanism co-optimization for system-on-chip," *JETTA*, vol. 18, pp. 213–230, Apr. 2002.
- [6] E. Larsson et al., "Efficient test solutions for core-based designs," *IEEE Trans. CAD*, vol. 23, pp. 758–775, May 2004.
- [7] Q. Xu and N. Nicolici, "Modular SoC testing with reduced wrapper count," *IEEE Trans. CAD*, vol. 24, pp. 1894–1908, Dec. 2005.
- [8] W. Zou, S. M. Reddy, and I. Pomeranz, "SoC test scheduling using simulated annealing," in *Proc. VLSI Test Symp.*, 2003, pp. 325–330.
- [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] A. B. Kahng, "The road ahead: The significance of packaging," *IEEE Design & Test*, vol. 52, pp. 1619–1632, Dec. 2003.
- [11] W. R. Mann et al., "The leading edge of production wafer probe test technology," in *High Frequency Electronics*, 2006, pp. 66–70.
- [12] G. Breed, "On-wafer testing verifies IC performance and process yield," in *Proc. Int. Test Conf.*, 2004, pp. 1168–1195.
- [13] S. K. Goel and E. J. Marinissen, "Test infrastructure design for optimal multi-site testing of system chips," in *Proc. DATE Conf.*, 2005, pp. 44–49.
- [14] J. Jahangiri et al., "Achieving high test quality with reduced pin count testing," in *Proc. Asian Test Symp.*, 2005, pp. 312–317.
- [15] R. W. Basset et al., "Low-cost testing of high-density logic components," in *Proc. Int. Test Conf.*, 1989, pp. 550–758.
- [16] J. Darringer et al., "EDA in IBM: Past, present, and future," *IEEE Trans. CAD*, vol. 19, pp. 1476–1497, Dec. 2000.
- [17] H. Vranken et al., "Enhanced reduced pin-count test for full scan design," in *Proc. Int. Test Conf.*, 2001, pp. 738–747.
- [18] E. Larsson et al., "Defect-aware SoC test scheduling," in *Proc. VLSI Test Symp.*, 2004, pp. 228–233.
- [19] U. Ingelsson et al., "Test scheduling for modular SOCs in an abort-on-fail environment," in *Proc. European Test Symp.*, 2005, pp. 8–13.

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 $W = 32$.

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

SoC	W^*	$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.50T_{SoC}$	
		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
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
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

- [20] 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* (to appear).
- [21] E. J. Marinissen et al., "A set of benchmarks for modular testing of SoCs," in *Proc. Int. Test Conf.*, 2002, pp. 519–528.
- [22] IEEE Std. 1500. IEEE Standard for Embedded Core Test - IEEE Std. 1500-2004. IEEE, New York, 2004.
- [23] K. Chakrabarty, V. Iyengar, and M. D. Krasniewski, "Test planning for modular testing of hierarchical SoCs," *IEEE Trans. CAD*, vol. 24, pp. 435–448, Mar. 2005.
- [24] M. Bushnell and V. Agarwal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Kluwer, 2000.
- [25] E. Larsson and H. Fujiwara, "Optimal system-on-chip test scheduling," in *Proc. Asian Test Symp.*, 2004, pp. 306–311.
- [26] E. Kreyszig, *Advanced Engineering Mathematics*, 8th ed. John Wiley & Sons Inc., 1998.
- [27] M. Berkelaar et al., "Ipsolve: Open source (mixed-integer) linear programming system". Version 5.5 dated May 16, 2005 URL: <http://www.geocities.com/lpsolve>.