Defect-Oriented and Time-Constrained Wafer-Level Test-Length Selection for 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

Product cost is a major driver in the consumer electronics market, which is characterized by low profit margins and the use of core-based system-on-chip (SoC) designs. Packaging has been recognized as a significant contributor to the product cost for such SoCs. To reduce packaging cost and the test cost for packaged chips, wafer-level testing (wafer sort) is used in the semiconductor industry to screen defective dies. However, since test time is a major practical constraint for wafer sort, even more so than for package test, not all the scan-based digital tests can be applied to the die under test. We present an optimal test-length selection technique for wafer-level testing of core-based SoCs. This technique, which is based on a combination of statistical yield modeling and integer linear programming, allows us to determine the number of patterns to use for each embedded core during wafer sort such that the probability of screening defective dies is maximized for a given upper limit on the SoC test time. Simulation results are presented for five of the ITC'02 SoC Test benchmarks.

1. Introduction

Rapid advances in process technology and design tools have led to the widespread use of system-on-chip (SoC) integrated circuits (ICs). In order to reduce the design cycle time, a large number of pre-designed embedded cores are often integrated in todays SoCs. While the testing of such core-based SoCs continues to be a major concern in the semiconductor industry [1], [2], the recent IEEE 1500 Standard addresses some aspects of the testing of core-based SoCs [3]. A standardized 1500 wrapper can either be provided by the core vendor or it can be implemented during system integration. This wrapper can be used, among other things, to provide isolation to an embedded core during test application. The design of test access mechanisms (TAMs) and SoC test scheduling are additional problems that must be tackled during system integration. A number of efficient solutions have recently been proposed for TAM optimization and test scheduling [4]–[10]; these methods are mostly aimed at reducing the test time for package test, where all the scan-based patterns for the digital embedded cores are applied to the SoC.

A recent SoC test scheduling method attempted to minimize the average test time for a packaged SoC, assuming an abort-on-first fail strategy [11], [12]. The key idea in this work is to use defect probabilities for the embedded cores to guide the test scheduling procedure. These defect probabilities are 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 [11] to be either known *a priori* or obtained by binning the failure information for each individual core over the product cycle [12]. 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. Therefore, a yield modeling technique is needed to accurately estimate these defect probabilities.

Many SoC designs today target the consumer electronics market, which is characterized by low product cost and narrow profit margins. Test and packaging costs are significant contributors to the product cost for such SoCs [13]. To reduce packaging cost and the test cost for packaged chips, wafer-level testing (wafer sort) is used in the semiconductor industry to screen defective dies [14]. However, since test time is a major practical constraint for wafer sort, even more so than for package test, not all the scan-based digital tests can be applied to the die under test. It is therefore important to determine the number of test patterns for each core that must be used for the given upper limit on SoC test time for wafer sort, such that the probability of successfully screening a defective die is maximized. The number of patterns need to be determined on the basis of a yield model that can estimate the defect probabilities of the embedded cores, as well as a "test escape model" that provides information on how the fault coverage for a core depends on its test length.

In this paper, we present an optimal test-length selection technique for wafer-level testing of core-based SoCs. This technique, which is based on a combination of statistical yield modeling and integer linear programming (ILP), allows us to determine the number of patterns to use for each embedded core during wafer sort such that the probability of screening defective dies is maximized for a given upper limit on the SoC test time. Therefore, this work complements prior work on SoC test scheduling that lead to efficient test schedules that reduce the testing time during package test. For a given test access architecture, designed to minimize test time for all the

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

1-4244-0292-1/06/$20.00 © 2006 IEEE

scan patterns during package test, the proposed method can be used at wafer sort to screen defective dies, thereby reducing package cost and the subsequent test time for the IC lot. 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.

The key contributions of this paper are as follows:

- We show how statistical yield modeling for defect-tolerant circuits can be used to estimate defect probabilities for embedded cores in an SoC.
- We formulate the test-length selection problem for wafer-level testing of core-based SoCs. To the best of our knowledge, this is the first attempt to define a core-based test selection problem for SoC wafer sort.
- We develop an ILP model to obtain optimal solutions for the test-length selection problem. The optimal approach is applied to five ITC'02 SoC test benchmarks, including three from industry.

The remainder of the paper is organized as follows. Section 2 describes prior work on a statistical yield modeling approach for defect-tolerant circuits and shows how it is adapted for estimating the defect probabilities for embedded cores. Section 3 formulates the test-length selection problem and describes an ILP model to solve this problem optimally. Section 4 presents simulation results for five of the ITC'02 SoC Test benchmarks. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Defect Probability Estimation for Embedded Cores

In this section, we show how defect probabilities for embedded cores in an SoC can be estimated using statistical yield modeling techniques.

2.1 Unified Negative-Binomial Model for Yield Estimation

We adapt the yield model presented in [15]–[17] to model the yield of the individual cores in a generic core-based SoC. The model presented in [15] unifies the "small-area clustering" and "large-area clustering" models presented in [16] and [17], respectively. It is assumed in [15], [16] that the number of defects in a given area A is a random variable that follows a negative-binomial distribution with parameters λ_A and α_A. The probability $\mathcal{P}(x, A)$ that x faults occur in area A is given by:

$$\mathcal{P}(x, A) = \frac{\Gamma(\alpha_A + x)}{x!\Gamma(\alpha_A)} \cdot \frac{(\lambda_A/\alpha_A)^x}{(1 + (\lambda_A/\alpha_A)^{\alpha_A + x}} \qquad (1)$$

The above yield model has been validated using industrial data [18]. An additional parameter incorporated in [15] is the *block size*, defined as the smallest value B such that the wafer can be divided into disjoint regions, each of size B, and these regions are statistically independent with respect to manufacturing defects. As in [15], we assume that the blocks are rectangular and can be represented by a tuple (B_1, B_2), corresponding to the dimensions of the rectangle. The goal of the yield model in [15] was to determine the effect of redundancy on yield in a fault-tolerant VLSI system. The basic redundant block is called a module, and the block is considered to be made up of an integer number of modules. Since our objective here is to model the yield of embedded (non-overlapping) cores in an SoC, we redefine the module to be an imaginary chip area denoted by (a_1, a_2). The size of the imaginary chip area, i.e., the values of a_1 and a_2 can be fixed depending on the resolution of the measurement system, e.g., an optical defect inspection setup.

2.2 Procedure to Determine Core Defect Probabilities

We use the following steps to estimate the defect probabilities for the embedded cores:

(1) Determine the block size: The block size needs to be determined first. Efficient techniques for determining the block size have been presented in [15], and these techniques have been validated using empirical data. The block size can be determined using a simple iterative procedure, in which the wafer is divided into rectangular sub-areas (blocks), whose sizes are increased at every step. Starting with blocks of size $I = 1$, $J = 1$, we alternately increase I and J. For each fixed value of block size $I \cdot J$, we then calculate the corresponding parameter $\alpha_B(I, J)$ and arrange these values in a matrix form. The value of (I, J), for which the difference between $\alpha_B(I, J)$ and $\alpha_B(1, 1)$ is minimum, is chosen as the block size. In our work, we make the following assumptions: (a) as in [15], we assume that the area of the block consists of an integer number of imaginary chip areas; (b) the block size and its negative binomial parameters are pre-determined using rigorous statistical information processing on wafer defect maps.

(2) We consider each core in the SoC to be an "independent chip". Let us consider a core represented by (C_1, C_2), block size (B_1, B_2), and imaginary chip (a_1, a_2). This representation is illustrated in Figure 1. We then determine the probability that the core is defective using the following steps:

(a) The number of possible orientations of the core with respect to the block in the wafer is given by $\min\{B_1, C_1\} \cdot \min\{B_2, C_2\}$.

(b) For each orientation, determine the distance from the left-top corner of the core to the closest block boundaries. This is represented as (R_1, R_2), the two values denoting distances in the X and Y directions, respectively; the placement of the core with respect to the block determines the way the core is divided into complete and partial blocks.

(c) The dimensions of the core can now be represented as $C_1 = R_1 + n_1 \cdot B_1 + m_1$, and $C_2 = R_2 + n_2 \cdot B_2 + m_2$, where n_1 and m_1 are defined as:

$$n_1 = \lfloor (\tfrac{C_1 - R_1}{B_1}) \rfloor$$
$$m_1 = (C_1 - B_1) \mod B_1$$

The parameters n_2 and m_2 are defined in a similar fashion.

(d) The core can be divided into a maximum of nine disjoint sub-areas as shown in Figure 1, each in a different block.

Dividing the core into independent sub-areas allows for the convolution of the probability of failure of each individual sub-area. Let us assume that there are a total of D sub-areas; the probability that the core is defect-free is given by $\mathcal{P}^{(R_1,R_2)} = \prod_{i=1}^{D} a(N_i)$. The superscript (R_1, R_2) indicates the dependence of this probability on the placement. Here $a(N_i)$ denotes the probability that all the N_i imaginary chip areas in the sub-area i are defect-free. This probability can be obtained from Equation (2) shown below, where $a(k, N)$ denotes the probability of k defect-free modules in a sub-area with N modules. By substituting N instead of k in Equation (2), we obtain Equation (3). This is done in order to estimate the probability that a block is fault-free.

$$a(k,N) = \binom{N}{k} \sum_{i=0}^{N-k} (-1)^i \binom{N-k}{i} \left(1 + \frac{(i+k)\lambda_m}{\alpha_m}\right)^{-\alpha_m} \tag{2}$$

$$a(N,N) = a(N) = \left(1 + \frac{N\lambda_m}{\alpha_m}\right)^{-\alpha_m} \tag{3}$$

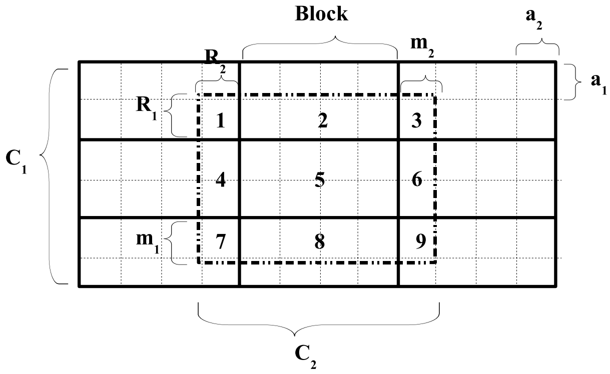

Fig. 1. Defect estimation: Placement of a core with respect to blocks.

The process of dividing the area of a core into multiple sub-areas facilitates the application of large-area clustering conditions on the individual sub-areas. It is important to distinguish between sub-areas $i = 1, 3, 7, 9$ and $i = 2, 4, 5, 6, 8$ in Figure 1. In the latter case, the sub-area i is divided into several parts, each contained in a different block. The derivation of the probability density function for these sub-areas is now a trivial extension of the base case represented by Equation (3).

(e) The final step is the estimation of the defect probability for the core. We first estimate the probability that the core is defect-free for all possible values of R_1 and R_2. The overall defect-free probability P is obtained by averaging the defect-free probability over all orientations, and it is given by:

$$\mathcal{P} = \frac{1}{\min(B_1, C_1) \cdot \min(B_2, C_2)} \sum_{R_1=1}^{\min(B_1,C_1)} \sum_{R_2=1}^{\min(B_2,C_2)} \mathcal{P}^{(R_1,R_2)} \tag{4}$$

We use Figure 1 to illustrate the calculation of the defect probability for an embedded core. The figure represents the relative placement of a core with respect to the blocks. We have a block size of $(4, 2)$, a core size of $(6, 4)$ and imaginary chip area of size $(1, 1)$. For the illustrated orientation, we have $R_1 = R_2 = 1$. The values of n_1, m_1, n_2, m_2 for the illustrated orientation are all 1. The core is divided into nine distinct sub-areas numbered $1-9$. For values of $\alpha_B = 0.25$ and $\lambda_B = 0.1$, we now determine the probability that the core is defect-free using Equation (3):

$$\begin{aligned}
\mathcal{P}^{(1,1)} &= a(R_1 \cdot R_2) \cdot a(R_1 \cdot B_2)^{n_1} \cdot a(R_1 \cdot m_2) \cdot a(B_1 \\
&\quad \cdot R_2)^{n_2} \cdot a(B_1 \cdot B_2)^{n_1} \cdot a(R_1 \cdot m_2) \cdot a(m_1 \cdot R_2) \cdot \\
&\quad a(m_1 \cdot B_2)^{n_1} \cdot a(m_1 \cdot m_2) \\
&= a(1) \cdot a(4)^{n_1} \cdot a(1) \cdot a(2)^{n_2} \cdot a(8)^{n_1} \cdot a(2) \cdot a(1) \cdot \\
&\quad a(4)^{n_1} \cdot a(1) \\
&= (0.9879) \cdot (0.9554) \cdot (0.9879) \cdot (0.9765) \cdot (0.9193) \\
&\quad \cdot (0.9765) \cdot (0.9879) \cdot (0.9879) \cdot (0.9554) \\
&= 0.76206
\end{aligned} \tag{5}$$

The above procedure is repeated until the defect-free probability for all $\min(B_1, C_1) \cdot \min(B_2, C_2)$ combinations of R_1 and R_2 are determined. The final core defect-free probability is then calculated using Equation (4). The probability that the core has a defect is simply $\bar{\mathcal{P}} = 1 - \mathcal{P}$. For a given SoC, this procedure can be repeated for every embedded core until all defect probabilities are obtained.

The knowledge of the dimensions of each individual core is necessary to determine the corresponding defect probabilities. In this paper, we use the overall SoC dimensions as given in [19] to derive information pertaining to the size of the individual modules in the ITC'02 SOC Test benchmarks. Since these benchmarks do not provide information about the sizes of the embedded cores, we use the total number of patterns for each core as an indicator of size. This assumption helps us extract the relative size of a core by normalizing it with respect to the overall SoC dimensions. Table I shows the defect probabilities for each core in the d695 benchmark circuit [20], estimated using the parameters $\alpha_B = 0.25$ and $\lambda_B = 0.035$. We use layout information in the form of X-Y coordinates for the cores as described in [19] to determine the defect probabilities.

3. Test-Length Selection for Wafer-Level Test

In this section, we formulate the problem of determining the test-length for each embedded core, such that for a given upper limit on the SoC test time (expressed as a percentage of the total SoC test time), the defect screening probability is maximized. We present a framework that incorporates the defect probabilities of the embedded cores in the SoC, the upper bound on SoC test time at wafer sort, the test lengths for the cores, and the probability that a defective SoC is screened. The defect probabilities for the cores are obtained using the yield model presented in Section 2. Let us now define the

TABLE I

CORE DEFECT PROBABILITIES FOR THE D695 BENCHMARK SOC.

$\alpha_b = 0.25$ and $\lambda_b = 0.035$	
SoC: d695	
Core Number	Defect Probability
1	0.0038
2	0.1104
3	0.116
4	0.2162
5	0.2339
6	0.6946
7	0.1766
8	0.1882
9	0.0038
10	0.096

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.

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 $\mathcal{P}(\bar{B}_i \mid \bar{A}_i)$ and $\mathcal{P}(B_i \mid A_i)$, respectively. Using a basic identity of probability theory, we can derive the probability that the test applied to Core i detects a defect:

$$\mathcal{P}(\bar{B}_i) = \mathcal{P}(\bar{B}_i \mid A_i) \cdot \mathcal{P}(A_i) + \mathcal{P}(\bar{B}_i \mid \bar{A}_i) \cdot \mathcal{P}(\bar{A}_i) \qquad (6)$$

Due to SoC test time 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 us suppose the difference between the SoC package test time and the upper limit on wafer sort test time is ΔT clock cycles. The test time for each TAM partition therefore needs to be reduced by ΔT clock cycles, if we assume that the package test times on the TAM partitions are equal. The value of p_i^* adopted for Core i depends on its wrapper design. The larger the difference between the external TAM width and internal test bitwidth (number of scan chains plus the number of I/Os), the greater the impact of $(p_i - p_i^*)$ on ΔT. In fact, given two cores (Core i and Core j) with different wrapper designs, the reduction in the number of patterns by the same amount, i.e., $p_i - p_i^* = p_j - p_j^*$, can lead to different amount of reductions in core test time (measured in clock cycles). Let $fc_i(p_i^*)$ be the fault coverage for Core i with p_i^* test patterns.

We next develop the objective function for the test-length selection problem. The goal of this objective function is to satisfy two objectives: (1) Maximize the probability that Core i fails the test; 2) Minimize the overall test escape probability. The ideal problem formulation is one that leads to an objective function satisfying both the above objectives.

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 (6) as:

$$\mathcal{P}(\bar{B}_i) = fc_i(p_i^*) \cdot \theta_i + \gamma_i \cdot (1 - \theta_i) \qquad (7)$$

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

$$\mathcal{P}(B_i) = 1 - \mathcal{P}(\bar{B}_i) = \theta_i \cdot \beta_i + (1 - \gamma_i) \cdot (1 - \theta_i) \qquad (8)$$

We therefore conclude that for a given value of α_i and γ_i, the objective function that maximizes the probability $\mathcal{P}(\bar{B}_i)$ that Core i fails the test, also minimizes the test escape β_i. Therefore, it is sufficient to maximize $\mathcal{P}(B_i)$ to ensure that the test escape rate is minimized. Assuming that the cores fail independently with the probabilities derived in Section 2, the defect screening probability $\mathcal{P}_S$ for an SoC with N embedded cores is given by $\mathcal{P}_S = 1 - \prod_{i=1}^{N} \mathcal{P}(\bar{B}_i)$.

3.1 Test-Length Selection Problem: $\mathscr{P}_{TLS}$

We next present the test-length selection problem $\mathscr{P}_{TLS}$, wherein we determine an optimal number of test patterns for each core in the SoC, such that we maximize the probability of screening defective dies at wafer sort for a given upper limit on the SoC test time. We assume a fixed-width TAM architecture as in [6], where the division of W wires into **B** TAM partitions, and the assignment of cores to the **B** TAM partitions have been determined *a priori* using methods described in [6], [21]–[23].

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. The fixed-width TAM architecture requires 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 $fc_i(p_i^*) = \frac{\log_{10}(p_i^*+1)}{\log_{10} p_i}$ to represent this relationship. A similar relationship was used in [25]. 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 (8) 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 Section 2.2.

The optimization problem $\mathscr{P}_{TLS}$ can now be formally stated as follows:

$\mathscr{P}_{TLS}$: Given a TAM architecture for a core-based SoC and an upper limit on the SoC test time, determine the total number of test patterns to be applied to each core such that: (i) the overall testing time on each TAM partition does not exceed the upper bound T_{max} and (ii) the defect screening probability $\mathcal{P}(\bar{B}_i)$ 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 $\sum_{i=1}^{p_i} \delta_{ij} = 1$. 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:

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

We next use the Taylor series expansion $\ln(1-x) = -\left(x + \frac{x^2}{2} + \frac{x^3}{3} + \cdots\right)$ 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:

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

In other words, our objective function can be stated as

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

Next we determine the constraints imposed by the upper limit on the SoC test time. Suppose the SoC-level TAM architecture consists of **B** TAM partitions. 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 [6] 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 $T_i(j) = (1 + \max\{s_i, s_o\} \cdot j + \min\{s_i, s_o\})$ [6]. 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 number of variables and constraints for a given ILP model determines the complexity of the problem. The number of variables in the ILP model is only $N + N\sum_{i=1}^{N} p_i$, and the number of constraints is only $\sum_{i=1}^{N} N \cdot p_i + B$; thus this exact approach is scalable for large problem instances. The complete ILP model is shown as Figure 2.

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

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

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

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. 2. Integer Linear Programming model for $\mathscr{P}_{TLS}$.

4. Experimental Results

In this section, we present experimental results for five SoCs from the ITC'02 SoC test benchmark suite [20]. 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-constrained wafer sort test environment, and at the same time maximize the defect-screening probability for the SoC, we present the following results:

- Given values of W and T_{max} relative to T_{SoC}, the percentage of test patterns for each individual core that must be applied at wafer sort to maximize the defect-screening probability for the SoC.
- 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 are presented in Figures 3-7 for three values of T_{max}: $0.75T_{SoC}$, $0.50T_{SoC}$, and $0.25T_{SoC}$. For the three large "p" SoCs from Philips, we select the value of **B** that minimizes the SoC package test time. The results show that the fraction of patterns applied per core, while close to 100% in many cases, varies significantly in order to maximize the SoC defect-screening probability. The maximum value of TAM width W (in bits) is set to 32 and we repeat the optimization procedure for all TAM widths ranging from

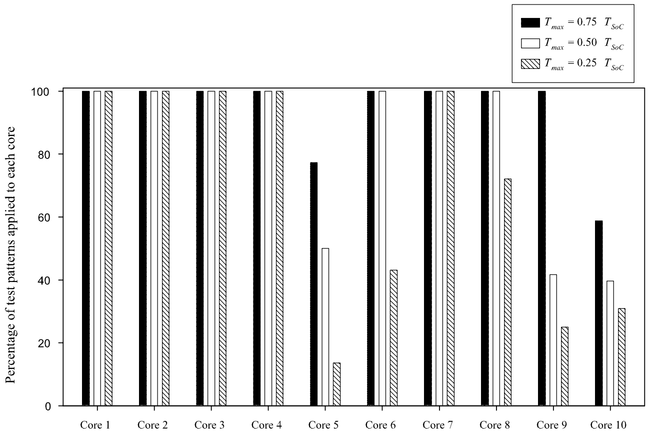

Fig. 3. Percentage of test patterns applied to each core in d695 for W = 8.

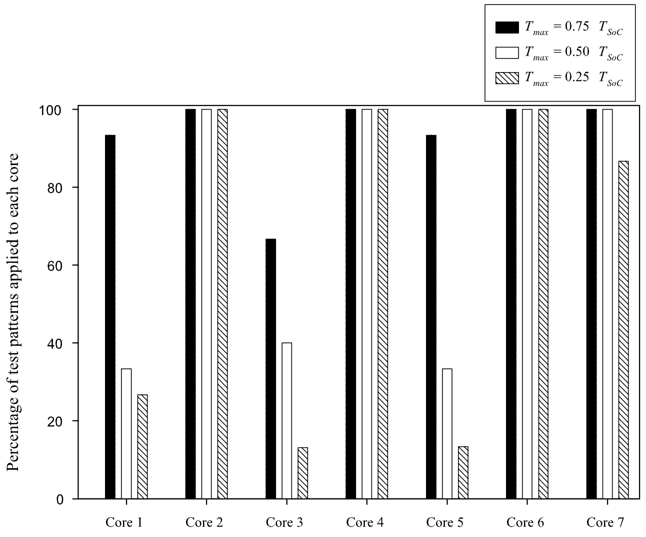

Fig. 4. Percentage of test patterns applied to each core in a586710 for W = 8.

8 to 32 in steps of eight. Results are reported only for $W = 8$; similar results are obtained for other values of W. The CPU time for *lpsolve* for the largest SoC benchmark was less than a second.

We next present the defect-screening probabilities for all the individual cores in the benchmark SoCs (Figures 8-11). The cores that are more likely to lead to fails during wafer sort exhibit higher defect screening probabilities, and vice versa. The values of the defect screening probabilities $\mathcal{P}_S$ of the benchmark SoCs obtained using our ILP-based model for varying TAM widths, as well as overall test time are summarized in Table II. The results show that a significant percentage of the faulty dies can be screened at wafer sort using our proposed technique.

5. Conclusions

We have formulated a test-length selection problem for wafer-level testing of core-based SoCs. To the best of our knowledge, this is the first attempt to formulate a test-length selection problem for wafer sort of core-based SoCs. To solve this problem, we first showed how defect probabilities for the individual cores in an SoC can be obtained using statistical modeling techniques. The defect probabilities were then used in an ILP model to solve the test-length selection problem. The ILP approach takes less than a second for the largest SoC test benchmarks from Philips. Experimental results for the ITC'02 SoC test benchmarks show that the ILP-based method can contribute significantly to defect-screening at wafer sort.

6. Acknowledgements

We thank Sandeep Kumar Goel and Erik Jan Marinissen of Philips Research Labs for providing us with the layout information for the ITC'02 SoC Test benchmarks.

7. References

[1] Y. Zorian, E. Marinissen, and S. Dey, "Testing embedded core-based system chips," *IEEE Computer*, vol. 32, pp. 52–60, June 1999.

[2] Int. Technology Roadmap for Semiconductors, 2005 http://www.itrs.net/Common/2005ITRS/Home2005.htm.

[3] IEEE Std. 1500. IEEE Standard for Embedded Core Test - IEEE Std. 1500-2004. IEEE, New York, 2004.

[4] C. P. Su and C. W. Wu, "A graph-based approach to power-constrained SoC test scheduling," *Journal of Electronic Testing: Theory and Applications*, vol. 19, pp. 45–60, February 2004.

[5] V. Iyengar, K. Chakrabarty, and E. Marinissen, "Test access mechanism optimization, test scheduling and tester data volume reduction for system-on-chip," *IEEE Transactions on Computers*, vol. 52, pp. 1619–1632, December 2003.

[6] ——, "Test wrapper and test access mechanism co-optimization for system-on-chip," *Journal of Electronic Testing Theory and Applications*, vol. 18, pp. 213–230, April 2002.

[7] E. Larsson, K. Arvidsson, H. Fujiwara, and Z. Peng, "Efficient test solutions for core-based designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, pp. 758–775, May 2004.

[8] Q. Xu and N. Nicolici, "Modular SoC testing with reduced wrapper count," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 1894–1908, December 2005.

[9] W. Zou, S. M. Reddy, and I. Pomeranz, "SoC test scheduling using simulated annealing," in *Proc. of the VLSI Test Symposium*, 2003, pp. 325–330.

[10] S. K. Goel and E. J. Marinissen, "Effective and efficient test architecture design for SOCs," in *Proc. of the Int. Test Conference*, 2002, pp. 529–538.

[11] E. Larsson et al., "Defect-aware SoC test scheduling," in *Proc of the VLSI Test Symposium*, 2004, pp. 228–233.

[12] U. Ingelsson et al., "Test scheduling for modular SOCs in an abort-on-fail environment," in *Proc of the European Test Symposium*, 2005, pp. 8–13.

[13] A. B. Kahng, "The road ahead: The significance of packaging," *IEEE Design and Test*, vol. 52, pp. 1619–1632, December 2003.

[14] P. C. Maxwell, "Wafer-package test mix for optimal defect detection and test time savings," *IEEE Design and Test of Computers*, vol. 20, pp. 84–89, September 2003.

[15] I. Koren, Z. Koren, and C. H. Strapper, "A unified negative-binomial distribution for yield analysis of defect-tolerant circuits," *IEEE Transactions on Computer*, vol. 42, pp. 724–734, June 1993.

[16] I. Koren and C. H. Strapper, *Yield Models for Defect Tolerant VLSI circuits: A Review*. Plennum, New York, 1989.

[17] C. H. Strapper, "Small-area fault clusters and fault-tolerance in VLSI systems," *IBM J. Res. Develop*, vol. 33, pp. 174–177, March 1989.

[18] J. A. Cunningham, "The use and evaluation of yield models in integrated circuit manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 3, pp. 60–71, May 1990.

[19] S. K. Goel and E. J. Marinissen, "Layout-driven SoC test architecture design for test time and wire length minimization," in *Proc. of the Design Automation and Test Conference*, 2003, pp. 10 738–10 743.

[20] E. J. Marinissen et al., "A set of benchmarks for modular testing of SOCs," in *Proc. of the IEEE Int. Test Conference*, 2002, pp. 519–528.

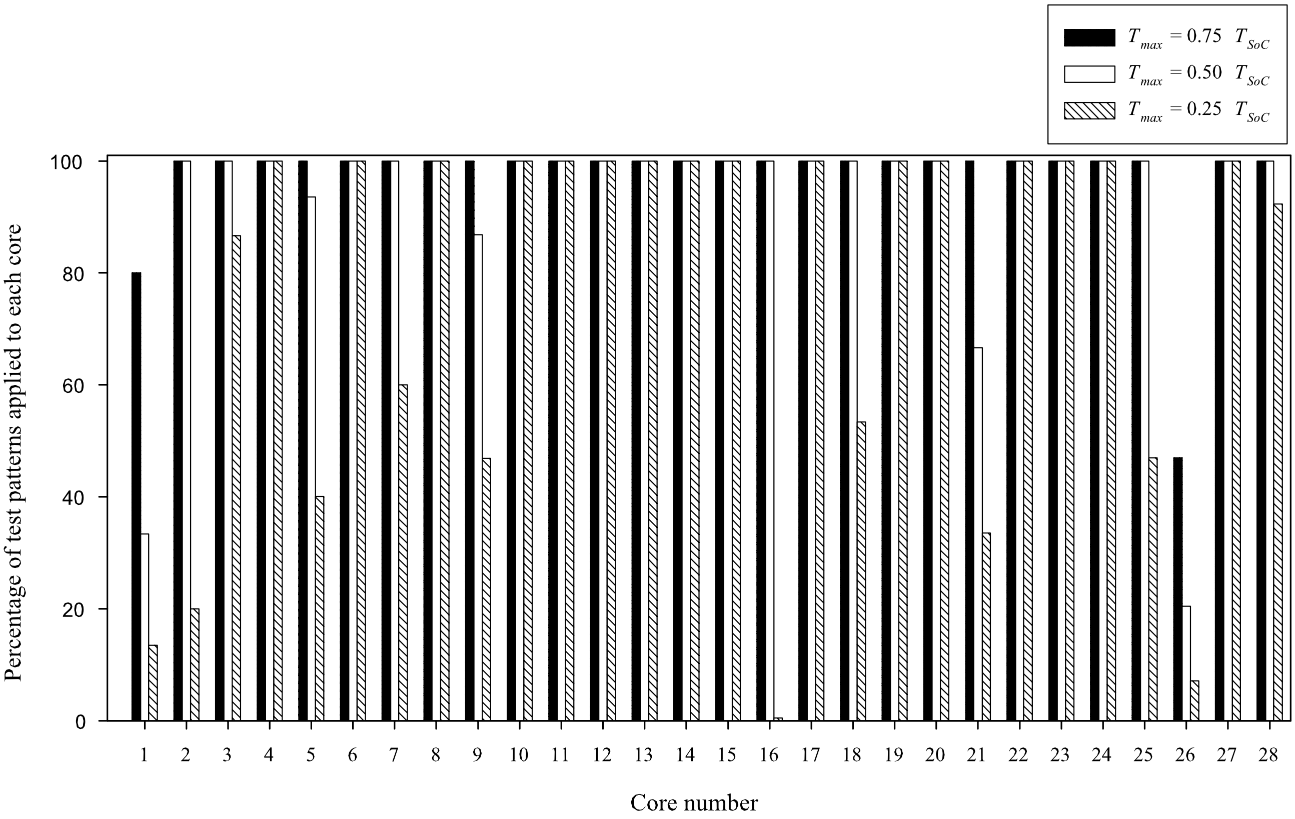

Fig. 5. Percentage of test patterns applied to each core in p22810 for W = 8.

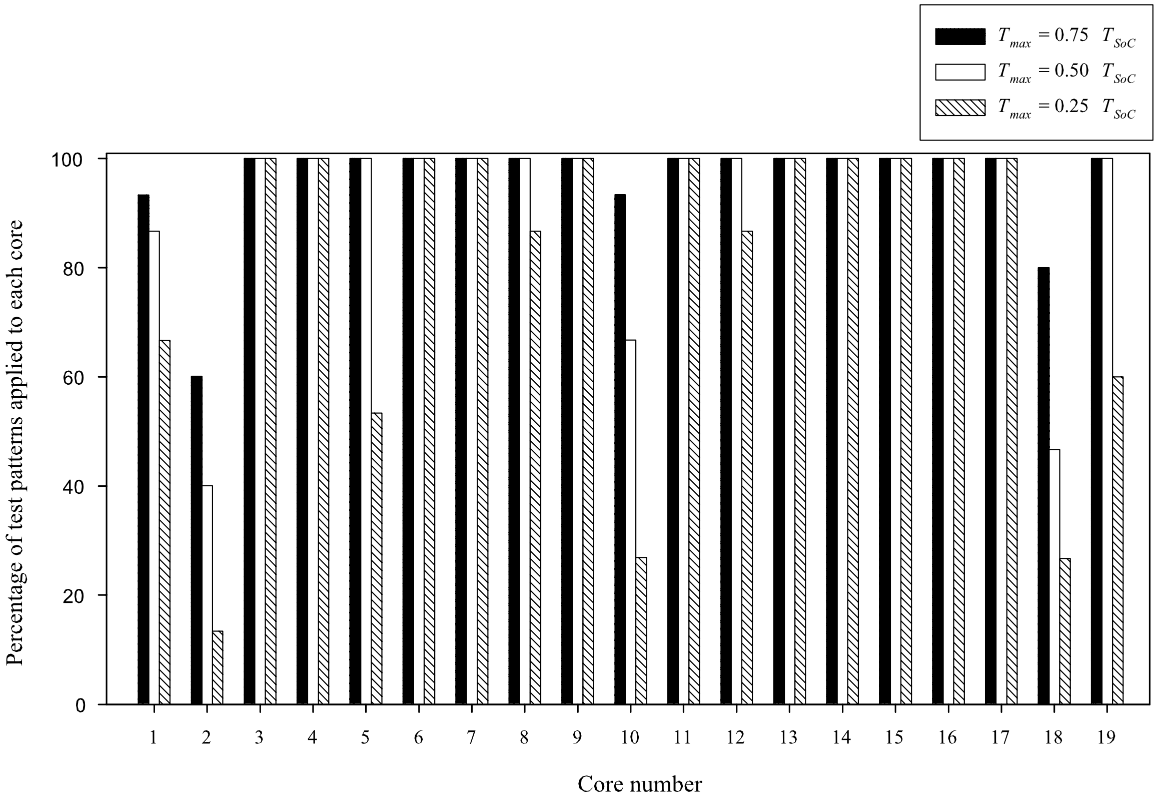

Fig. 6. Percentage of test patterns applied to each core in p34392 for W = 8.

[21] E. Larsson and Z. Peng, “An integrated framework for the design and optimization of SoC test solutions,” *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 385–400, February 2002.

[22] V. Iyengar and K. Chakrabarty, “Test bus sizing for system-on-a-chip,”

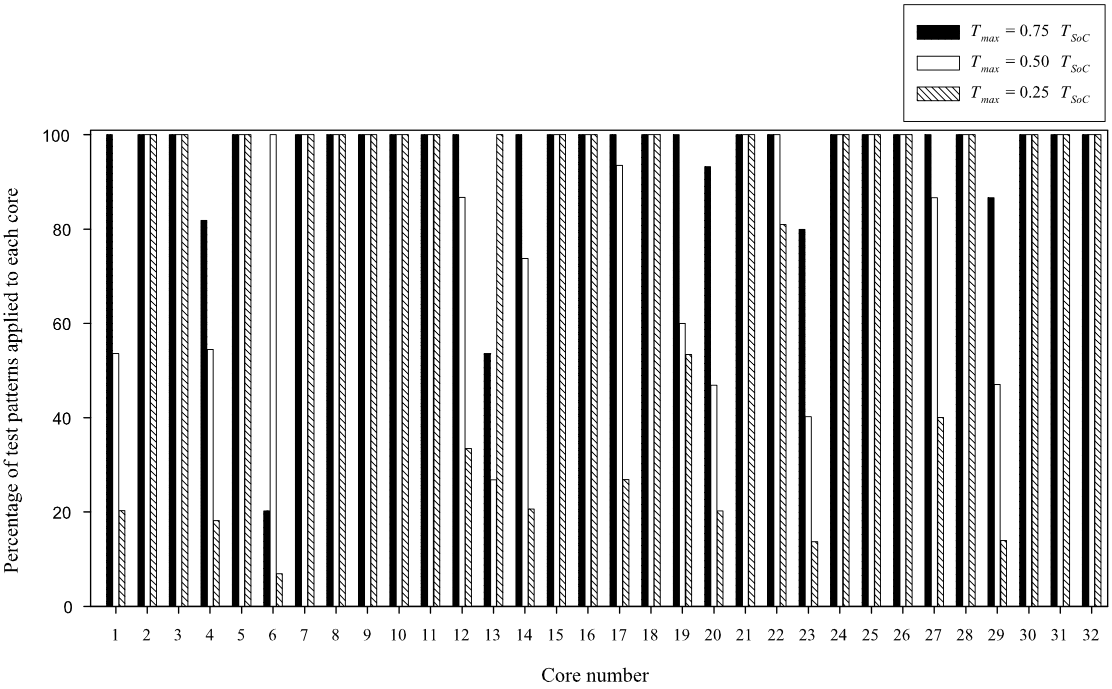

Fig. 7. Percentage of test patterns applied to each core in p93791 for W = 8.

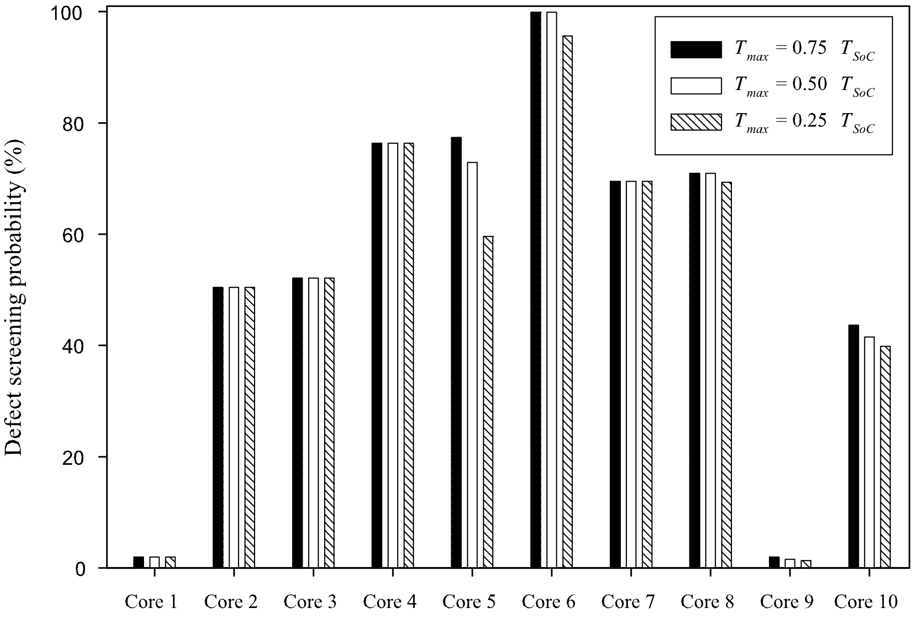

Fig. 8. Relative defect screening probabilities for the individual cores in d695 for W = 8.

IEEE Transactions on Computers, vol. 51, pp. 449–459, May 2005.

[23] S. Koranne, "A novel reconfigurable wrapper for testing of embedded core-based socs and its associated scheduling algorithm," *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 415–434, August 2002.

[24] M. Bushnell and V. Agarwal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers, 2000.

[25] E. Larsson and H. Fujiwara, "Optimal system-on-chip test scheduling," in *Proc. of the Asian Test Symposium*, 2004, pp. 306–311.

[26] E. Kreyszig, *Advanced Engineering Mathematics*, 8th ed. John Wiley & Sons Inc., 1998.

[27] M. Berkelaar et al., "lpsolve: Open source (mixed-integer) linear programming system". Version 5.5 dated May 16, 2005
URL: http://www.geocities.com/lpsolve.

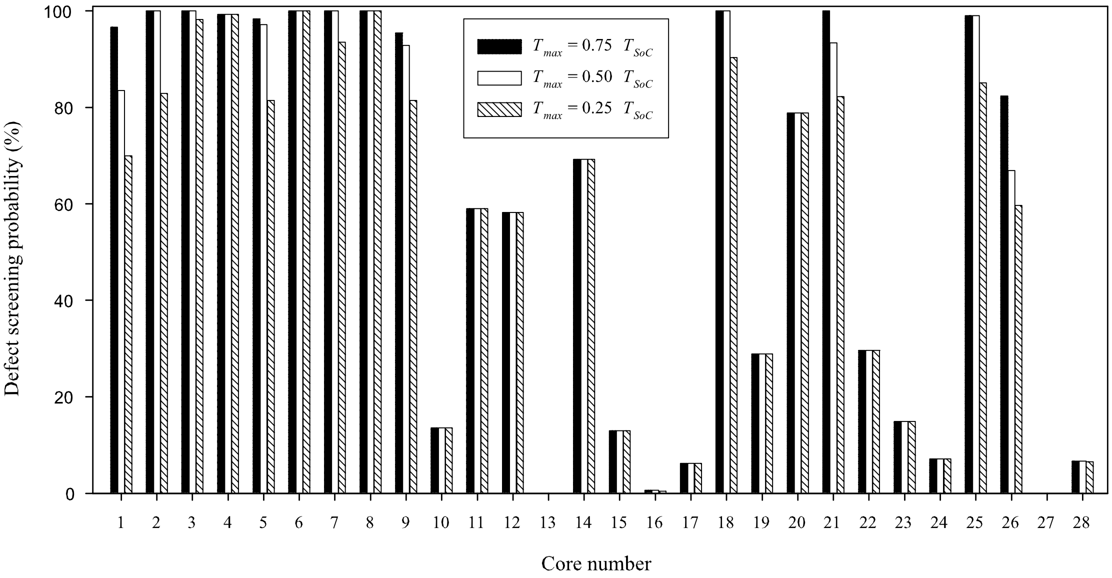

Fig. 9. Relative defect screening probabilities for the individual cores in p22810 for W = 8.

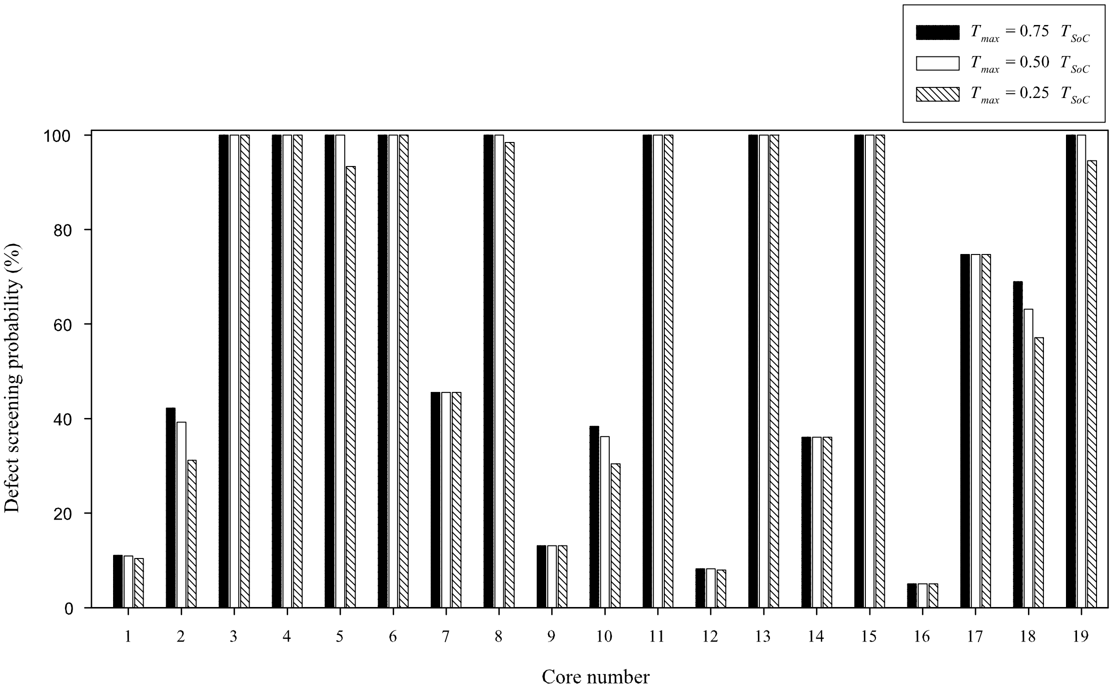

Fig. 10. Relative defect screening probabilities for the individual cores in p34392 for W = 8.

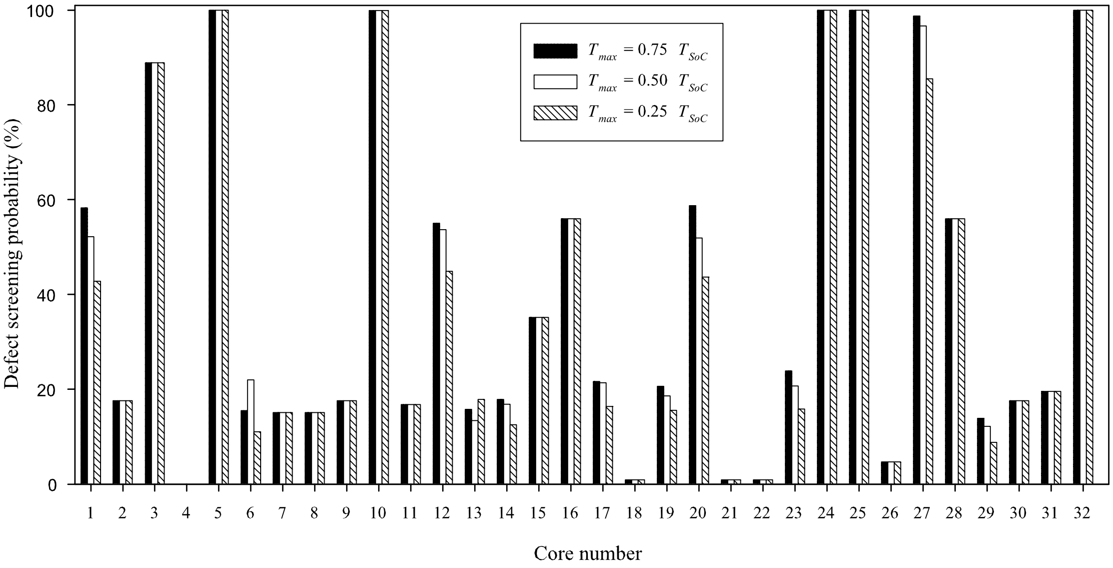

Fig. 11. Relative defect screening probabilities for the individual cores in p93791 for W = 8.

SoC	W	$T_{max} = 0.75\ T_{SoC}$	$T_{max} = 0.5\ T_{SoC}$	$T_{max} = 0.25\ T_{SoC}$
d695	8	0.9229	0.6487	0.4095
	16	0.9229	0.6487	0.4308
	24	0.9047	0.5985	0.3604
	32	0.8765	0.5245	0.1666
p22810	8	0.7693	0.5947	0.0969
	16	0.8137	0.5996	0.1699
	24	0.7871	0.334	0.0105
	32	0.7656	0.3435	0.0414
p34392	8	0.8661	0.6869	0.3521
	16	0.8807	0.7118	0.2157
	24	0.899	0.7207	0.2569
	32	0.9161	0.6715	0.2278
p93791	8	0.4883	0.2299	0.0097
	16	0.5341	0.2438	0.0168
	24	0.7234	0.2535	0.0826
	32	0.7098	0.3335	0.0548

TABLE II
Relative Defect Screening Probabilities Obtained Using the ILP technique.