Optimal Test Access Architectures for System-on-a-Chip

KRISHNENDU CHAKRABARTY
Duke University

Test access is a major problem for core-based system-on-a-chip (SOC) designs. Since embedded cores in an SOC are not directly accessible via chip inputs and outputs, special access mechanisms are required to test them at the system level. An efficient test access architecture should also reduce test cost by minimizing test application time. We address several issues related to the design of optimal test access architectures that minimize testing time., including the assignment of cores to test buses, distribution of test data width between multiple test buses, and analysis of test data width required to satisfy an upper bound on the testing time. Even though the decision versions of all these problems are shown to be NP-complete, they can be solved exactly for practical instances using integer linear programming (ILP). As a case study, the ILP models for two hypothetical but nontrivial systems are solved using a public-domain ILP software package.

Categories and Subject Descriptors: B.7.1 [**Integrated Circuits**]: Types and Design Styles; B.7.2 [**Integrated Circuits**]: Design Aids; B.7.3 [**Integrated Circuits**]: Reliability and Testing

General Terms: Design, Reliability

1. INTRODUCTION

Embedded cores are now increasingly being used in large system-on-a-chip (SOC) designs [Zorian et al. 1998]. These complex, predesigned functional blocks facilitate design reuse, allow greater on-chip functionality, and lead to shorter product development cycles. However, the manufacturing test and debug of such SOC designs remains a major challenge. Since embedded cores are not directly accessible via chip inputs and outputs, special access mechanisms are required to test them at the system level. The development of efficient test access architectures is therefore of considerable interest to the SOC design and test community.

This research was supported in part by the National Science Foundation under grant CCR-9875324. An abridged version of this paper appeared in *Proceedings of the IEEE VLSI Test Symposium*, Montreal, Canada, May 2000, pp. 127-134.

Author's address: Department of Electrical and Computer Engineering, Duke University, 130 Hudson Hall, Box 90291, Durham, NC 27708; email: krish@ee.duke.edu.

A test access architecture, also referred to as a test access mechanism (TAM), provides means for on-chip test data transport [Zorian et al. 1998]. It can be used to transport test patterns from a pattern source to a core-under-test, and to transport test responses from a core-under-test to a response monitor. A number of test access architectures have been proposed in the literature [Zorian et al. 1998]. These include macro test [Marinissen and Lousberg 1999]; core transparency [Ghosh et al. 1993; Ghosh et al. 1998]; dedicated test bus [Varma and Bhatia 1998]; multiplexed access [Immaneni and Raman 1990]; and a bus architecture based on the concept of a TESTRAIL [Marinissen et al. 1998]. A TESTRAIL provides a flexible and scalable test access mechanism; a single TESTRAIL can provide access to one or more cores, and an IC may contain one or more TESTRAILs of varying widths. The width of a TESTRAIL is referred to as the test data width, since it determines the overall system testing time. Figure 1, derived from Marinissen et al. [1998], illustrates one possible implementation of the TESTRAIL architecture. (The core wrapper and the bypass mechanism are explicitly not shown in the figure.)

In order to reduce test cost, the testing time for a core-based system should be minimized by adopting an appropriate test access architecture. While the TESTRAIL architecture allows the system designer to trade-off testing time with area overhead by varying tests data widths, the precise relationship between the testing time and the test access architecture has not been formally studied. Related prior work has either been limited to test scheduling for a given test access mechanism [Chakrabarty 1999; 2000a; Sugihara et al. 1998], or to determining the optimal number of internal scan chains in the cores [Aerts and Marinissen 1998]. The latter requires redesign of the scan chains for each customer, and thereby affects core reuse. While Aerts and Marinissen [1998] present several novel strategies for TAM design (e.g. multiplexing, daisy chaining, and distribution), it does not directly address the problem of optimal sizing of test buses in the SOC. We are interested here in the problem of minimizing the SOC testing time via optimal test bus design, and without any redesign of the embedded cores. The design of the test access architecture is especially important for the system designer/integrator, since the IEEE P1500 standard (being developed for embedded core testing) leaves TAM design upto the system integrator [Marinissen et al. 1999].

The system integrator is interested in the following TAM design problems: (1) Given an SOC and maximum test data width, how should the width be distributed among the various test buses in order to minimize the testing time? (2) How should the embedded cores in the system be assigned to the test buses? (3) For a given test access architecture, how much test data width is required to meet specified testing time objectives? To the best of our knowledge, this paper presents the first systematic solutions to these SOC design problems.

The main contributions of the paper are listed below.

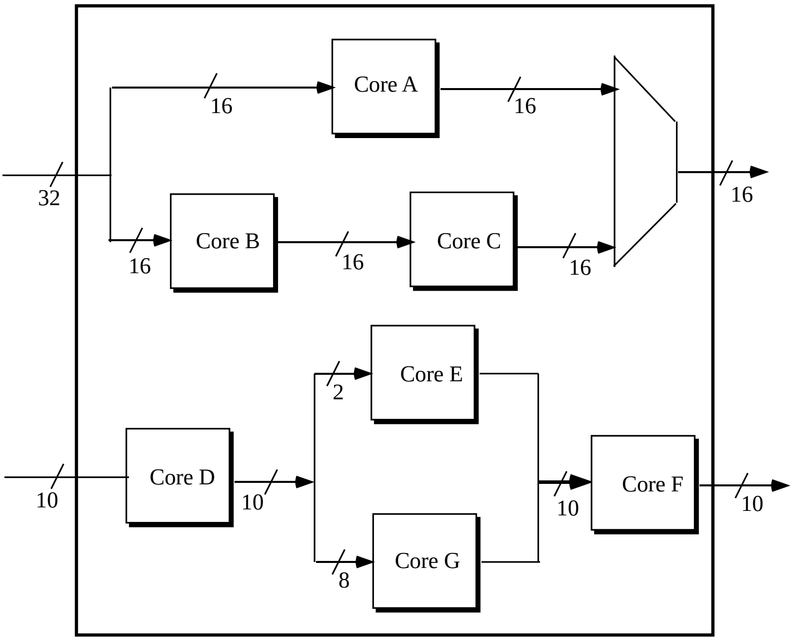


Fig. 1. An example of the TESTRAIL architecture.

- We formulate several design problems related to test access architectures, and show that the decision versions of all these problems are NP-complete.
- Even though these design problems are NP-complete, we show that they can be solved exactly using integer linear programming (ILP). We first develop an ILP model for optimally assigning cores to test buses when the widths of these test buses are known. We refer to this as the "test bus assignment problem".
- We note that the testing time can be reduced further by optimally distributing the total test width among the individual test buses. We develop an ILP model for minimizing the testing time by combining optimal width distribution with optimal test bus assignment.
- Given a constraint on the maximum testing time, we develop an ILP model to determine the minimum test data width and an optimal assignment of cores to test buses.
- The above ILP models make the simplifying assumption that a test bus cannot be subdivided into test buses of smaller width which subsequently merge before the test data is transported to the IC outputs. In order to account for this realistic scenario, we refine our ILP models to allow a test buses to fork into test buses of smaller width.
- We evaluate the feasibility of the proposed ILP models by solving them using an ILP solver for two hypothetical, but nontrivial and representative SOCs. The experimental results demonstrate that optimal solutions to these important design problems in SOC testing are indeed feasible.

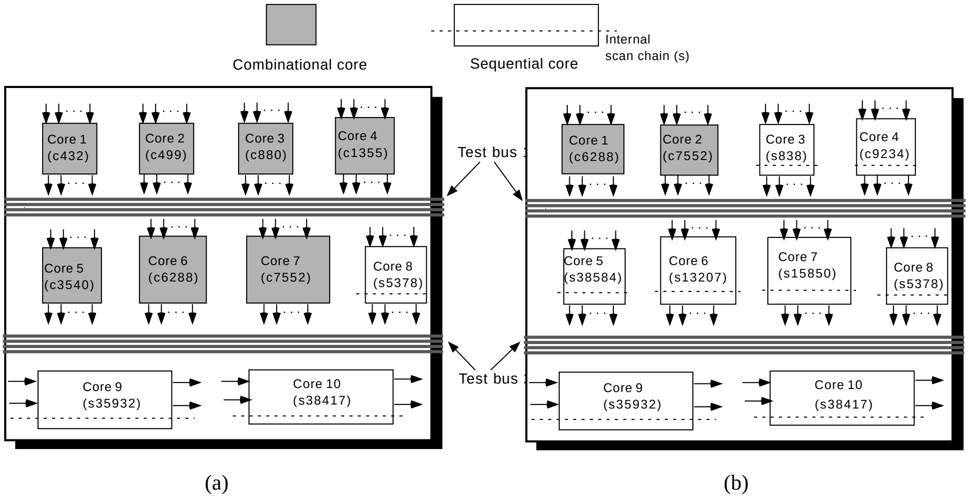


Fig. 2. Two examples of core-based SOCs with two test buses each: (a) System $\mathcal{S}_1$ containing seven combinational cores and three cores with internal scan; (b) System $\mathcal{S}_2$ containing two combinational cores and eight cores with internal scan.

Our ILP models do not address the problem of testing the interconnect and wiring between the cores. A complete solution for SOC testing that also addresses these issues requires enhancements to the basic ILP framework presented here for isolation testing of embedded cores.

The organization of the paper is as follows. In Section 2 we briefly review integer linear programming and formulate the problem of optimal test bus assignment. In Section 3 we develop ILP models for minimizing the testing time by determining an optimal test width distribution. In Section 4 we present case studies for two example SOCs (described below). We solve the various ILP models for this system using the *lpsolve* software package from Eindhoven University of Technology [Berkelaar 1999]. Finally, in Section 5, we extend our basic ILP models to handle cases where a test bus may fork into several test buses that subsequently merge before the test data is transported to the IC outputs. We present experimental results on optimal and near-optimal subdivision of the test buses.

In order to illustrate the proposed optimization methods, throughout the paper, we use as examples the core-based SOCs $\mathcal{S}_1$ and $\mathcal{S}_2$ shown in Figure 2. These hyopthetical but nontrivial SOCs consist of ten ISCAS 85 [Brglez and Fujiwara 1985] and ISCAS 89 benchmark circuits [Brglez et al. 1989] each. We assume that the three ISCAS 89 circuits contain internal scan chains. $\mathcal{S}_1$ contains seven combinational cores and seven sequential cores, while $\mathcal{S}_2$ consists of two combinational cores and eight sequential cores. The complexity of the ILP models depends more on the number of cores in the SOC than on the sizes of the cores. For the sake of illustration, only two test buses are shown in Figure 2. Our ILP models can easily be used for any number of test buses.

2. OPTIMAL ASSIGNMENT OF CORES TO TEST BUSES

We first briefly review ILP by means of matrix notation [Williams 1985]. The goal of ILP is to minimize a linear objective function on a set of integer variables while satisfying a set of linear constraints. A typical ILP model is described as follows:

$$\text{minimize} : \mathbf{Ax}$$

$$\text{subject to} : \mathbf{Bx} \leq \mathbf{C}, \mathbf{x} \geq 0,$$

where **A** is a cost vector, **B** is a constraint matrix, **C** is a column vector of constants, and **x** is a vector of integer variables. Efficient ILP solvers are now readily available, both commercially and in the public domain. For our experiments (described in Sections 4 and 5), we used the *lpsolve* package from Eindhoven University of Technology in the Netherlands.

Let the SOC design consist of N_C cores and let core i, $1 \leq i \leq N_C$ have n_i inputs and m_i outputs. We assume that the n_i inputs of core i include data inputs and scan inputs. Similarly, the m_i outputs of core i include data outputs and scan outputs. Each full or partial scan core may have one or more internal scan chains. (A combinational or nonscan legacy core has no scan inputs and outputs.)

Test data serialization is required at core I/Os when the width of the test bus is less than the number of core terminals [Marinissen et al. 1999]. This happens often because the number of core terminals is determined by the functionality of the core, while test bus width is determined by the test pattern source, the SOC routing and area constraints and, in some cases, the width of the existing system bus. Therefore, test data serialization must be performed in the test wrapper if the number of core terminals is larger than the test bus width.

We assume in our serialization model that the test sets for the SOC cores are available in scan format, in which the functional input values remain unchanged during successive scan cycles. (If the test sets for the cores are obtained before the translation to scan format, then alternative serialization models involving the lengths of the scan chains in the cores can be used to reduce the testing time even further.) The amount of test data serialization necessary at the inputs and outputs of core i is therefore determined by its *test width* $\phi_i = \max\{n_i, m_i\}$. This influences the testing time for core i. We assume that core i requires t_i (scan) cycles for testing. Finally, we assume that the system contains N_B test buses with widths $w_1, w_2, \ldots, w_{N_B}$.

The problem that we address in this section is to minimize the system testing time by optimally assigning cores to test buses. It is formally stated as follows:

- $\mathcal{P}1$: Given N_C cores and N_B test buses of widths $w_1, w_2, \ldots, w_{N_B}$, respectively, determine an assignment of cores to test buses such that the total testing time is minimized.

Note that $\mathcal{P}1$ is equivalent to the well-known multiprocessor scheduling problem, and is therefore NP-complete ([Garey and Johnson 1979, p. 65]). The multiprocessor scheduling problem is stated as follows:

Instance: A finite set A of "tasks," a "length" $l(a) > 0$ for each $a \in A$, a number $m > 0$ of "processors", and a deadline, $D > 0$.

Question: Is there a partition $A = A_1 \cup A_2 \cup \cdots \cup A_m$ of A into m disjoint sets such that $\max\{\sum_{a \in A} l(a) : 1 \leq i \leq m\} \leq D$?

The equivalence between a decision version of $\mathcal{P}1$ and multiprocessor scheduling can easily be established by noting the correspondence between processors and test buses and between tasks and test sets. The deadline D corresponds to the overall SOC testing time.

Even though $\mathcal{P}1$ is NP-complete, we show that, as in the case of many other NP-complete problems, it can be solved exactly for practical instances using integer linear programming. We assume for now that a test bus does not fork (split) into multiple branches that may merge later. This restriction is removed in Section 5. We also assume that all cores on any given test bus are tested sequentially. Two or more test buses can be used simultaneously for delivering test data to cores and for propagating test responses. We assume that the number of test buses (and thereby the amount of test parallelism) is determined by the core user (system integrator) after a careful consideration of system-level I/O, area, and power dissipation issues.

We first note that if core i is assigned to bus j, then the testing time for core i is given by

$$T_{ij} = \begin{cases} t_i & \text{if } \phi_i \leq w_j \\ (\phi_i - w_j + 1)t_i & \text{if } \phi_i > w_j \end{cases} \tag{1}$$

If $\phi_i > w_j$ then the width of the test bus is insufficient for parallel loading of test data, and serialization is necessary at the inputs and/or outputs of core i. In order to calculate the test time due to serialization, we use an interconnection strategy similar to the one suggested in [Marinissen et al. 1998] for connecting core I/Os to the test bus, namely, provide direct (parallel) connection to core I/Os that transport more test data. We assume a worst-case scenario of test data serialization, in which the first $(w_j - 1)$ test bus lines are connected to $(w_j - 1)$ core I/Os in parallel and the last test bus line is serially connected to the remaining $(\phi_i - w_j + 1)$ core I/Os; see Figure 3. This can potentially reduce the amount of interconnect within the wrapper. If the width of bus j is adequate, i.e. $\phi_i \leq w_j$, then no serialization is necessary, and core i can be tested in exactly t_i cycles.

Let x_{ij} be a 0-1 variable defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if core } i \text{ is assigned to bus } j \\ 0 & \text{otherwise} \end{cases}$$

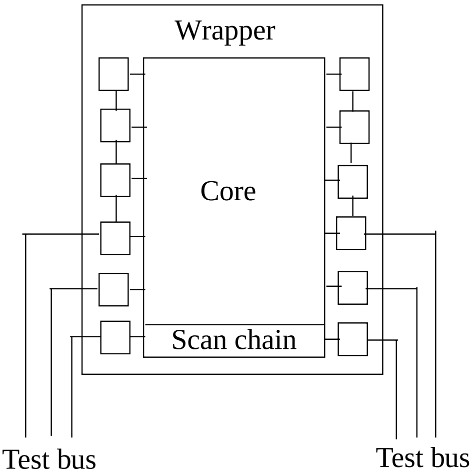


Fig. 3. The serialization model.

The time needed to test all cores on bus j is therefore $\sum_{i=1}^{N_C} T_{ij}x_{ij}$. Since all the test buses can be used simultaneously for testing, the system testing time equals $\max_{j \in \{1, 2, \ldots N_B\}} \sum_{i=1}^{N_C} T_{ij}x_{ij}$. We now formulate an integer programming model for minimizing the system testing time.

Objective: Minimize the cost function $C = \max_{j \in \{1, 2, \ldots N_B\}} \sum_{i=1}^{N_C} T_{ij}x_{ij}$ subject to

(1) $\sum_{j=1}^{N_B} x_{ij} = 1,\ 1 \le i \le N_C\ ,\ 1 \le i \le N_C$

(2) $x_{ij} = 0$ or 1

The above minmax nonlinear cost function can easily be linearized [Williams 1985]. The resulting integer linear programming model is shown in Figure 4. It can be easily seen that the integer linear program ILP model for $\mathcal{P}1$ contains N_BN_C 0-1 variables, one nonbinary, integer variable, and $N_B + N_C + 2N_BN_C$ constraint inequalities.

As an example, we consider the SOCs $\mathcal{S}_1$ and $\mathcal{S}_2$ introduced in Section 1. Table I presents the test data for each embedded core in these systems. We assume that s838 contains one internal scan chain, and s5378 and s9234 contain 4 internal scan chains each. We also assume that s35392 and s38417 contain 32 internal scan chains each, and s13207 and s15850 contain 16 scan chains each. For the combinational cores, $1 \le i \le 7$, the number of test cycles t_i is equal to the number of test patterns p_i. However, for the remaining three cores with internal scan, $t_i = (p_i + 1)\lceil f_i/N_i \rceil + p_i$, where core i contains f_i flip-flops and n_i internal scan chains [Aerts and Marinissen 1998]. The test patterns for these circuits were obtained from [Hamzaoglu and Patel 1998].

Example. Let $N_B = 2$, and let the total test data width for $\mathcal{S}_1$ be 48 bits, i.e. $w_1 + w_2 = 48$. In addition, let $w_1 = 32$ and $w_2 = 16$. The optimal

Minimize C subject to:

1. $C \geq \sum_{i=1}^{N_C} T_{ij} x_{ij},\ 1 \leq j \leq N_B$
2. $\sum_{j=1}^{N_B} x_{ij} = 1,\ 1 \leq i \leq N_C$
3. $x_{ij} = 0$ or 1

Fig. 4. Integer linear programming model for $\mathcal{P}1$.

Table I. Test Data for Cores in $\mathcal{S}_1$ and $\mathcal{S}_2$

Circuit (core)	i	Number of test inputs n_i	Number of test outputs m_i	$\phi_i =$ max$\{n_i, m_i\}$	Number of test patterns p_i	Number of test cycles t_i
c432	1	36	7	36	27	27
c499	2	41	32	41	52	52
c880	3	60	26	60	16	16
c1355	4	41	32	41	84	84
c3540	5	50	22	50	84	84
c6288	6	32	32	32	12	12
c7552	7	207	108	207	73	73
s5378	8	39	53	53	97	4507
s35932	9	67	352	352	12	714
s38417	10	60	138	138	68	3656

(a)

Circuit (core)	i	Number of test inputs n_i	Number of test outputs m_i	$\phi_i =$ max$\{n_i, m_i\}$	Number of test patterns p_i	Number of test cycles t_i
c6288	1	32	32	32	12	12
c7552	2	207	108	207	73	73
s838	3	36	3	36	75	2507
s9234	4	40	43	43	105	5723
s38584	5	70	336	336	110	5105
s13207	6	78	168	168	234	9634
s15850	7	93	166	166	95	3359
s5378	8	39	53	53	97	4507
s35932	9	67	352	352	12	714
s38417	10	60	138	138	68	3656

(b)

assignment of cores to these two test buses is given by the vector (1,1,1,1,1,1,1,2,2,1), where a 1 (2) in position i of the vector indicates that core i is assigned to bus 1 (2). This is shown in Figure 5. The optimal testing time for these values of w_1 and w_2 obtained using *lpsolve* is 411884 cycles. Note that this is not the minimum testing time that can be achieved with a total test width of 48 bits. For example, a testing time of 408077 cycles is achieved using $w_1 = 28$, $w_2 = 20$, and the test bus assignment vector (1,1,2,1,2,1,2,2,2,1). In the next section we examine the problem of

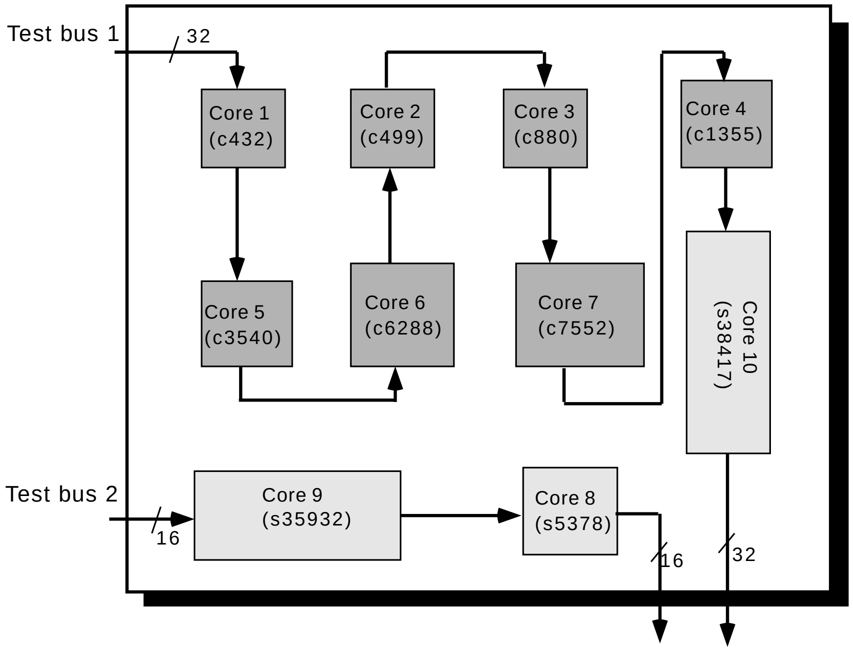


Fig. 5. Optimal test bus assignment for system $\mathcal{S}_1$ with two test buses of 32 bits and 16 bits, respectively.

determining an optimal distribution of the total test data width among the individual test buses.

The following theorem presents a lower bound on the total testing time when the widths of the test buses are known. This lower bound can indeed be achieved in practice—we illustrate this below using the system $\mathcal{S}_1$ as an example. We also make use of this theorem in Section 3 to derive a lower bound on the testing time when only the total test data width is known and the optimal widths of the test buses have to be determined.

THEOREM 1. *For a SOC with N_C cores and N_B test buses with widths $w_1, w_2, \ldots w_{N_B}$, respectively, a lower bound on the total testing time $\mathcal{T}$ is given by $\mathcal{T} \geq \max_i\{\min_j\{T_{ij}\}\}$ where ϕ_i is the test width of core i and T_{ij} is defined by (1).*

PROOF. The testing time for core i depends on the width of the test bus to which it is assigned. Clearly, the testing time for core i is at least $\min_j\{T_{ij}\}$. Since the overall system testing time is determined by the core that has the longest test time, $\mathcal{T} \geq \max_i\{\min_j\{T_{ij}\}\}$. □

For the system $\mathcal{S}_1$ with two test buses of 32 bits and 16 bits, respectively, Theorem 1 provides a lower bound on the testing time of 391,190 cycles. This corresponds to a test bus assignment in which only core 10 is assigned to the 32-bit first test bus. Such an assignment is indeed optimal and the

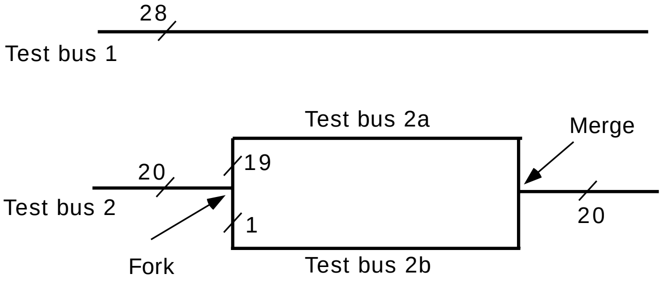


Fig. 6. A test bus architecture involving fork and merge of test buses.

lower bound of Theorem 1 is achieved if the width of the second test bus is increased, or if a third test bus is used.

The ILP model presented in this section can also be used for optimally assigning cores to test buses for more general test access architectures. For example, Figure 6 shows a test access architecture consisting of two test buses in which the 20-bit test bus forks into two sets of buses, which in turn merge into the original 20-bit-wide test bus. If we use this test bus architecture for $\mathcal{S}_1$, then a minimum testing time of 407,991 cycles is obtained using the test bus assignment vector (1,2,2b,2,1,2a,2,2,2,1). A more general discussion of this problem is presented in Section 5.

3. OPTIMAL TEST BUS WIDTH

In this section we examine the problem of minimizing system testing time by determining (i) optimal widths for the test buses, and (ii) optimal assignment of cores to test buses. This generalizes the optimization problem discussed in Section 2. We assume that the total system test width can be at most W. We also assume that the width of a test bus does not exceed the width required for any given core, i.e., $\max_j\{w_j\} \leq \min_i\{\phi_i\}$ for all values i and j, and test data serialization is required for every core. This assumption is necessary to avoid complex nonlinear models that are difficult to linearize. From a practical point of view, this assumption implies that cores with very small test widths are assigned to test buses after the cores with larger test widths are optimally assigned. We extend the ILP model and remove this restriction in Section 4.

We now formulate the problem of optimally allocating the total width among the N_B buses, as well as determining the optimal allocation of cores to these buses. The optimization problem is formally stated as follows:

- $\mathcal{P}2$: Given N_C cores and N_B test buses of total width W, determine the optimal width of the test buses and an assignment of cores to test buses such that the total testing time is minimized.

THEOREM 2. *The decision version of $\mathcal{P}2$ is NP-complete.*

PROOF. To show that $\mathcal{P}2$ belongs to NP, we consider the following decision problem version of $\mathcal{P}2$: Given N_C cores and N_B test buses of total width W, does there exist a width distribution for the test buses and an assignment of cores to test buses such that the total testing time is less than or equal to $\mathcal{T}$? A nondeterministic algorithm can guess a width distribution and a test bus assignment for the cores and check in polynomial time if the testing time is less than or equal to $\mathcal{T}$. To show that $\mathcal{P}2$ is NP-hard, we use the method of restriction [Garey and Johnson 1979]. Consider an instance of $\mathcal{P}2$ for which the $W = N_B \min_i\{\phi_i\}$. Since the width of a test bus is at least $\min_i\{\phi_i\}$, this implies that every test bus has a width of $\min_i\{\phi_i\}$. This is equivalent to an instance of $\mathcal{P}1$ which, as discussed in Section 2, is NP-complete. Therefore, $\mathcal{P}2$ is NP-hard. □

Even though $\mathcal{P}2$ is NP-complete, the sizes of practical SOC problem instances allow it to be solved exactly. We now present an integer programming model for $\mathcal{P}2$ that allows us to determine optimal widths and an optimal assignment of cores to buses simultaneously. We use the 0-1 variable x_{ij} defined in Section 2.

Minimize C, subject to:

(1) $C \geq \sum_{i=1}^{N_C}(\phi_i - w_j + 1)t_i x_{ij}\ ,\ 1 \leq j \leq N_B$

(2) $\sum_{j=1}^{N_B} x_{ij} = 1\ ,\ 1 \leq i \leq N_C$

(3) $\sum_{j=1}^{N_B} w_j = W\ ,\ 1 \leq j \leq N_B$

(4) $w_j \leq \phi_i\ ,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B$

(5) $x_{ij} = 0$ or 1

Note that constraint (1) above is nonlinear because it contains a product term. We linearize it by replacing the product term $w_j x_{ij}$ with a new integer variable y_{ij} ($y_{ij} \geq 0$) and adding the following three constraints for every such product term:

(1) $y_{ij} - w_{max} x_{ij} \leq 0$, where $w_{max} = W$ is an upper bound on the widths of the test buses.

(2) $-w_j + y_{ij} \leq 0$

(3) $w_j - y_{ij} + w_{max} x_{ij} \leq w_{max}$

The intuitive reasoning behind the above three constraints is as follows. Since x_{ij} can take only 0-1 values, y_{ij} is restricted to be either 0 (if $x_{ij} = 0$) or w_j (if $x_{ij} = 0$). This implies that $0 \leq y_{ij} \leq w_{max}$. The three additional inequalities are necessary and sufficient to constrain the values that y_{ij} can take. This leads us to the (linearized) ILP model for $\mathcal{P}2$ shown in Figure 7.

As expected, the ILP model for $\mathcal{P}2$ is bigger in size than the ILP model for $\mathcal{P}1$. It contains $N_B N_C$ 0-1 variables, $N_C N_B + N_B + 1$ nonbinary integer

Minimize W subject to:

1. $\sum_{j=1}^{N_B} w_j = W$
2. $\sum_{i=1}^{N_C}((\phi_i + 1)t_i x_{ij} - t_i y_{ij}) \leq \mathcal{T}$, $1 \leq j \leq N_B$
3. $y_{ij} - w_{max}x_{ij} \leq 0$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$, where w_{max} is an upper bound on the w_j's.
4. $-w_j + y_{ij} \leq 0$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
5. $w_j - y_{ij} + w_{max}x_{ij} \leq w_{max}$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
6. $\sum_{j=1}^{N_B} x_{ij} = 1$, $1 \leq i \leq N_C$
7. $w_j \leq \phi_i$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
8. $x_{ij} = 0$ or 1

Fig. 7. Integer linear programming model for $\mathcal{P}2$.

variables, and $(6N_BN_C + N_B + N_C + 1)$ constraint inequalities. The ILP model for $\mathcal{P}_2$ is especially useful in determining the effect of increased test data width on the testing time. However, there is a limit to which the testing time can be decreased by simply increasing the system test width. The following theorem provides a lower bound on the testing time $\mathcal{T}$ for a core-based system. It is useful in determining the maximum test width beyond which the testing time cannot be decreased by simply increasing width.

THEOREM 3. *For a core-based system with N_C cores, a lower bound on the total testing time $\mathcal{T}$ is given by* $\mathcal{T} \geq \max_{i\in\{1,2,\ldots,N_C\}}\{(\phi_i - \min_k\{\phi_k\} + 1)t_i\}$.

PROOF. Let the system consist of N_B test buses with (undetermined) test widths $w_1, w_2, \ldots w_{N_B}$ such that $\min_k\{\phi_k\} \geq \max_j\{w_j\}$. We know from Theorem 1 that $\mathcal{T} \geq \max_i\{\min_j\{(\phi_i - w_j+1)t_i\}\}$. Since $\min_j\{(\phi_i - w_j + 1)t_i\} = (\phi_i - \max_j\{w_j\}+1)t_i$ and $\max_j\{w_j\} \leq \min_k\{\phi_k\}$, we have

$$\begin{aligned}\mathcal{T} &\geq \max_i\{(\phi_i - \max_j\{w_j\}+1)t_i\}\\ &\geq \max_i\{(\phi_i - \min_k\{\phi_k\}+1)t_i\}\end{aligned}$$

This completes the proof of the theorem. □

Next, we address the related optimization problem of determining the minimum system test width required to meet a minimum testing time objective. In addition, we determine an optimal distribution of the width among the test buses, and an optimal test bus assignment. The optimization problem is formally stated as follows:

- $\mathcal{P}3$: Given N_C cores, N_B test buses, and a maximum testing time $\mathcal{T}$, determine the minimum total test width, an optimal distribution of the test width among the test buses, and an optimal assignment of cores to test buses.

THEOREM 4. *The decision version of $\mathcal{P}3$ is NP-complete.*

Minimize W subject to:

1. $\sum_{j=1}^{N_B} w_j = W$
2. $\sum_{i=1}^{N_C}((\phi_i + 1)t_i x_{ij} - t_i y_{ij}) \leq \mathcal{T}$, $1 \leq j \leq N_B$
3. $y_{ij} - w_{max} x_{ij} \leq 0$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$, where w_{max} is an upper bound on the w_j's.
4. $-w_j + y_{ij} \leq 0$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
5. $w_j - y_{ij} + w_{max} x_{ij} \leq w_{max}$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
6. $\sum_{j=1}^{N_B} x_{ij} = 1$, $1 \leq i \leq N_C$
7. $w_j \leq \phi_i$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
8. $x_{ij} = 0$ or 1

Fig. 8. Integer linear programming model for $\mathcal{P}3$.

PROOF. Once again, using the same strategy as in the proof of Theorem 2, it is straightforward to show that $\mathcal{P}3$ belongs to NP. To show that $\mathcal{P}3$ is NP-hard, we polynomially transform an arbitrary instance of the known NP-complete problem $\mathcal{P}2$ to an instance of $\mathcal{P}3$. Consider an instance of $\mathcal{P}2$ parameterized by (N_C, N_B, W), with the decision problem version, checking if the testing time is less than or equal to $\mathcal{T}$. The corresponding instance of $\mathcal{P}3$ that we consider is parameterized by $(N_C, N_B, \mathcal{T})$. Suppose a solution to $\mathcal{P}3$ is obtained in polynomial time with a width of $W^\star$. We now check if $W^\star \leq W$. This provides a solution in polynomial time for $\mathcal{P}2$. Thus, we conclude that $\mathcal{P}3$ is NP-hard, and therefore NP-complete. □

As in the case of $\mathcal{P}2$, even though $\mathcal{P}3$ is NP-complete, it can be solved exactly for instances of realistic core-based systems. The ILP model for $\mathcal{P}3$ can be derived directly from $\mathcal{P}2$, and is shown in Figure 8. This model is of the same size as that for $\mathcal{P}2$, i.e., it has the same number of variables and constraints. The following theorem relates the width of the widest test bus to the minimum testing time $\mathcal{T}$ and the test widths of the cores.

THEOREM 5. *Let* $\{w_1, w_2, \ldots, w_{N_B}\}$ *be the optimal width distribution for a core-based system with* N_C *cores and maximum testing time* $\mathcal{T}$. *A lower bound on the width of the widest test bus is given by* $\max_j\{w_j\} \geq \max_i\{\phi(i) - \mathcal{T}/t_i + 1\}$.

PROOF. From the proof of Theorem 3, we know that $\mathcal{T} \geq \max_{i \in \{1, 2, \ldots, N_C\}} \{(\phi_i - \max_j\{w_j\} + 1)t_i\}$. Therefore, $\phi_i - \max_j\{w_j\} + 1 \leq \mathcal{T}/t_i$, $1 \leq i \leq N_C$. This implies that $\max_j\{w_j\} \geq \phi_i - \mathcal{T}/t_i + 1$, $1 \leq i \leq N_C$. □

As examples, consider $\mathcal{S}_1$ with two test buses as shown in Figure 2. If an upper bound $\mathcal{T} = 430000$ cycles is placed on the testing time, then Theorem 5 yields $\max_j\{w_j\} = 22$. As demonstrated in Table III, this lower bound on the test bus width is achieved using the ILP model for $\mathcal{P}3$, hence Theorem 5 provides a tight lower bound.

4. CASE STUDIES

In this section we present case studies using $\mathcal{S}_1$ and $\mathcal{S}_2$ for the optimization problems $\mathcal{P}2$ and $\mathcal{P}3$. (Solutions for the optimization problem $\mathcal{P}1$ are presented in Section 2.) We also remove some of the restrictions imposed in Sections 2 and 3 in order to simplify the ILP models. We solved the ILP models using the *lpsolve* software package on a Sun Ultra 10 workstation with a 333 MHz processor and 128 MB memory. We were unable to obtain actual CPU times from *lpsolve*; however, the user time for $\mathcal{P}1$ is less than one minute in all cases, while the user time for $\mathcal{P}2$ and $\mathcal{P}3$ is less than one hour in all cases—in fact, in most cases, the CPU time is only a few minutes. The problem instances, while realistic and representative of real-world SOCs, are small enough to be solved exactly using ILP.

Table II presents the optimal test data width, optimal width distribution, and test bus assignment vector when two test buses are considered for $\mathcal{S}_1$ and $\mathcal{S}_2$. For $\mathcal{S}_1$, the lower bound of 391,190 cycles predicted by Theorem 3 is reached for $W = 56$ bits. Any further increase in the system test width W does not decrease testing time, since the widest test bus can be at most $\min_i\{\phi_i\} = 32$ bits. Table III shows the optimal width and width distribution for $\mathcal{S}_1$ and $\mathcal{S}_2$ with two test buses for various values of the maximum testing time $\mathcal{T}$.

In Figure 9, we report experimental data for $\mathcal{P}2$ and $\mathcal{P}3$ when $\mathcal{S}_1$ contains three test buses. As expected, for a given total test width, the testing time with three buses is less than with two buses; see Figure 9(a). Not surprisingly, this difference becomes more pronounced as the total width increases. Figure 9(b) shows the total width needed for two and three buses, respectively, for a given maximum testing time $\mathcal{T}$. As $\mathcal{T}$ increases, the difference between the two two cases decreases. This is expected, since less stringent testing time requirements imply lower width requirements and decreases the need for more test buses.

Finally, we present experimental results for $\mathcal{S}2$ when a greedy heuristic test bus design is used. The heuristic divides the total test width W equally among the two test buses. In the first set of experiments, we solve $P1$ to determine the testing time and an optimal test bus assignment for this equidistribution. In the second set of experiments, we simply calculate the testing time using the assignment vector of Table II. Table IV lists the testing times for various values of W. If $\mathcal{P}1$ is applied to $\mathcal{S}_2$ for an equidistribution of W, the testing time increases by 4%. For high-volume production, this increase may translate to substantial incrases in test costs. If the test bus assignment vector of Table II is used, the increase in testing time is as high as 15%. This motivates the need for an optimal test bus design approach.

Next, we describe how the ILP models can be extended to remove the restriction $\max_j\{w_j\} \geq \min_i\{\phi_i\}$. This is necessary to decrease the testing

Table II. Optimum Testing Time and Optimal Width Distribution with Two Test Buses and System Test Width for (a) $\mathcal{S}_1$ (b) $\mathcal{S}_2$

Total test width W	Optimal width distribution (w_1, w_2)	Optimum testing time	Test bus assignment vector
8	(4,4)	497200	(2,2,2,1,2,1,2,2,2,1)
12	(6,6)	487940	(2,1,2,1,1,1,1,1,1,2)
16	(8,8)	478936	(2,2,2,2,2,2,2,2,2,1)
20	(11,9)	470380	(2,1,1,2,2,2,2,2,2,1)
24	(11,13)	461277	(2,1,1,1,1,1,1,1,1,2)
28	(16,12)	452781	(1,2,2,1,2,1,2,2,2,1)
32	(18,14)	443620	(2,1,2,2,2,2,2,2,2,1)
36	(21,15)	435042	(1,1,2,1,2,1,2,2,2,1)
40	(17,23)	426043	(2,2,2,1,1,1,2,1,1,2)
44	(25,19)	417057	(2,2,2,2,2,1,2,2,2,1)
48	(28,20)	408077	(1,1,2,1,2,1,2,2,2,1)
52	(22,30)	399290	(2,2,2,2,2,2,2,2,2,1)
56	(32,24)	391190*	(2,2,2,2,2,2,2,2,2,1)
60	(32,28)	391190*	(2,2,2,2,2,2,2,2,2,1)
64	(32,32)	391190*	(2,2,2,2,2,2,2,2,2,1)

(a)

Total test width W	Optimal width distribution (w_1, w_2)	Optimum testing time	Test bus assignment vector
16	(15,1)	2423712	(2,2,2,1,2,1,2,1,1,1)
20	(1,19)	2363126	(2,2,1,2,1,2,1,2,2,2)
24	(23,1)	2278443	(2,1,1,1,2,1,2,1,1,1)
32	(3,29)	2202286	(2,2,2,2,1,2,2,2,2,1)
36	(4,32)	2174501	(2,2,2,2,1,2,2,1,1,2)
40	(9,31)	2149720	(2,2,2,2,1,2,2,2,2,1)
44	(12,32)	2123437	(2,2,2,2,1,2,2,2,2,1)
48	(32,16)	2099390	(2,1,1,1,2,1,1,1,1,2)
52	(32,20)	2086542	(2,2,1,1,2,1,1,1,1,2)
56	(25,31)	2069738	(2,2,2,2,1,2,1,2,2,2)
60	(28,32)	2044346	(2,2,2,2,1,2,1,2,2,2)
64	(32,32)	2029753	(2,2,1,2,2,1,1,1,1,2)

(b)

*Lower bound on the system testing time (Theorem 3)

Table III. Optimal Width and Width Distribution for $\mathcal{S}_1$ with Two Test Buses and a Maximum Testing Time

Maximum testing time $\mathcal{T}$	Optimal test width W	Optimal width distribution (w_1, w_2)	Test bus assignment vector
400000	52	(21,31)	(2,2,2,2,2,2,1,1,1,2)
410000	48	(27,21)	(2,2,2,2,2,2,2,2,2,1)
420000	43	(25,18)	(2,2,2,2,1,1,2,2,2,1)
430000	39	(22,17)	(2,2,2,2,2,2,2,2,2,1)
440000	34	(19,15)	(2,2,2,2,2,2,2,2,2,1)
450000	30	(16,14)	(2,2,2,2,2,2,2,2,2,1)
460000	25	(14,11)	(2,2,2,1,2,2,2,2,2,1)
470000	21	(11,10)	(2,2,2,2,2,2,2,2,2,1)
480000	16	(8,8)	(2,2,2,2,2,2,2,2,2,1)

Table IV. Experimental Results for a Greedy Test Bus Width Distribution

Total test width W	Test bus assignment vector	Testing time	Percentage increase over optimum	Testing time for assignment of Table 2	Percentage increase over optimum
24	(2,2,2,2,2,1,1,1,1,2)	2383808	4.6	2573397	12.9
32	(2,2,2,1,1,1,2,1,1,2,2)	2306158	4.7	2523866	14.6
36	(2,1,2,2,2,1,1,1,1,2)	2274197	4.6	2511742	15.5
40	(2,1,2,2,2,1,1,1,1,2)	2237623	4.1	2417750	12.5

time below the limit of Theorem 3 if greater test width is available. Let δ_{ij} be an "indicator" 0-1 variable defined as follows:

$$\delta_{ij} = \begin{cases} 1 & \text{if } w_j > \phi_i \\ 0 & \text{otherwise} \end{cases}$$

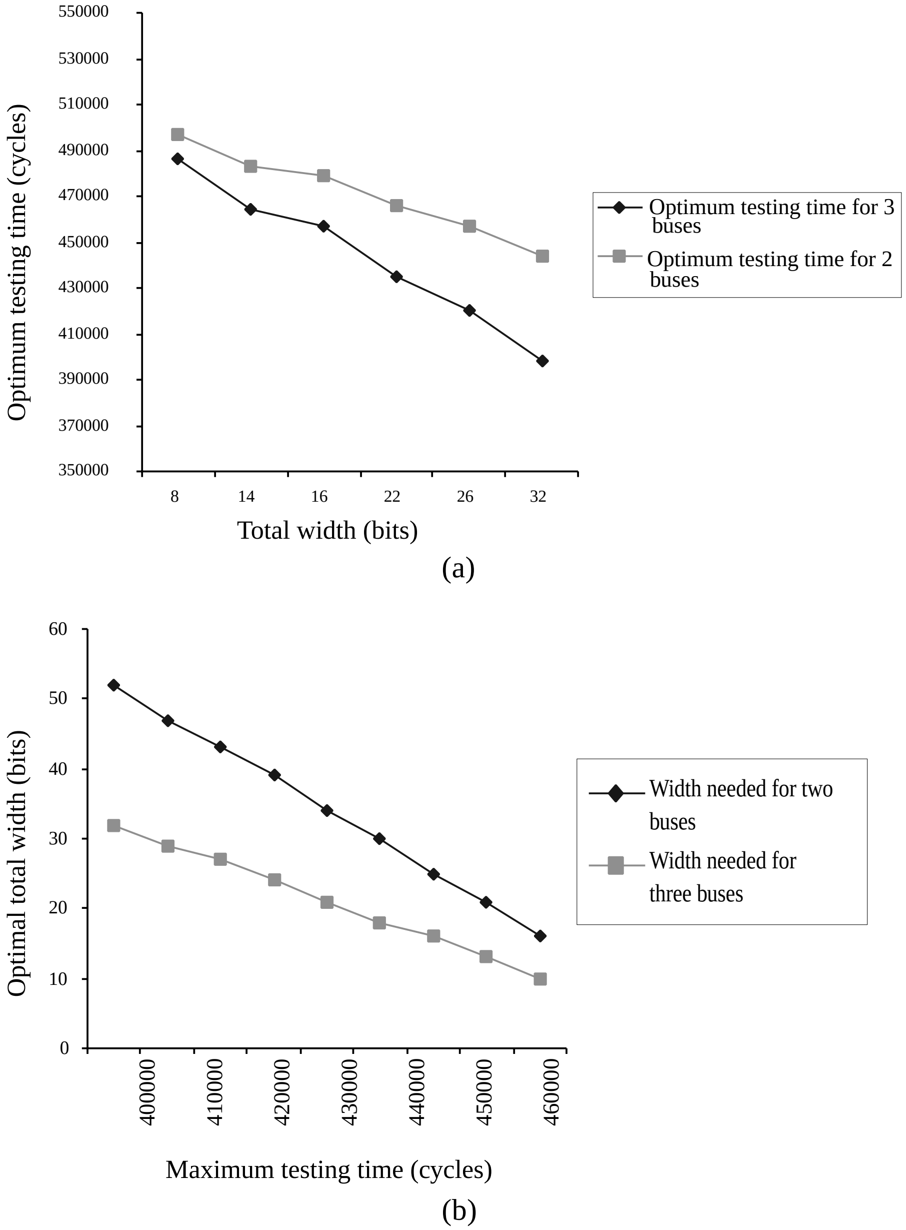
The testing time T_{ij} for core i assigned to test bus j can now be expressed as

$$T_{ij} = \delta_{ij} x_{ij} t_i + (1 - \delta_{ij})(\phi_i - w_j + 1) t_i x_{ij}$$

with the constraint that $\delta_{ij}(w_j - \phi_i) + (1 - \delta_{ij})(\phi_i - w_j) \geq 0$. The nonlinear terms in this formulation can be linearized as in Section 3, and the resulting ILP model can be solved easily to obtain optimal width distribution and test bus assignment. We solved the ILP model for $\mathcal{S}_2$, and the results in Table V indicate that significant reductions in testing time are achieved, especially for higher test widths.

The ILP formulation also allows us to sometimes decrease the test width of the cores in the system, i.e., the number of lines that connect the cores to their respective test buses, without increasing the overall testing time. For example, if core i is connected to test bus j of width w_j, then even though w_j lines are available for propagating test data to and from core i, it is not always necessary to use all these lines. This is especially the case if bus j is not the test bottleneck. In such situations, $b_i < w_j$ lines connect core i to test bus j. We refer to b_i as the test width of core i. The motivation for using smaller core test width lies in the reduction of routing and interconnect area for SOC testing.

We allow b_i to be less than w_j in our ILP model by introducing the constraint $x_{ij} b_i \leq w_j$, which implies that the test width for core i may be less than the width of test bus j if core i is assigned to bus j. We also replace w_j by b_i in the expression for T_{ij}. The product term $x_{ij} b_i$ does not pose a problem, since it is linearized for both $\mathcal{P}1$ and $\mathcal{P}2$. Table VI shows how the test width cores in $\mathcal{S}_2$ can be reduced without increasing the system testing time.



(a)

(b)

Fig. 9. Comparison of the optimal testing time and optimal width for $\mathcal{S}_1$ with two and three test buses.

5. OPTIMAL SUBDIVISION OF TEST BUSES

In this section we allow the width of the test buses to be distributed among several buses with smaller widths. This allows the w_j bits of test bus j to be divided into several parts, each of which can test one or more cores in parallel (see Figure 6, for example). For a given total test width, the

Table V. Optimum Testing Time and Optimal Width Distribution with Two Test Buses and a System Test Width for $\mathcal{S}_2$ (with no explicit bound on width of individual test buses)

Total test width W	Optimal width distribution (w_1, w_2)	Optimum testing time	Test bus assignment vector
48	(40,8)	2020973	(2,2,2,1,2,1,1,1,2,1)
56	(42,14)	1967215	(2,2,2,1,2,1,1,1,2,1)
60	(43,17)	1940336	(2,2,2,1,2,1,1,1,2,1)
64	(53,11)	1937253	(2,2,2,2,2,1,1,1,1,1)
80	(48,32)	1867442	(2,1,2,2,2,1,1,1,2,1)

Table VI. Reducing Test Data Width for Each Core in $\mathcal{S}_2$

Total test width W	w_1	w_2	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
20	1	19	**4**	**3**	1	**17**	1	**19**	1	**18**	**17**	**18**
32	3	29	**1**	**12**	29	**26**	3	29	29	**28**	29	3
36	4	32	**1**	**16**	32	32	4	32	**31**	4	4	**30**
40	9	31	**7**	**6**	31	**28**	9	31	31	**30**	31	9

subdivision of test buses allows further reductions in the testing time. We first make the simplifying assumption that the width w_j for each test bus j is known. Later we extend our model to the case where the widths are not known and optimal widths have to be determined. The optimization problem being considered here is stated formally below.

- $\mathcal{P}4$: Given N_C cores, N_B test buses with known widths $w_1, w_2, \ldots, w_{N_B}$, respectively, and an upper limit j_{max} on the number of subdivisions allowed for test bus j, $1 \leq j \leq N_B$, determine (i) an optimal subdivision of test bus widths, and (ii) an optimal assignment of cores to test buses such that the total testing time is minimized.

Note that the decision version of $\mathcal{P}4$ can also be shown to be NP-complete using the method of restriction. A restriction of $\mathcal{P}4$ to $\mathcal{P}1$ is achieved by setting $j_{max} = 1$, $1 \leq j \leq N_B$.

Let x_{ij} be a 0-1 variable as defined in Section 2. Test bus j can be divided into a maximum of j_{max} parts, each part serving as a test bus for a subset of cores in the system. Suppose that these parts have widths w_{j1}, $w_{j2}, \ldots w_{jj_{max}}$, respectively, such that $\sum_{k=1}^{j_{max}} w_{jk} = w_j$, $1 \leq j \leq N_B$.

Let y_{ijk} be a 0-1 variable defined as follows:

$$y_{ijk} = \begin{cases} 1 & \text{if core } i \text{ is assigned to the } k\text{th part of bus } j \\ 0 & \text{otherwise} \end{cases}$$

The following constraint follows directly from the definitions of the 0-1 variables. It denotes the fact that a core is either assigned to a test bus with its complete width or to a portion of a test bus (with reduced width).

$$\sum_{j=1}^{N_B}\sum_{k=1}^{j_{max}} y_{ijk} + \sum_{j=1}^{N_B} x_{ij} = 1,\ 1 \le i \le N_C$$

If core i is assigned the entire width of bus j, then its testing time is $(\phi_i - w_j + 1)t_i$. (We assume as before that $\max_j\{w_j\} \le \min_i\{\phi_i\}$.) On the other hand, if it is assigned to the kth test bus derived from bus j, then its testing time is $(\phi_i - w_{jk} + 1)t_i$.

The cost function (system testing time) C can now be expressed in terms of the above parameters.

$$C = \max_{j\in\{1,2,\ldots,N_B\}}\Big\{\max_{k\in\{1,2,\ldots,j_{max}\}} \textstyle\sum_{i=1}^{N_C}(\phi_i - w_{jk} + 1)t_i y_{ijk} + \sum_{i=1}^{N_C}(\phi_i - w_j + 1)t_i x_{ij}\Big\}.$$

The right-hand side of the above equation is nonlinear, but it can be linearized as before by a sequence of transformations. Let

$$C_{1j} = \max_{k\in\{1,2,\ldots,j_{max}\}} \textstyle\sum_{i=1}^{N_C}(\phi_i - w_{jk} + 1)t_i y_{ijk}$$

and $C_{2j} = \sum_{i=1}^{N_C}(\phi_i - w_j + 1)t_i x_{ij}$. The cost function can then be expressed as $C = \max_{j\in\{1,2,\ldots N_B\}}(C_{1j} + C_{2j})$ and the optimization problem can be formulated as minimize C, subject to

(1) $C \ge (C_{1j} + C_{2j})$, $1 \le j \le N_B$

(2) $\sum_{k=1}^{j_{max}} w_{jk} = w_j$, $1 \le j \le N_B$

(3) $w_j \le \phi_i$, $1 \le i \le N_C$, $1 \le j \le N_B$

(4) $x_{ij} = 0$ or 1, $1 \le i \le N_C$, $1 \le j \le N_B$

(5) $y_{ijk} = 0$ or 1, $1 \le i \le N_C$, $1 \le j \le N_B$, $1 \le k \le j_{max}$

We next linearize constraint (1). Note that $C_{1j} \ge \sum_{i=1}^{N_C}(\phi_i - w_{jk} + 1)t_i y_{ijk}$ can be linearized by adding a nonbinary, integer variable r_{ijk} for each i, j, k, and adding three constraints as in the case of $\mathcal{P}2$ in Section 3. This yields the ILP model for $\mathcal{P}4$ shown in Figure 10. It contains at most $N_B N_C + N_B N_C l$ binary variables, $N_B N_C l + N_B l + 2N_B + 1$ nonbinary, integer variables, and $5N_B N_C + 3N_B N_C + 4N_B + N_B l$ constraint inequalities, where $l = \max_j\{j_{max}\}$.

We next generalize $\mathcal{P}4$ to the case where the widths of the test buses also need to be optimally determined. The formal statement of this problem is given below:

Minimize C subject to:

1. $C \geq (C_{1j} + C_{2j})$, $1 \leq j \leq N_B$
2. $C_{1j} \geq \sum_{i=1}^{N_C}((\phi_i + 1)t_i y_{ijk} - t_i r_{ijk})$, $1 \leq j \leq N_B$, $1 \leq k \leq j_{max}$
3. $r_{ijk} - W y_{ijk}$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$, $1 \leq k \leq j_{max}$, W is an upper bound on w_{jk}
4. $-w_{jk} + r_{ijk} \geq 0$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$, $1 \leq k \leq j_{max}$
5. $w_{jk} - r_{ijk} + W y_{ijk} \leq W$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$, $1 \leq k \leq j_{max}$
6. $C_{2j} \geq \sum_{i=1}^{N_C}(\phi_i - w_j + 1)t_i x_{ij}$, $1 \leq j \leq N_B$
7. $\sum_{j=1}^{N_B}(x_{ij} + \sum_{k=1}^{j_{max}} y_{ijk}) = 1$, $1 \leq i \leq N_C$
8. $\sum_{k=1}^{j_{max}} w_{jk} = w_j$, $1 \leq j \leq N_B$.
9. $w_j \leq \phi_i$, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
10. $x_{ij} = 0$ or 1, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$
11. $y_{ijk} = 0$ or 1, $1 \leq i \leq N_C$, $1 \leq j \leq N_B$, $1 \leq k \leq j_{max}$

Fig. 10. Integer linear programming model for $\mathcal{P}4$.

- $\mathcal{P}5$: Given N_C cores, N_B test buses with total width W, and an upper limit j_{max} on the number of subdivisions allowed for test bus j, $1 \leq j \leq N_B$, determine (i) an optimal width for each test bus, the optimal subdivision of the width of every test bus, and (ii) an assignment of cores to test buses such that the total testing time is minimized.

The problem decision version of $\mathcal{P}5$ can also be shown to be NP-complete by restricting it to $\mathcal{P}2$. This is achieved by imposing the restriction $j_{max} = 1$, $1 \leq j \leq N_B$. The ILP model for $\mathcal{P}5$, shown in Figure 11, is obtained by combining the ILP models for $\mathcal{P}2$ and $\mathcal{P}4$. Integer variables s_{ij} are introduced for linearization. It contains at most $N_B N_C + N_B N_C l$ binary variables, $N_B N_C l + N_B l + N_B N_C + 2N_B + 1$ nonbinary integer variables, and $5N_B N_C + 6N_B N_C + 4N_B + N_B l$ constraint inequalities, where $l = \max_j\{j_{max}\}$ as before.

Finally, we present experimental results in solving optimization problems, $\mathcal{P}4$ and $\mathcal{P}5$. We considered $\mathcal{S}_1$ and $\mathcal{S}_2$ with two test buses (1 and 2), and we modeled the situation where the first test bus can fork into at most two branches (1a and 1b). The objective of this set of experiments was twofold: (i) demonstrate that $\mathcal{P}4$ ($\mathcal{P}5$) provides lower testing time than $\mathcal{P}1$ ($\mathcal{P}2$), and (ii) show that even nonoptimal solutions for $\mathcal{P}5$ provide lower testing time than $\mathcal{P}2$.

Minimize C subject to:

1. $C \geq (C_{1j} + C_{2j}),\ 1 \leq j \leq N_B$
2. $C_{1j} \geq \sum_{i=1}^{N_C}((\phi_i + 1)t_i y_{ijk} - t_i r_{ijk}),\ 1 \leq j \leq N_B,\ 1 \leq k \leq j_{max}$
3. $r_{ijk} - W y_{ijk},\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B,\ 1 \leq k \leq j_{max}$, W is an upper bound on w_{jk}
4. $-w_{jk} + r_{ijk} \geq 0,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B,\ 1 \leq k \leq j_{max}$
5. $w_{jk} - r_{ijk} + W y_{ijk} \leq W,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B,\ 1 \leq k \leq j_{max}$
6. $C_{2j} \geq \sum_{i=1}^{N_C}((\phi_i + 1)t_i x_{ij} - t_i s_{ij}),\ 1 \leq j \leq N_B$
7. $s_{ij} - w_{max} x_{ij} \leq 0,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B$, where w_{max} is an upper bound on the w_j's.
8. $-w_j + s_{ij} \leq 0,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B$
9. $w_j - s_{ij} + w_{max} x_{ij} \leq w_{max},\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B$
10. $\sum_{j=1}^{N_B}(x_{ij} + \sum_{k=1}^{j_{max}} y_{ijk}) = 1,\ 1 \leq i \leq N_C$
11. $\sum_{k=1}^{j_{max}} w_{jk} = w_j,\ 1 \leq j \leq N_B.$
12. $w_j \leq \phi_i,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B$
13. $x_{ij} = 0$ or $1,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B$
14. $y_{ijk} = 0$ or $1,\ 1 \leq i \leq N_C,\ 1 \leq j \leq N_B,\ 1 \leq k \leq j_{max}$

Fig. 11. Integer linear programming model for $\mathcal{P}5$.

We first return to the example based on $\mathcal{S}_1$, which we presented in Section 2 to illustrate $\mathcal{P}1$. For this example, $w_1 = 32$, $w_2 = 16$, and an optimal testing time of 411,884 cycles was obtained using $\mathcal{P}1$. By allowing the 32-bit bus to fork into two branches of 27 bits (1a) and 5 bits (1b) each, we achieve a reduced, but nonoptimal, testing time of 409,472 cycles using the test bus assignment (2,2,2,2,2,2,2,2,2,1b,1a).

Unfortunately, *lpsolve* did not run to completion for all cases when we attempted to solve $\mathcal{P}4$ and $\mathcal{P}5$. Nevertheless, we allowed it to run for up to two hours, after which we tabulated the best solution obtained. These results (optimal and nonoptimal) for $\mathcal{P}5$ are presented in Table VII. The experimental results show that the added flexibility of allowing test buses to be subdivided can reduce the testing time significantly, especially for an SOC such as $\mathcal{S}_2$. Note also that subdivision also provides the same minimum testing time with 36-bit width as with a 44-bit width for $\mathcal{S}_1$. Figure 12 illustrates an optimal test access architecture based on $\mathcal{P}5$ for $\mathcal{S}_1$, when the total width of 36 bits and at most one subdivision of the first test bus is allowed.

6. CONCLUSIONS

We have presented a formal methodology for designing optimal test access architectures for testing SOC designs. In doing so, we have attempted to provide a formal basis for comparing the several ad hoc test access

Table VII. Optimum Testing Time and Optimal Width Distribution with a Total Test Data Width and Two Test Buses (one forks into two branches): (a) $\mathcal{S}_1$ (b) $\mathcal{S}_2$

Total test width W	Distribution of W: (w_1, w_2)	Distribution of w_1	Testing time	Test bus assignment vector	Improvement over $\mathcal{P}2$ (percent)
20	(19,1)	(1,18)	441404*	(1b,1a,1a,1a,1a,1a,2,2,1a,1b)	6.16
24	(23,1)	(1,22)	426564*	(1b,1a,1a,1a,1a,1a,2,2,1b,1b)	7.53
28	(27,1)	(26,1)	412427*	(1a,1b,1b,1b,1b,1b,2,2,1b,1a)	8.91
32	(19,13)	(1,18)	441404	(1b,1a,1a,1a,2,1a,2,2,2,1b)	0.50
36	(32,4)	(31,1)	394012*	(1a,1b,1b,1b,1b,1b,2,2,1a,1a)	9.43
44	(32,12)	(31,1)	394012*	(1a,1b,1b,1b,1b,1b,2,2,1b,1a)	5.53
52	(22,20)	(1,21)	397748	(1b,1a,1a,1a,2,1a,1,1,1a,2)	0.39

(a)

Total test width W	Distribution of W: (w_1, w_2)	Distribution of w_1	Testing time	Test bus assignment vector	Improvement over $\mathcal{P}2$ (percent)
24	(23,1)	(12,11)	1677735*	(1a,1a,1a,1a,1b,1a,2,2,2,2)	23.36
36	(28,8)	(16,12)	1672265	(1a,1,1a,1a,1b,1a,2,2,2,2)	23.10
40	(30,10)	(18,12)	1672119	(1a,1,1a,1a,1b,1a,2,2,2,2)	22.22
44	(32,12)	(17,15)	1633600	(1a,1,1a,1a,1b,1a,2,2,2,2)	23.19

(b)

* Optimum testing time (*lpsolve* ran to completion)

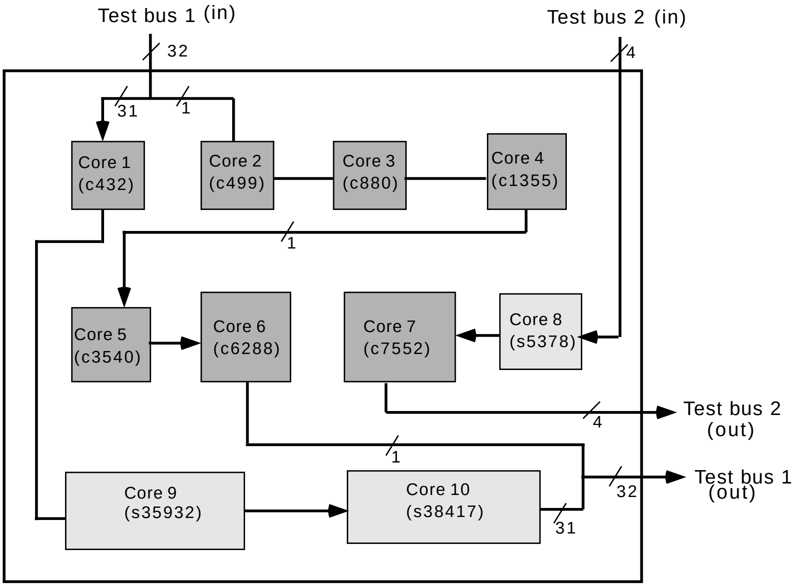


Fig. 12. An optimal test bus architecture for $\mathcal{S}_1$ with two test buses, total width of 36 bits, and only one subdivision allowed for the first test bus.

architectures proposed in the literature. The methodology allows designers to explore design options and make appropriate choices. We examined several problems related to the design of optimal test architectures, including assigning cores to test buses, distributing a given test data width among multiple test buses, and determining the amount of test data width required to satisfy an upper bound on testing time. We have shown that even though the decision versions of these design problems are NP-complete, they can be modeled efficiently using integer linear programming for practical instances. We applied these models to two nontrivial core-based systems and solved them using a standard software package available in the public domain. We are currently extending the ILP models to incorporate routing and additional power constraints, and recently reported initial results in this direction [Chakrabarty 2000b].

Our results give rise to a number of useful extensions and new directions for further research, summarized below.

- The ILP models need to be generalized to handle test access architectures of the type shown in Figure 1, where a test bus may fork, but not necessarily merge.
- Test access architectures may also be designed hierachically. Hence, ILP models should be able to handle hierarchical compositions, where *complex* cores embed one or more *simple* cores. Moreover, $\mathcal{P}4$ and $\mathcal{P}5$ should be extended to handle recursive sudivision of the test buses.
- The ILP model descriptions that we have used in our experiments are problem-specific, i.e., they are described in a format specific to the problem instance and to the *lpsolve* program. This is a cumbersome process. It is far more convenient to use high-level languages such as AMPL [Fourer 1993] and GAMS [GAMS Development Corporation 1993] that allow the model to be described in a *parameterized form*, independent of the ILP solver and the input data used for a specific instance of the model.
- Finally, significant advances have been made in recent years in solving nonlinear integer programs, and a number of these solvers are now readily available, e.g., through the Argonne National Laboratory <http://www.mcs.anl.gov/otc/Server/neos.html>. We are examining the feasibility of using such nonlinear solvers for designing optimal test access architectures.

ACKNOWLEDGMENTS

The author thanks Erik Jan Marinissen of Philips Research Laboratories for valuable comments on an earlier version of this manuscript.

REFERENCES

AERTS, J. AND MARINISSEN, E. J. 1998. Scan chain design for test time reduction in core-based ICs. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 448–457.

BERKELAAR, M. 1999. lpsolve, version 2.0. Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, Netherlands. E-mail: michel@es.ele.tue.nl

BRGLEZ, F., BRYANT, D., AND KOZMINSKI, K. 1989. Combinational profiles of sequential benchmark circuits. In *Proceedings of the IEEE International Symposium on Circuits and Systems* (Portland, OR, May). IEEE Press, Piscataway, NJ, 1929–1934.

BRGLEZ, F. AND FUJIWARA, H. 1985. A neural netlist of 10 combinational benchmark circuits and a target translator in Fortran. In *Proceedings of IEEE International Symposium on Circuits and Systems*. IEEE Computer Society, Washington, DC, 695–698.

CHAKRABARTY, K. 2000a. Test scheduling for core-based systems using mixed-integer linear programming. *IEEE Trans. Comput.-Aided Des. 19*, 10 (Oct.), 1163–1174.

CHAKRABARTY, K. 2000b. Design of system-on-a-chip test access architectures under place-and-route and power constraints. In *Proceedings of the Conference on Design Automation* (DAC '2000). ACM Press, New York, NY, 432–437.

CHAKRABARTY, K. 1999. Test scheduling for core-based systems. In *Proceedings of International Conference on Computer-Aided Design* (Nov.). IEEE Computer Society Press, Los Alamitos, CA, 391–394.

FOURER, R., GAY, D. M., AND KERNIGHAN, B. W. 1993. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, Boston, MA.

GAMS DEVELOPMENT CORPORATION. 1993. *GAMS: A User's Guide*. Boyd and Fraser Publishing Co., Danvers, MA.

GAREY, M. AND JOHNSON, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., New York, NY.

GHOSH, I., DEY, S., AND JHA, N. K. 1998. A fast and low cost testing technique for core-based system-on-chip. In *Proceedings of the 35th Annual Conference on Design Automation* (DAC '98, San Francisco, CA, June 15–19), B. R. Chawla, R. E. Bryant, and J. M. Rabaey, Chairs. ACM Press, New York, NY, 542–547.

GHOSH, I., JHA, N. K., AND DEY, S. 1997. A low overhead design for testability and test generation technique for core-based systems. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 50–59.

HAMZAOGLU, I. AND PATEL, J. H. 1998. Test set compaction algorithms for combinational circuits. In *Proceedings of the 1998 IEEE/ACM International Conference on Computer-Aided Design* (ICCAD '98, San Jose, CA, Nov. 8–12), H. Yasuura, Chair. ACM Press, New York, NY, 283–289.

IMMANENI, V. AND RAMAN, S. 1990. Direct access test scheme: Design of block and core cells for embedded ASICs. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 488–492.

MARINISSEN, E. J. AND LOUSBERG, M. 1999. The role of test protocols in testing embedded-core-based system ICs. In *Proceedings of the IEEE European Test Workshop*. IEEE Computer Society Press, Los Alamitos, CA, 70–75.

MARINISSEN, E. J., ZORIAN, Y., KAPUR, R., TAYLOR, T., AND WHETSEL, L. 1999. Towards a standard for embedded core test: An example. In *Proceedings of the IEEE International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 616–627.

MARINISSEN, E. J., ARENDSEN, R., BOS, G., DINGEMANSE, H., LOUSBERG, M., AND WOUTERS, C. 1998. A structured and scalable mechanism for test access to embedded reusable cores. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 284–293.

SUGIHARA, M., DATE, H., AND YASUURA, H. 1998. A novel test methodology for core-based system LSIs and a testing time minimization problem. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 465–472.

WILLIAMS, H. P. 1985. *Model Building in Mathematical Programming*. 2nd ed. John Wiley, New York, NY.

VARMA, P. AND BHATIA, S. 1998. A structured test re-use methodology for core-based system chips. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 294–302.

ZORIAN, Y., MARINISSEN, E. J., AND DEY, S. 1998. Testing embedded-core based system chips. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 130–143.

ZORIAN, Y. 1997. Test requirements for embedded core-based systems and IEEE P1500. In *Proceedings of the International Test Conference*. IEEE Computer Society Press, Los Alamitos, CA, 191–199.

Received: May 1999; accepted: July 2000