J Electron Test (2017) 33:7–23
DOI 10.1007/s10836-016-5638-5

Test Planning for Core-based Integrated Circuits Under Power Constraints

Breeta SenGupta[1] · Dimitar Nikolov[1] · Urban Ingelsson[2] · Erik Larsson[1]

Received: 6 July 2016 / Accepted: 28 December 2016 / Published online: 30 January 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract This paper addresses reduction of test cost for core-based non-stacked integrated circuits (ICs) and stacked integrated circuits (SICs) by test planning, under power constraint. Test planning involves co-optimization of cost associated with test time and test hardware. Test architecture is considered compliant with IEEE 1149.1 standard. A cost model is presented for calculating the cost of any test plan for a given non-stacked IC and a SIC. An algorithm is proposed for minimizing the cost. Experiments are performed with several ITC'02 benchmark circuits to compare the efficiency of the proposed power constrained test planning algorithm against near optimal results obtained with Simulated Annealing. Results validate test cost obtained by the proposed algorithm are very close to those obtained with Simulated Annealing, at significantly lower computation time.

Keywords Test plan · Test schedule · Test cost · Power constraint · Stacked integrated circuit · JTAG · IEEE 1149.1 · Boundary scan

Responsible Editor: Y. Zorian

✉ Breeta SenGupta
breeta.sengupta@eit.lth.se; breetasg@gmail.com

Dimitar Nikolov
dimitar.nikolov@eit.lth.se

Urban Ingelsson
urban.ingelsson@semcon.com

Erik Larsson
erik.larsson@eit.lth.se

[1] Lund University, Lund, Sweden

[2] Semcon, Linköping, Sweden

1 Introduction

Advances in IC fabrication technology has reduced the cost of production of ICs [25]. With increasing complexity of the fabrication process, ICs are increasingly prone to manufacturing defects. Thus, each IC must be carefully tested. The relatively rapid decline in the production cost as compared to the test cost has resulted in the cost of testing to hold a considerable part of the cost of production of present-day ICs. Majority of IC manufacturers expect the cost related to test to be the bottleneck of the production cost of SICs in the forthcoming years [25]. The introduction of SICs, where multiple chips (dies) are stacked and bonded in a package, have attracted the attention of manufacturers with benefits such as enhanced performance, reduced power consumption and smaller form factor [14–16, 18, 30]. SICs include a broad range of package technologies, *viz.*, System-in-Package (SiP), Package-on-Package (PoP) and 3D Stacked ICs with Through-Silicon-Vias (3D SIC-TSV). However, SICs need new means of addressing issues related to test cost, process complexity and potential for damage. Hence, it is essential to reduce test cost to achieve a low manufacturing cost. Two important contributors to the test cost of ICs are the test time and the Design-for-Test (DfT) hardware.

The test time of ICs correspond to the time taken to execute the applied test schedule. An efficient test schedule determines the order in which the various logic blocks, *i.e.*, cores of an IC are tested, such that the time taken to test all cores is minimized. A common approach for reducing the test time for core-based ICs would be to perform concurrent core tests when possible. However, concurrent testing leads to higher power consumption compared to sequential testing. The power consumption during testing must be regulated [5], to avoid false test positives due to voltage drop or damage due to overheating. Considerable

research has addressed reduction of test time by scheduling tests for non-stacked ICs [4–6, 9, 10, 19, 29, 32]. While [5, 19, 32] address scheduling tests in sessions for non-stacked ICs under resource and power constraints, in our previous work [26], we have addressed test scheduling for SICs under power constraints. We have proposed an algorithm for session-based test scheduling of core-based SICs with TSVs. In addition to the test schedule, the test time of an IC also depends on the number of times the IC or a component/part of the IC is tested. An IC may be tested at multiple stages during the manufacturing process. These stages are known as test instances. The test instances during which testing is performed comprise the test flow. In case of non-stacked ICs, the test flow typically includes: test of the bare die during wafer sort, and eventually after packaging at package test. Unlike non-stacked ICs, SICs have additional test instances after each die is stacked [18]. The test instances for SICs can be broadly classified as:

1) Wafer sort: Testing each chip prior to integration into the stack to sort out known good dies (KGDs).
2) Intermediate test: Testing the partially constructed chip stack.
3) Package test: Testing the packaged assembly.

For non-stacked ICs, the same test schedule is applied to the bare chip at wafer sort and to the complete packaged chip during package test. However, in case of SICs, the cores tested at each test instance are different, and hence the individual test schedules. Therefore, it is important to consider all test instances of the test flow simultaneously to determine a test schedule for the entire SIC with reduced test time. In this paper we consider this for a generic test flow for all SICs, that consists of the wafer sort tests of individual chips, directly followed by the package test.

Another factor contributing largely to the overall test cost is the cost associated with DfT hardware. Thus, to reduce the total test cost, it is also important to optimize the test architecture. This paper adopts the IEEE 1149.1 test architecture standard, commonly known as JTAG or boundary scan. Various authors have discussed testing, test architecture design and optimization for non-stacked ICs with IEEE 1149.1 [1–3, 8–10, 21, 22, 27, 28, 31]. For SICs with TSVs, optimization of DfT architecture has also been addressed [17, 20]. However, reduction of overall test cost for core based ICs supported by the IEEE 1149.1 test infrastructure, considering both test time and DfT hardware while meeting a power constraint, which remains unexplored, is addressed in this paper.

Recent research addresses various aspects of testing SICs, including specific defects for SICs with TSVs [14–16, 18, 30]; yet the issue of test planning, considering the components of test-cost, remains unexplored. Therefore, it is crucial that a comprehensive test plan to minimize the test cost considers both test time and DfT hardware.

To arrive at a cost efficient test plan, we define test cost as a function of both test time and the DfT hardware. While the test time is a sum of wafer sort and package test times, the DfT hardware is represented by the number of Test Data Registers (TDRs). An efficient way of keeping the DfT hardware cost at a minimum is by enabling the DfT hardware at wafer sort to be re-used during package test. Therefore we consider the test architecture as proposed in [17]; where each chip in the stack is provided with IEEE 1149.1, which supports wafer sort, while the IEEE 1149.1 input and output of the chips in the stack are interconnected to provide access during package test. Scheduling tests to minimize the test time increases the number of TDRs required, while minimizing the number of TDRs reduces the flexibility of scheduling core tests, increasing the overall test time. Hence, optimizing only one aspect of test cost may provide sub-optimal results overall. However, scheduling tests under power constraints, with the objective of minimizing test cost, is NP-hard. Therefore, we propose a test planning algorithm designed for core-based non-stacked ICs and SICs to minimize the test cost by co-optimizing the test time and DfT hardware. We also make use of Simulated Annealing [13], where we perform nearly exhaustive search, thus obtaining near-optimal test cost, at the expense of much longer computational time. Comparison of the results produced by Simulated Annealing and our proposed heuristic show that our heuristic produces results which are close to the results produced by Simulated Annealing, but at a significantly lower computational cost.

This paper is focussed towards test planning considering resource and power constraints. We limit the work assuming a test architecture based on the IEEE 1149.1 standard, which despite its convenience and extensive use for testing, limits the width of Test Access Mechanism (TAM). Second, a major challenge in testing SICs involves the test of TSVs. In this paper we treat the TSVs in a similar manner to a core of the SIC, but cannot be tested in the same session with a core. Third, a common power constraint is assumed in the experiments for the entire SIC, during all test instances, which may be limited in terms of practicality. Fourth, we limited the work to test planning, which means we do not address a number of practical and still open problems such as scan enable control and interconnect test between chips with glue logic. Lastly, a test flow comprising of wafer sort followed by package test is assumed in this paper, both for non-stacked ICs and SICs.

This paper proceeds with related work on test scheduling, test access architecture and test cost analysis for non-stacked ICs and SICs in Section 2, followed by a background on the manufacturing and testing process of SICs in

Section 3. The test architecture considered in this paper is discussed in Section 4. The test planning problem is defined in Section 5 and illustrated with an example in Section 6 leading to the test planning approach in Section 7. Experiments designed to validate the proposal are described in Section 8 and finally, conclusions are drawn in Section 9.

2 Related Work

A brief discussion of test architecture of non-stacked ICs and SICs provided with the IEEE 1149.1 standard, followed by test scheduling of non-stacked ICs, and test planning for SICs with TSVs is presented in this section.

The IEEE 1149.1 standard, also known as JTAG or boundary scan, is used for testing digital chips and interconnects between chips [2, 8, 22, 27]. Test architecture based on IEEE 1149.1 has been proposed for various embedded core-based systems [1, 21, 28]. For non-stacked ICs, extensive work is available which describes and optimizes test architecture using IEEE 1149.1 [3, 31]. Based on the IEEE 1149.1 test architecture, test scheduling for embedded core-based non-stacked ICs has been addressed in [6, 29]. For SICs with TSVs, Marinissen et. al. [17], proposed a scalable test architecture based on IEEE1149.1 and IEEE1500 connected on wide TAMs, with the following key features:

- Each chip is equipped with dedicated probe pads for wafer sort.
- All signals from a chip are transferred to the chip on top of it via TSVs, a.k.a. test elevators, which are situated on the top side of the chip.
- Hierarchical Wrapper Instruction Register (WIR) chains are used to prevent unbridled growth of length.

Marinissen et. al. [17] highlight the importance of standardization and optimization of test architecture of SICs, which is also addressed in this paper.

Test scheduling for non-stacked ICs to minimize the test time has been addressed in several publications [5, 7, 19, 23, 24, 32]. While Samii et. al. in [24], demonstrated reduction in test time by scheduling tests under power constraints, taking as input the power at each clock cycle; Rosinger et. al. in [23] addressed test scheduling for core-based non-stacked ICs ensuring thermal safety, as a global power constraint may not take into account power density distribution, or core layout, thus limiting lateral heat removal. However, such test scheduling approaches require precise input data such as heat dissipation map and core layout, which are difficult to obtain. Huang et. al. [7] addressed optimization of test time while considering resource constraints for core-based System-on-Chip (SoC) designs. The proposed method is achieved in the following three steps:

1) Rectangular transformation is applied to cores having more I/O pins than the number of I/O pins of the SoC.
2) A modified best fit algorithm is applied, where if a rectangle cannot be packed on an existing level, a transformation will be applied to see if it can be packed on some existing levels before a new level is created.
3) The minimum resource requirement given an upper bound on the test time can be solved using 2D bin-packing algorithm, by optimizing the width of the SoC for a given test time.

However, the issue of power constraint has not been considered in [7].

In [32], Zorian discussed power constrained test scheduling for built-in self-tested (BISTed) cores for a non-stacked IC using test sessions. In [5], Chou et al. proposed a solution for the same problem while considering resource constraints, by the formation of sessions. A session is defined as a group of tests that start simultaneously and no other tests are initiated until all tests of the session are finished. The concept of sessions simplifies test scheduling. Muresan et al. [19] developed an algorithm to schedule tests in sessions while reducing the test time for non-stacked chips under power constraints. The algorithm is described as follows:

- All core tests are sorted in descending order of their test times. No core individually violates the power or resource constraints.
- Each core test is considered in descending order of their length until all core tests in the list are assigned to a session.
- The longest core test is considered first, which constitutes the first session.
- While descending through the list of sorted core tests, each test is checked for power and resource compliance with the previously formed sessions.
- Each core test is included in the first (longest) session with which it complies in terms of power and resource constraints. This core test forms a new session, if its inclusion in the prior sessions makes it non-compliant with the constraints.

Cost related to DfT hardware is, however, not considered in [19].

The test scheduling approaches discussed in [5, 7, 19, 32] perform well for non-stacked ICs, as the same set of cores are tested during wafer sort and package test, and identical test schedules may be applied. In case of SICs, the package test involves simultaneous testing of all chips in the stack. Therefore, optimized wafer sort schedules for individual chips using [5, 19] may prove sub-optimal during package test [26]. Thus, in our previous work [26], we have addressed test scheduling under power constraints for SICs

with TSVs, considering simultaneous wafer sort and package test. We have considered core-based systems, where each core is provided with a BIST engine. Although no cost model has been developed in [26] the trade-off between test time and the number of TDRs has been accounted. Jiang et. al. in [11, 12] have proposed test architecture optimization for core based SICs with TSVs. They propose a reduction in test cost while considering both test time and DfT hardware as weighed factors of the test cost. In case of SICs with TSVs, all test data commute via the lowermost chip [14, 17, 18] unlike illustrated in [11, 12], where the TAMs start and end on any chip. Furthermore, during wafer sort and package tests, in several instances, separate TAMs are used. This would result in increase in test cost due to the nonavailability of re-usable test architecture, and the proposed approach is not scalable in case intermediate tests are necessary. Therefore, in this paper we address test planning for SICs, co-optimizing test time and DfT hardware, where each die is provided with a JTAG Test Access Port (TAP) for test access during wafer sort and package test. A scalable test architecture based on IEEE 1149.1 is considered, as proposed in [17], to develop a test plan for co-optimization of test time and DfT hardware to minimize the overall test cost, while meeting a preset power constraint.

3 Background

Some background studies pertaining to the test architecture and scheduling of SICs with TSVs are presented in this section. Stacking with TSV technologies on SIC manufacturing offers the promise of integration across multiple chips at very fine levels of granularity and concomitant savings in wiring, delay, power and form factor [25]. Earlier versions of high integration in non-stacked multiple chip ICs include:

- Printed Circuit Boards (PCBs) with multiple ICs on the same board
- SoCs with multiple cores in a chip
- Multi-Chip-Package (MCPs), where multiple chips are integrated in a single package [18]

MCPs stacked vertically, but not bonded with TSV interconnects include:

- SiPs, where chips are vertically stacked within a package, interconnected by wire-bonds to the substrate
- PoPs, where multiple chips are vertically stacked

SICs with TSVs are the next in evolution beyond SiP. In the short term, incremental evolution will present similar challenges to those already presented for SiP, except potential changes in chip to chip interconnects. In the medium to long term, as chip stacking becomes more prevalent and more complex chip stacks appear, test challenges will become increasingly more difficult. It is certain that new and additional DfT features will be needed to mitigate increased tester resource and time requirements as well as increased test complexity due to a large number of different chips in the same package. Although SICs have their advantages in terms of performance and power requirements, the manufacturing process introduces new challenges in terms of achieving high yield, testing and power constraints [14, 18, 25].

The manufacturing process of SICs is very different from non-stacked ICs - each chip in the SIC needs to be stacked, aligned and bonded [14–16, 18, 30]. SICs can be obtained by three stacking processes, *viz*, Die-to-Die (D2D), Wafer-to-Wafer (W2W) and Die-to-Wafer (D2W). In W2W stacking, complete wafers are stacked over one another, resulting in exponentially decreasing yields with increasing number of layers in the stack [20]. It is applicable on both D2W and D2D stacking.

While stacking, the orientation of the stacked chips has to be considered. Chips in a SIC can be connected through the active front-side, called the face, of a silicon chip, or the silicon substrate, called the back. There are three possible variations in this regard: face-to-face, back-to-back and face-to-back. In this context, the face of a chip is the side of the transistors and the metal interconnect layers and the back is the silicon substrate layer. Among the three possibilities, only face-to-back bonding is scalable to stacks of more than two chips [18], which is considered in this paper.

The test flow model for traditional non-stacked ICs comprise two test instances, *viz.* (i) wafer sort and (ii) package test. Wafer sort is motivated by the fact that packaging faulty products is more expensive than the test itself. By testing, unnecessary packing of faulty chips is avoided. For non-stacked chips, the only possible introduction of faults after wafer sort might occur while packaging the same chip. Therefore, the test performed at wafer sort is repeated at the package test. In case of SICs, there are four instances during the stacking process when faults may be introduced to any chip of the stack: (i) chip fabrication, (ii) stacking of each chip, (iii) once all chips have been stacked prior to packaging, (iv) post packaging [18]. Based on these steps, several test instances can be considered, one for each step that can introduce faults. It should be noted that testing after stacking includes testing the newly constructed TSVs. Chip-specific test schedules that are optimized for wafer sort do not consider testing of other chips in the stack. Similarly, test schedules that are optimized for the package test are not necessarily optimal for wafer sort. Thus, it can be seen that a complete view of test scheduling from wafer sort to package test is required to arrive at a minimal test time [26].

A major factor of the overall test cost besides test time comprises the DfT hardware. Various standards have been proposed for test architecture of non-stacked ICs, among

which one of the most successful is the IEEE 1149.1 [27]. The use of IEEE 1149.1 for accessing the cores of the individual chips has the following main advantages:

- It adheres to an existing standard.
- IEEE 1149.1 can be used for test access both in wafer sort and package test of SICs. Typically, after stacking, test access is only possible through the bottommost chip. For the remaining chips, dedicated test infrastructure TSVs are required to access the cores. When testing the chips individually in wafer sort, IEEE 1149.1 enables test access. The IEEE 1149.1 TAPs of different chips in the stack can be connected in series to enable test access for package test for all chips in the stack.
- For each chip in the stack, up to only five TSVs are required during test, which correspond to the five terminals of IEEE 1149.1.

To extend the test architecture used for non-stacked ICs to SICs, several factors must be considered:

- For SICs with TSVs all data flow in and out of the IC via the lowermost chip [14, 18].
- The test architecture must support wafer sort, intermediate tests and package test.

An efficient test architecture should contain DfT hardware, which is reused at all instances of test. Therefore, a core-based SIC complying to the IEEE 1149.1 standard requires the core tests scheduled for wafer sort and package tests with minimal test time, with total reutilization of the DfT hardware as elaborated in Section 6.

4 Test Architecture

In this section we first illustrate the test architecture for core-based non-stacked ICs and SICs based on IEEE 1149.1, followed by the test mechanism of the SIC provided with the IEEE 1149.1 test architecture.

4.1 Non-stacked IC

The test architecture of a non-stacked IC, considered in this paper, is shown in Fig.1, and the corresponding test schedule is illustrated in Fig.2. A chip consists of a number of cores that are accessed by an on-chip IEEE 1149.1 infrastructure [26]. The IEEE 1149.1 TAP may have up to five terminals, namely Test Data Input (TDI), Test Data Output (TDO), Test Mode Select (TMS), Test Clock (TCK) and an optional Test Reset (TRST) as shown in Fig.1. The scan chain of each core is accessed by the TAP controller via TDRs. If tests of multiple cores of a chip are to run concurrently in a session, these cores are connected in series on the IEEE 1149.1 interface, via a single TDR. In Fig.1, the IC contains three cores: Core1, Core2 and Core3. Core1 and Core2 comprise a single TDR, while Core3 is considered as a separate TDR. Only one TDR can be accessed at a time. This enforces the session concept that was introduced in Section 2. Only those cores of a chip that are in the same TDR can be tested concurrently. Consequently, if two cores are to be tested in sequence, in different sessions, they belong to different TDRs. Thus, if tests for more than one core of a chip are to be executed concurrently in a session, as shown in Fig.2, these cores are to be connected in series as one TDR. Since Core1 and Core2 are tested in the same session, denoted by $(Core1, Core2)$, the two cores are connected to the TAP controller by the same TDR, as seen in Fig. 1. Correspondingly, in the other session denoted as $(Core3)$, only Core3 is tested, which is connected to the TAP controller by a dedicated TDR. The resulting test schedule is illustrated in Fig. 2. The horizontal axis indicates the time required by the tests, while the vertical axis indicates the power dissipated by each session. The session on the left indicates the test session comprising the tests $(Core1, Core2)$, and the right session indicates the test of $(Core3)$.

4.2 SIC

A SIC is formed by stacking, aligning and bonding multiple chips and interconnecting them with TSVs. Figure 3 illustrates one such example, with two chips, where Chip2 has been stacked on top of Chip1. Chip1 contains Core1, Core2 and Core3, while Chip2 hosts Core4 and Core5.

During package test of the SIC in Fig. 3, the TDOup of the lower chip in the stack, Chip1, serves as the TDIdown of the chip on top, Chip2. The TDOup of the topmost chip is directed out via the TSVs by the TDOdown of Chip1. The TDIup and the TDOdown of the lowermost chip, TMS, TCK and an optional TRST serve as the package test interface for the SIC. A session of tests from one chip can be performed concurrently with a session of tests from another

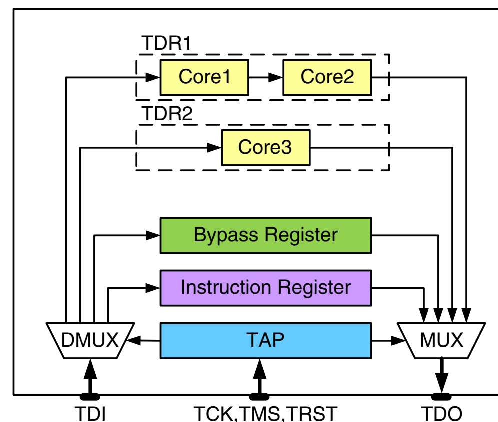

Fig. 1 Test architecture of a non-stacked chip with JTAG

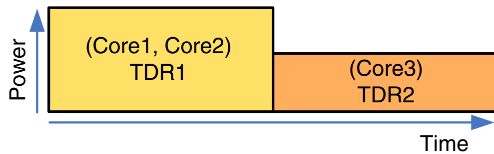

Fig. 2 Sessions formed by core tests: $S = ((Core1, Core2), (Core3))$

chip by selecting the corresponding TDRs by the respective on-chip TAPs of to the two chips.

4.3 Test mechanism of the SIC

The test process is conducted as follows.

1) The appropriate instruction is loaded onto the Boundary Scan Instruction Register in each chip.
2) A boundary-scan test instruction is shifted into the Instruction Register (IR) of the lowermost chip (Chip1, in Fig. 3) through the TDIdown.
3) The instruction is decoded by the decoder associated with the IR to generate the required control signals in order to properly configure the test logic.
4) A test pattern is shifted onto the selected TDRs through the respective TDIdown of the chips and then applied to the core(s) to be tested.
5) The test response is captured in a TDR.
6) The captured response is shifted out through the TDOup of the lower chip, which acts as the TDIdown of the chip above (Chip2, in Fig. 3). The test response, is shifted out through the TDOup of the topmost chip (Chip2, in Fig. 3), which takes a U-turn and exits each chip through the serial TDOdowns.
7) Each test response exits the SIC via the TDOdown of the lowermost chip. At the same time, a new test pattern can be shifted in through the TDIdown of the lower chip (Chip1, in Fig. 3).
8) Steps 4 to 7 are repeated until all test patterns are shifted in and applied, and all test responses are shifted out.
9) Interconnect tests are not performed concurrently with any core tests. Hence, the time taken for interconnect tests adds a constant test time to the test schedule of the SIC.

The TSV interconnect between two chips may be tested by using a special TDR called the boundary scan register, which connects all input/output pins and TSVs with special scan cells forming a shift register. The scan cells are transparent when the SIC is in functional mode, but in test mode, the scan cells are control points and observation points. Boundary scan registers are implemented on both the chips and both are used in TSV interconnect test. Test stimuli are applied on out-going TSVs and test responses are captured on in-coming TSVs. Since the boundary scan register is a separate TDR, testing of TSVs cannot be performed concurrently with any other test.

It should be noted that the TSV interconnect tests will contribute with a constant term to the test time and cannot be scheduled with any other core tests, according to the IEEE 1149.1 test architecture standard. Test of TSVs, using a chip level wrapper based on the IEEE 1149.1 standard, has been broadly explained in [17, 18]. Therefore, TSV interconnect tests will not be considered when addressing test scheduling in the remainder of the paper.

5 Problem Definition

Issues related to test planning for non-stacked ICs and SICs are presented in this section. Each chip is supported by an

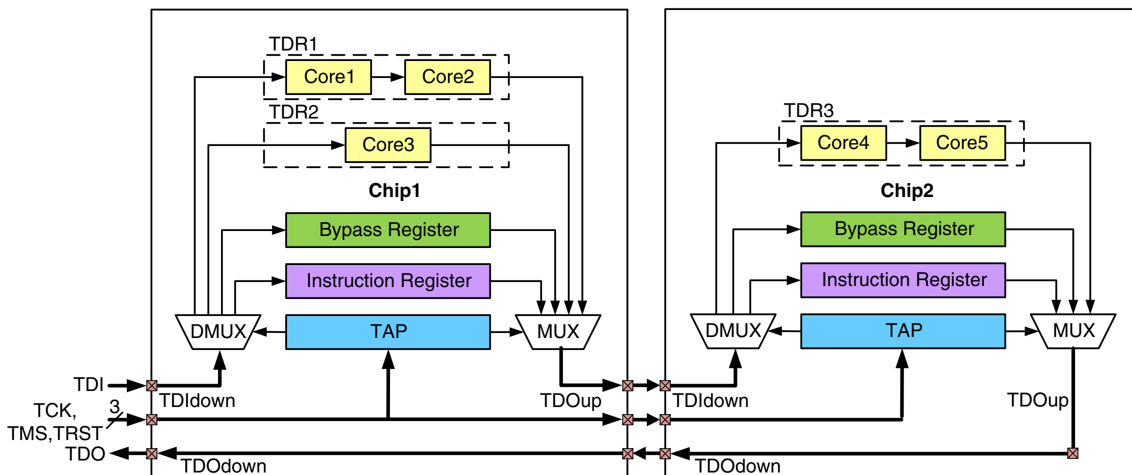

Fig. 3 Test architecture of a SIC with JTAG

Table 1 Notations for non-stacked ICs

Input data	
C	Set of cores in the IC
c	Each core $c \in C$
$l(c)$	Length of scan chain of core c
$p(c)$	Patterns required by core c
$w(c)$	Power dissipated by core c
w_{max}	Power constraint
α	Designer specified constant to estimate the cost of time
β	Designer specified constant to estimate the cost of DfT hardware
Calculated	
H	Set of TDRs
h	Each TDR $h \in H$
S	Set of sessions
s	Each session $s \in S$
$t(c)$	Time required to test core c
$t(s)$	Time required to test session s
$w(s)$	Power dissipated by session s
T	Total time required to test IC
$Cost(T, H)$	Test cost of IC defined in terms of T and H

IEEE 1149.1 based test infrastructure. The objective is to minimize the test cost, defined as the weighted sum of the test time and the DfT hardware, meeting a given power constraint.

5.1 Non-stacked IC

For a non-stacked IC supported by the IEEE 1149.1 test architecture, the notations are collated in Table 1. The values in Table 3 are used for the explanation of the terms, and Fig. 1 illustrates a typical architecture. The IC comprises of a set of $C = \{Core1, Core2, Core3\}$ cores, each denoted by c, where $c \in C$. Each core c has a scan chain of length $l(c)$, requires $p(c)$ test patterns, and dissipates $w(c)$ units of power. Each column in Table 3, represents a core of the SIC with the corresponding scan chain length $l(c)$, patterns required $p(c)$ and power dissipated $w(c)$. For example, Core1 has a scan chain of length $l(Core1) = 30$, requires $p(Core1) = 30$ patterns, and dissipates $w(Core1) = 50$ units of power during testing.

The test time for a core c is given as $t(c)$:

$$t(c) = (\delta + l(c)) \cdot p(c) + l(c) \tag{1}$$

where, δ accounts for the number of clock cycles required for update and capture, which is equal to 5 in the case of JTAG.

The time taken by Core1 maybe calculated as:

$$\begin{aligned} t(Core3) &= (\delta + l(Core3)) \cdot p(Core3) + l(Core3) \\ &= (5 + 70) \cdot 70 + 70 = 5320 \ \textit{time units} \end{aligned}$$

For the IEEE 1149.1 test architecture, there is a set of TDRs $H = \{TDR1, TDR2\}$, each denoted as h, where $h \in H$. Several cores may share a single TDR. For instance, Core1 and Core2 share TDR1.

The test schedule for the set of C cores consists of a set of S sessions. A set of cores are tested in each session s, where $s \in S$. As illustrated in Fig. 2, the set of sessions is denoted as $S = ((Core1, Core2), (Core3))$. Every core belongs to a unique session, $c \in s$. For example, Core1 and Core2 are tested in the first session, which is denoted as $s = (Core1, Core2)$. The test time $t(s)$ for any session is calculated as:

$$t(s) = \left(\delta + \sum_{c \in s} l(c)\right) \cdot \max_{\forall c \in s} p(c) + \sum_{c \in s} l(c) \tag{2}$$

The time taken by session $s = (Core1, Core2)$ is calculated as:

$$\begin{aligned} & t(Core1,\ Core2) \\ &\quad = (5 + l(Core1) + l(Core2)) \cdot \\ &\quad \max\{p(Core1),\ p(Core2)\} \\ &\quad +\{l(Core1) + l(Core2)\} \\ &\quad = (5 + 30 + 30) \cdot max(30, 30) \\ &\quad +(30 + 30) = 2010 timeunits \end{aligned}$$

The power dissipated while testing each session s is given by $w(s)$, which is the sum of the power dissipated by each individual core tested in the session:

$$w(s) = \sum_{c \in s} w(c) \tag{3}$$

The power dissipated by session $s = (Core1, Core2)$ is calculated as:

$$\begin{aligned} & w(Core1,\ Core2) \\ &\quad = w(Core1) + w(Core2) \\ &\quad = 50 + 40 = 90 units \end{aligned}$$

The overall test time for a test schedule T is given as the sum of the times taken by each session:

$$T = T_{ws} + T_{pt} = 2 \cdot \sum_{\forall s \in S} t(s) \tag{4}$$

The time required by the test schedule is multiplied by 2, as the same test schedule is applied both at wafer sort and package test.

In case of the given IC:

$$T = 2 \cdot (t(Core1, Core2) + t(Core3)) = 2 \cdot (2010 + 5320) = 14660 \; time\ units$$

The DfT hardware cost is directly related to the number of sessions, since each session corresponds to one TDR; hence, $|H| = |S|$, which is $|H| = 2$ as seen in Fig. 1.

The test cost for any given configuration of the non-stacked IC is calculated as follows:

$$Cost(T, H) = \alpha \cdot T + \beta \cdot |H| \quad s.t. w(s) \leq w_{max}, \forall s \tag{5}$$

where,

the power dissipated by each session, $w(s)$, is within the power constraint w_{max}, and α and β are weight constants set by the designer depending on the co-relation between test time and TDR of the particular system.

The cost in this example is calculated as:

$$Cost(T, H) = 1 \cdot 14660 + 2000 \cdot 2 = 18660 units$$

where the weighting constants are set as: $\alpha = 1$ and $\beta = 2000$.

The problem is to find a test schedule such that the total test time and the number of TDRs required result in a minimized cost while meeting the power constraint.

5.2 SIC

For a SIC design having a stack of multiple chips, where each chip is supported by the IEEE 1149.1 test architecture, the notations are collated in Table 2. Figure 3, along with the values provided in Table 3, is used for illustration. The SIC, which comprises of a set of $N = \{Chip1, Chip2\}$ chips in the stack. Each chip is denoted as n, $n \in N$, and has a set of $C(n)$ cores, each denoted by c, where $c \in C(n)$. For example, Chip1 comprises of three cores $C(Chip1) = \{Core1, Core2, Core3\}$. A core c has a scan chain of length $l(c)$, requires $p(c)$ test patterns, and the power dissipated is $w(c)$.

For the IEEE 1149.1 test architecture, there is a set of TDRs $H(Chip1) = \{TDR1, TDR2\}$, each denoted by h, where $h \in H(n)$.

The test schedule in chip n comprises $S(n)$ sessions, each denoted by s, where $s \in S(n)$. The test time $t(s)$ of a session s is given as in Eq. 2:

$$t(s) = \left(\delta + \sum_{c \in s} l(c)\right) \cdot \max_{\forall c \in s} p(c) + \sum_{c \in s} l(c) \tag{6}$$

Table 2 Notations for SICs

Input data	
N	Set of chips in the SIC
n	Each chip $n \in N$
$C(n)$	Set of cores in the chip n
c	Each core $c \in C(n)$
$l(c)$	Length of scan chain of core c
$p(c)$	Patterns required by core c
$w(c)$	Power dissipated to test core c
w_{max}	Power constraint
α	Designer specified constant to estimate the cost of time
β	Designer specified constant to estimate the cost of DfT hardware
Calculated	
$H(n)$	Set of TDRs in the chip n
h	Each TDR $h \in H(n)$
$S(n)$	Set of sessions in the chip n
s	Each session $s \in S(n)$
$t(c)$	Time required to test core c
$t(s)$	Time required to test session s
$w(s)$	Power dissipated by session s
$t(n)$	Wafer sort time of chip n
T_{ws}	Total wafer sort time required
S_{pt}	Set of sessions during package test
s_{pt}	Each session during package test $s_{pt} \in S_{pt}$
$t(s_{pt})$	Time required to test session s_{pt}
$w(s_{pt})$	Power dissipated by session s_{pt}
T_{pt}	Package test time
H	TDRs required to test SIC
T	Total time required to test SIC
$Cost_{SIC}(T, H)$	Test cost of SIC defined in terms of T and H

The power dissipated while testing a session s, is given by $w(s)$, is given as in Eq. 3:

$$w(s) = \sum_{c \in s} w(c) \tag{7}$$

The time taken by each chip n during wafer sort is $t(n)$, which is calculated similar to the total time in case of non-stacked ICs in Eq. 4:

$$t(n) = \sum_{\forall s \in S(n)} t(s) \tag{8}$$

Thus, the total time taken for wafer sort of the SIC, T_{ws}, is given as:

$$T_{ws} = \sum_{\forall n \in N} t(n) \tag{9}$$

In the given example, the total wafer sort time is calculated as the sum of the wafer sort time for Chip1 $t(Chip1) = 7330$ and that of Chip2 $t(Chip1) = 7450$:

$$T_{ws} = t(Chip1) + t(Chip2)$$
$$= 7330 + 7450 = 14780 timeunits$$

For package test of the SIC, a test schedule is formed with S_{pt} sessions. Each core c belongs to a unique session s_{pt}, where $s_{pt} \in S_{pt}$. The test time $t(s_{pt})$ is represented in a similar manner as in case of wafer sort (6):

$$t(s_{pt}) = \left(\delta + \sum_{c \in s_{pt}} l(c)\right) \cdot \max_{\forall c \in s_{pt}} p(c) + \sum_{c \in s_{pt}} l(c) \quad (10)$$

The overall test time for package test of the SIC, T_{pt}, is given as the sum of the time taken by all sessions during package test, similar to Eq. 8:

$$T_{pt} = \sum_{\forall s_{pt} \in S_{pt}} t(s_{pt}) \quad (11)$$

The power dissipated by each session s_{pt} of the package test, $w(s_{pt})$, is the sum of the power dissipated by each core belonging to all chips which are tested during the session, similar to Eq. 7:

$$w(s_{pt}) = \sum_{c \in s_{pt}} w(c) \quad (12)$$

The total time taken to test the SIC, T, is calculated as:

$$T = T_{ws} + T_{pt} \quad (13)$$

Assuming the package test schedule as $S_{pt} = ((Core1, Core2, Core3), (Core4, Core5))$, we get $T_{pt} = 14780$. Therefore, we can calculate T for the given SIC as:

$$T = T_{ws} + T_{pt} = 14780 + 14780 = 29560 \; time \; units$$

The DfT hardware, H, is given by the total number of TDRs, which is equal to the sum of the number of sessions during wafer sort of each chip.

$$H = \sum_{\forall n \in N} |H(n)| = \sum_{\forall n \in N} |S(n)| \quad (14)$$

For the given SIC, Chip1 and Chip2 require one TDR each during wafer sort. Thus, H may be calculated as:

$$H = |H(Chip1)| + |H(Chip2)| = 1 + 1 = 2$$

Table 3 Data used for the SIC

	Chip 1			Chip 2	
	Core1	Core2	Core3	Core4	Core5
Scan chain length $l(c)$	30	30	70	70	30
Patterns required $p(c)$	30	30	70	70	10
Power dissipated $w(c)$	50	40	40	20	10
Maximum power constraint w_{max}:				100	
Time coefficient α:				1	
Hardware coefficient β:				2000	

The overall test cost for any given configuration of the SIC is similar to Eq. 5:

$$Cost_{SIC}(T, H) = \alpha \cdot T + \beta \cdot H$$
$$s.t. \quad w(s_{pt}) \leq w_{max}, \forall s_{pt} \quad (15)$$

The power dissipated by any session during package test, $w(s_{pt})$, is within the power constraint w_{max}.

The motive is to obtain the wafer sort schedule for each chip in the stack, and the package test schedule; such that the overall test time and the total number of TDRs required by all the N chips during wafer sort result in a minimized cost while meeting the power constraint.

6 Motivational Example

In this section we motivate the need of test planning of SICs, while meeting a power constraint, by demonstrating the trade-off between test time and DfT hardware. Figure 3 illustrates the SIC, and the corresponding values are provided in Table 3. The SIC comprises of two chips, $N = \{Chip1, Chip2\}$. The five cores are distributed among the two chips as: $C(Chip1) = \{Core1, Core2, Core3\}$ and $C(Chip2) = \{Core4, Core5\}$. Each column in Table 3, represents a core of the SIC with the corresponding scan chain length $l(c)$, patterns required $p(c)$ and power dissipated $w(c)$. It is assumed that the set value of the power constraint, $w_{max} = 100 units$. The constants α and β are 1 and 2000 respectively.

Different test plans for the SIC are explored. The results are presented in Tables 4 and 5. Table 4 lists five test plans generated for the given SIC, while Table 5 shows the corresponding test costs. Each test plan is analyzed individually in the remainder of this section.

For the *first test plan*, all three cores in Chip1 share a common TDR, $|H(Chip1)| = 1$. Hence, the wafer sort schedule of Chip1 comprises of a single session, $S(Chip1) = (Core1, Core2, Core3)$, and the

Table 4 Test schedule alternatives

Test Plan	Wafer sort Chip 1 $S(Chip1)$	Chip 2 $S(Chip2)$	Package test S_{pt}
1	(Core1, Core2, Core3)	(Core4, Core5)	(Core1, Core2, Core3), (Core4, Core5)
2	(Core1, Core2, Core3)	(Core4), (Core5)	(Core1, Core2, Core3), (Core4), (Core5)
3	(Core1, Core2), (Core3)	(Core4, Core5)	(Core1, Core2), (Core3, Core4, Core5)
4	(Core1, Core2), (Core3)	(Core4), (Core5)	(Core1, Core2, Core5), (Core3), (Core4)
5	(Core1, Core2), (Core3)	(Core4), (Core5)	(Core1, Core2, Core5), (Core3, Core4)

corresponding test time is calculated to be $t(Chip1) = 9580 timeunits$ as in Eq. 8.

$$\begin{aligned} t(Chip1) &= \{l(Core1) + l(Core2) + l(Core3) + 5\} \\ &\quad \cdot \ max\,\{p(Core1), p(Core2), p(Core3)\} \\ &\quad + \{l(Core1) + l(Core2) + l(Core3)\} \\ &= (30 + 30 + 70 + 5) \cdot max(30, 30, 70) \\ &\quad + (30 + 30 + 70) = 9580\ time\ units \end{aligned} \tag{16}$$

Similarly, for Chip2, Core4 and Core5 share a common TDR, $|H(Chip2)| = 1$ for the third test plan. Hence, the wafer sort schedule of Chip2 is also comprised of single session, $S(Chip2) = (Core4, Core5)$, and the corresponding test time is $t(Chip2) = 7450 timeunits$. The package test schedule consists of two sessions $S_{pt} = ((Core1, Core2, Core3), (Core4, Core5))$, which requires a test time of $T_{pt} = 17030 timeunits$. Thus, the total time taken to test the 3D Stacked IC with the first test plan in Table 4 is calculated from Eq. 13 as the sum of the time taken for the wafer sort of each chip and the package test time to be $T = T_{ws} + T_{pt} = 34060 timeunits$.

The cores in either of the chips share a single TDR, hence from Eq. 14, we get $H = |H(Chip1)| + |H(Chip2)| = 2$. Therefore, the cost incurred $Cost_{SIC}(T, H) = 38060 units$ by the test plan proposed in case 1, calculated as in Eq. 15. Alccording to Eq. 12 the power dissipated for session $s_{pt} = (Core1, Core2, Core3)$ is $w(Core1, Core2, Core3) = 130 \nleq 100 units$, violating the power constraint. Hence, the first test plan is not valid.

For the *second test plan*, it can be seen that the overall test cost is reduced, despite an increase in the cost related to the DfT hardware. This is because the cost related to the test time more than compensates for it. Although the overall test cost for the second test plan is lower than that of the first, the test plan is infeasible due to the violation of the power constraint, similar to the first test plan.

For the *third test plan* in Table 4, during wafer sort of Chip1, (Core1, Core2) + (Core3), *i.e.*, Core1 and Core2 share a common TDR, while Core3 forms a separate session with a dedicated TDR. The test time of $t(Chip1) = 7330 timeunits$. For Chip2 session $S(Chip2) = (Core4, Core5)$ which means Core4 and Core5 share a TDR. The wafer sort time of $t(Chip2) = 7450 timeunits$. In Chip2, the hardware cost is one TDR. All power constraints are met during the wafer sorts. The cost incurred during package test is $Cost_{pt} = 14430 units$. Core1 and Core2 form a package test session, while Core3, Core4 and Core5 form a different session, thus giving $S_{pt} = ((Core1, Core2), (Core3, Core4, Core5))$, and takes a test time of $T_{pt} = 35210 timeunits$. Each session meets the power constraint of $W_{max} = 100 units$. This results in a total test cost of $Cost_{case3} = 35210 units$. The third test plan has a lower test cost than the first test plan while meeting the power constraint.

Table 5 Test costs achieved

Test Plan	Wafer sort Chip 1 Time $t(Chip1)$	Chip 1 TDR $\lvert H(Chip1)\rvert$	Chip 2 Time $t(Chip2)$	Chip 2 TDR $\lvert H(Chip2)\rvert$	Package test Time T_{pt}	Cost Time T	Hardware H	Overall $Cost_{SIC}(T, H)$	Power Constraint $w_{pt} \leq w_{max}$
1	9580	1	7450	1	17030	34060	4000	38060	$130 \nleq 100$
2	9580	1	5700	2	15280	30560	6000	36560	$130 \nleq 100$
3	7330	2	7450	1	14430	29210	6000	35210	$90 \leq 100$
4	7330	2	5700	2	13580	26610	8000	34610	$100 \leq 100$
5	7330	2	5700	2	13230	26260	8000	34260	$100 \leq 100$

For the *fourth test plan*, the wafer sort test time for Chip1 is $t(Chip1) = 7330 timeunits$, when two TDRs are used. Core1 and Core2 form a single session and Core3 forms another session, *i.e.*, $S(Chip1) = ((Core1, Core2), (Core3))$. For Chip2, where Core4 and Core5 are tested in separate sessions, represented as $S(Chip2) = ((Core4), (Core5))$, the time taken is $t(Chip2) = 5700 timeunits$. During package test, where Core1, Core2 and Core5 form a session, while Core3 and Core4 forms individual sessions $S_{pt} = ((Core1, Core2, Core5), (Core3), (Core4))$, and each session is power constrained. The total test cost sums up to $Cost_{case4} = 34610 units$, which is less than $Cost_{case3}$. Thus, it can be seen that although the cost incurred at wafer sort for the fourth test plan is more than that of the third test plan, the overall test cost for the fourth test plan is less than that of the third test plan.

In the *fifth test plan*, where the wafer sort configurations for both Chip1 and Chip2 are same as in case 4, but during package test Core1, Core2 and Core5 are tested simultaneously during a package test session, while Core3 and Core4 are tested together in another session, giving $S_{pt} = ((Core1, Core2, Core5), (Core3, Core4))$, each meeting the power constraint, when the cost incurred is $Cost_{pt} = 13230 units$. The total test cost is $Cost_{case5} = 34260 units$, which is even less than $Cost_{case4}$. Hence, by properly forming sessions during package test, the test cost can be further reduced and satisfy power constraints.

From the above studies, by comparing the wafer sort schedules of both Chip1 and Chip2 in the first test plan against the fifth test plan, it can be seen that the test time can be reduced by increasing the number of sessions and thereby increasing the number of TDRs. Although, an increased number of sessions implies increased DfT hardware cost. Hence, it is important to find a proper trade-off between the DfT hardware cost and the test time, meeting a power constraint, to obtain a test plan with the minimum test cost.

7 Proposed Approaches

Test planning for core-based ICs and SICs, under a power constraint, can be portrayed as a constrained bin-packing problem, which is NP-hard. Therefore, in this section we present heuristics designed to minimize the test cost for non-stacked ICs and SICs, while meeting a given power constraint. The test cost is given as a weighted sum of the test time and DfT hardware. Table 3 is used for the elaboration of the algorithms.

7.1 Non-stacked IC

For non-stacked ICs, the identical test schedules are applied both during wafer sort and package test. Consequently, minimizing the test cost of either the wafer sort or package test, minimizes the overall test cost. An iterative heuristic is proposed in Algorithm 1 to minimize the test cost of non-stacked ICs. The basic procedure of the algorithm is as follows:

The set of cores C comprises of all cores in the IC, sorted in descending order of the number of patterns $p(c)$. One core c is selected at a time from the given set of cores C. Core c is first assigned to a dedicated TDR, and the corresponding cost $Cost(T, H)$, is calculated. Thereafter, core c is included in each of the previous sessions s, one at a time. The corresponding test cost $\acute{C}ost(T, H)$, is calculated, provided that the power constraint is met. The test schedule is finalized with the lowest test cost obtained on including core c.

For example, considering Chip1 Table 3, illustrated by Fig. 1, we have a set of three cores $C = \{Core1, Core2, Core3\}$. On sorting the cores in descending order of the number of patterns required $p(c)$, the elements of C can be rearranged as $C = \{Core3, Core1, Core2\}$. First we select Core3, which forms the first session $S = (Core3)$. The corresponding test cost is calculated as $Cost(T, H) = 12640$. Core1 is selected next, which gives a test cost of $Cost(T, H) = 16800$ with a dedicated session, and $\acute{C}ost = 16900$ along with Core3 in the first session, while meeting the power constraint. Since, the cost is lower with a new session, the test schedule is updated to $S = ((Core3), (Core1))$. Finally, Core2 is selected, which gives a test cost $Cost(T, H) = 20960$ with a new session. By including Core2 on the same session as Core3, the test cost obtained is $\acute{C}ost(T, H) = 21060$, which is higher than the test cost obtained by including Core2 in the second session with Core1, calculated as $\acute{C}ost(T, H) = 18660$, meeting the power constraint during both instances. In addition since $\acute{C}ost(T, H) < Cost(T, H)$, the cost of including Core2 in the same session as Core1 gives the lowest test cost. Thus, the test schedule for Chip1 can be given as $S = ((Core3), (Core1, Core2))$.

Algorithm 1 Test plan for non-stacked ICs

```
1:  Given: w_max, C, l(c), p(c), w(c), α, β
2:  Initialize: Cost(T, H) = 0
3:  Sort C in descending order of p(c)
4:  for all c ∈ C do
5:      Calculate cost in new session: Cost(T, H)
6:      for Each existing session s do
7:          if Power constraint is met, i.e., w(s) + w(c) ≤ w_max then
8:              Cost in existing session: Ćost(T, H)
9:              if Ćost(T, H) ≤ Cost(T, H) then
10:                 Cost(T, H) = Ćost(T, H)
11:                 Relocate core c to session S
12:             end if
13:         end if
14:     end for
15:     Update: Session S ∪ c
16: end for
17: Output Test plan: Cost(T, H), S
```

7.2 SIC

A test planning heuristic to minimize the test cost of SICs is presented in Algorithm 2. Since, wafer sort of individual chips is followed by the package test of all chips together, in case of SICs, optimizing the test plan of each chip individually may lead to a suboptimal test plan for the SIC. Hence, this heuristic takes into account both wafer sort and package test simultaneously, to arrive at a test plan. The principle behind the proposed test planning heuristic is:

One core c is selected at a time from the given set of all cores $C(n) \quad \forall n \in N$, in the SIC. Core c is first assigned a dedicated TDR, such that only the selected core c is tested in a session s during wafer sort. Next, core c is included in the first package test session s_{pt}, subject to complying with the power constraint, and the corresponding cost $Cost(T, H)_{SIC}$ is recorded. The total cost for each such possible package test session is calculated as $\acute{C}ost(T, H)_{SIC}$, after including wafer sort session s, using Eq. 15. Finally the wafer sort session s is tested in a dedicated package test session s_{pt}, and the corresponding cost $\acute{C}ost(T, H)_{SIC}$ is updated. Thereafter, core c is included in each existing wafer sort session s, one at a time. The test cost for the SIC, $\acute{C}ost(T, H)_{SIC}$, is recalculated after including wafer sort session s in each possible package test session s_{pt} as earlier. If, during any iteration, $\acute{C}ost(T, H)_{SIC}$ is lower than $Cost(T, H)_{SIC}$, then the test schedule is recorded with wafer sort session s in the current package test session s_{pt}. It may be noted here that during package test, wafer sort session s belonging to chip n may not be scheduled in any package test session s_{pt} containing another session from the same chip (n).

For example, considering Table 3, illustrated by Fig. 3, we have a set of five cores $C(Chip1, Chip2) = \{Core1, Core2, Core3, Core4, Core5\}$, in the SIC. On sorting the cores in descending order of the number of patterns required $p(c)$, the elements of $C(n)$ can be rearranged as $C = \{Core3, Core4, Core1, Core2, Core5\}$. First, we select Core3, which forms the first package test session $S_{pt} = (Core3)$. The corresponding test cost is calculated as $Cost(T, H) = 12640$. Core4 is selected second, which gives a test cost of $Cost(T, H) = 25280$ in a new wafer sort as well as package test session, while $\acute{C}ost = 24930$ along with Core3 in the first session, complying with the power constraints. Since, the cost is lower in the existing session, the test schedule is updated to $S_{pt} = (Core3, Core4)$. Next, Core1 is selected, which violates the power constraint when included in the same session with Core3 and Core4, $S_{pt} = (Core3, Core4, Core1)$, with $w(s_{pt}) = 110 > w_{max}$. Hence, a new session is inevitable, with the test schedule as $S_{pt} = ((Core3, Core4), (Core1))$ and $Cost(T, H) = 29090$. Similarly, Core2 is selected, which gives a minimum test cost $Cost(T, H) = 30950$, with a package test schedule $S_{pt} = ((Core3, Core4), (Core1, Core2))$. Eventually, Core5 gives a minimum test cost $Cost(T, H) = 33710$, with a package test schedule $S_{pt} = ((Core3, Core4), (Core1, Core2), (Core5))$. The wafer sort schedules can be obtained as $S(Chip1) = ((Core3), (Core1, Core2))$, and $S(Chip2) = ((Core4), (Core5))$.

8 Experimental Results

To demonstrate the benefits of the test planning approach, proposed in Section 7, for non-stacked ICs and SICs, the test planning algorithm was implemented on several benchmark designs minimizing test cost, to validate the proposal. A near optimal test cost for the benchmark designs is achieved by Simulated Annealing, and finally the two results are compared to demonstrate the performance of the proposed heuristic.

The experiments have been performed on the following six ITC'02 benchmark SoC designs:

p22810, p93791, g1023, d695, h953 and d281.

The following assumptions were made while considering the non-stacked ITC'02 SoC benchmarks as SICs:

- The modules in the benchmark SoC designs are also considered as cores in a non-stacked IC;
- All scan elements (inputs, outputs, and scan cells) in a core are connected to a single scan-chain;
- Modules without any scan chains are not considered;
- SIC designs are constructed by using multiple ITC'02 benchmarks;
- The constant α for all designs is set to 1.0;
- Several experiments were performed with varying values of β on the selected benchmark designs, and the most suitable value of β was found by dividing the test time of the core with the that requires the maximum test time, $t(c)$, by the number of cores, $|C(n)|$.

$$\beta = \frac{t(c)_{max}}{|C(n)|} \tag{17}$$

The intention is to find a β such that the DfT hardware and test time are in similar order of magnitude.

Algorithm 2 Test plan for 3D Stacked ICs with N chips in the stack

1: **Given:** w_{max}, N, $C(n)$, $l(c)$, $p(c)$, $w(c)$, α, β
2: **Initialize:** $Cost_{SIC}(T, H) = 0$
3: **Sort** $C(n)$ in descending order of $p(c)$
4: **for all** $c \in C$ **do**
5: **Calculate:** Cost in existing package test session:
6: **for** Each existing session s_{pt} **do**
7: **if** Power constraint is met, *i.e.*, $w(s_{pt}) + w(c) \leq w_{max}$ **then**
8: **Calculate:** $\acute{C}ost(T, H)_{SIC}$
9: **if** $\acute{C}ost(T, H)_{SIC} \leq Cost(T, H)_{SIC}$ **then**
10: $Cost(T, H)_{SIC} = \acute{C}ost(T, H)_{SIC}$
11: Relocate core c to session S_{pt}
12: **end if**
13: **end if**
14: **end for**
15: **Calculate:** Cost in existing wafer sort session:
16: **for** Each existing session s **do**
17: **if** Power constraint is met, *i.e.*, $w(s) + w(c) \leq w_{max}$ **then**
18: **Calculate:** $\acute{C}ost(T, H)_{SIC}$
19: **Calculate:** Cost in existing package test session:
20: **for** Each existing session s_{pt} **do**
21: **if** Power constraint is met, *i.e.*, $w(s_{pt}) + w(s) \leq w_{max}$ **then**
22: **Calculate:** $\acute{C}ost(T, H)_{SIC}$
23: **if** $\acute{C}ost(T, H)_{SIC} \leq Cost(T, H)_{SIC}$ **then**
24: $Cost(T, H)_{SIC} = \acute{C}ost(T, H)_{SIC}$
25: Relocate core c to session S_{pt}
26: **end if**
27: **end if**
28: **end for**
29: **end if**
30: **end for**
31: **if** $\acute{C}ost \leq Cost$ **then**
32: $Cost = \acute{C}ost$
33: Update: $S \cup c$
$S_{pt} \cup c$
$C(n) \cap c$
34: **end if**
35: **end for**
36: **Output** Test plan: $Cost_{SIC}(T, H)$, $S(n)$, S_{pt}

The minimized test cost obtained by the proposed approach is compared against the test cost obtained by Simulated Annealing, which models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy. Simulated Annealing was used to reach a near optimal test cost in a manner stated below:

1) Simulated Annealing algorithm is initiated with a random test plan, each core being allotted to a unique TDR. The cost is calculated by Eq. 15, which serves as the first trial $point$.

   A starting temperature τ_0 is assumed to be the temperature of the initial state of the system. τ_0 in this case was chosen to be $10units$, which provided sufficiently many iterations until the stopping criteria in each case.

   The algorithm iterates in the following manner until the stopping criteria is reached.
2) During each iteration, a new $point$ is generated, which is a previously unexplored test plan, in the following way: A core c is selected randomly.

   A random TDR, h, is selected, which is not the same as the TDR to which core c belongs.

   Core c is reallocated to TDR h, and the new cost is calculated by Eq. 15 iff $w(s) \leq w_{max}$.
3) The distance Δ of the new point from the previous point is the difference of the cost calculated at the new point generated and the previous trial point. In other words, $\Delta = Cost - \acute{C}ost$. In this paper we have assumed temperature to be decreasing exponentially, hence allowing exploration of all possible configurations. The trial point distance distribution, *i.e.*, the extent of the search, equals the current temperature with a uniformly random direction.
4) The algorithm not only accepts all new points that lower the overall test cost, but also, according to the acceptance function, points that raise the overall test cost. By accepting points that raise the overall test cost, the algorithm avoids being trapped in local minima in early iterations and is able to explore globally for better solutions. The acceptance parameter being:

$$\frac{1}{1 + exp\left(\frac{\Delta}{\tau}\right)} \tag{18}$$

   where,
   τ = current temperature.

   The probability of acceptance is maintained between 0 and 0.5. Smaller temperature or larger Δ leads to smaller acceptance probability.

Table 6 Test cost for non-stacked ICs

No.	Design	No. of Cores	Heuristic TDR	Heuristic Time	Heuristic Cost	Simulated annealing TDR	Simulated annealing Time	Simulated annealing Cost	Cost difference with Simulated annealing (%)
1	p22810	22	12	485857	497425	13	479130	491662	1.15
2	p93791	13	6	598922	605096	9	587300	596561	1.14
3	g1023	12	5	44950	46270	5	44950	46270	0.00
4	d695	8	4	37453	38533	5	35063	36413	5.50
5	h953	7	5	236220	253465	5	236220	253465	0.00
6	d281	5	3	111210	120306	3	111210	120306	0.00
Average:									1.34

5) The temperature after each iteration is lowered as per:

$$\tau_i = \frac{\tau_0}{log(k)} \tag{19}$$

where,
the annealing parameter, k, is the number of iterations since the latest reannealing.

6) The algorithm reanneals at certain intervals. Reannealing sets the annealing parameters to lower values than the iteration number, thus raising the temperature in each dimension. The annealing parameters depend on the values of estimated gradients of the cost function in each dimension. The basic formula is

$$k_i = log\left(\frac{\tau_0}{\tau_i} \cdot \frac{max_j(g_j)}{g_i}\right) \tag{20}$$

where,
k_i is the annealing parameter for iteration i.
τ_0 is the initial temperature.
τ_i is the current temperature at iteration i.
g_i is the total reduction is $Cost$ after i iterations, times the total reduction in temperature over the i iterations.

Reannealing safeguards the annealing parameter values against improper values.

For this system, reannealing intervals were chosen, as a product of the number of cores in the SIC and the total number of TDRs, which ensured that each core test would traverse all plausible sessions.

7) The algorithm stops when the average change in the objective function is small relative to the function tolerance, or when it reaches any other stopping criterion.

Thus, in this case, when the change in temperature was higher than the change in the test cost, the process was terminated. In other words, between two consecutive iterations i and $i-1$, if:

$$\frac{\tau_0}{log(k(i))} - \frac{\tau_0}{log(k(i+1))} > Cost - \acute{C}ost$$
$$\tau_i - \tau_{i+1} > \Delta \tag{21}$$

It refers to the temperature below which no more cores can be allocated to different TDRs.

Table 7 Test cost for SICs with 2 and 3 chips

No. of chips	Design nos.	Total no. of Cores	Heuristic TDR	Heuristic Time	Heuristic Cost	Simulated annealing TDR	Simulated annealing Time	Simulated annealing Cost	Cost difference with Simulated annealing (%)
2	1,2	35	18	1087919	1103905	22	1066430	1085968	1.62
	2,3	25	11	643872	655204	17	629525	647038	1.25
	3,4	20	9	82403	84257	13	77288	79966	5.09
	4,5	15	9	275037	286038	10	272647	284871	0.41
	5,6	12	8	348794	368171	8	337646	357023	3.03
Average:									2.28
3	2,3,4	33	15	681325	690710	22	664588	678352	1.79
	3,4,5	27	14	319987	326132	18	314872	322773	1.03
	4,5,6	20	12	386247	397834	13	372709	385262	3.16
Average:									1.99

Simulated Annealing in general gives a fairly near optimal result. However, repeated annealing schedule is expensive in terms of computation time, which rises exponentially with increasing number of cores in the SIC. In addition, the various parameters for the Simulated Annealing algorithm, *viz.*, temperature, reannealing and acceptance; if not properly set, may further raise the computation time. On the contrary, the proposed heuristic, which is customized to the specific problem takes advantage of extra information about the system, and often performs at par with general methods as Simulated Annealing. Hence, the proposed heuristic in Section 7 is compared against the cost obtained by Simulated Annealing for non-stacked ICs, SICs with 2 and 3 chips in the stack, as is visible in Tables 6 and 7 below.

In case of non-stacked ICs, the minimized overall test cost under a power constraint is presented in Table 6. In the table, each row corresponds to a SoC benchmark design, identified in the second column. The number of cores in each design is shown in column three. The first group of three columns, entitled Heuristic, show the minimal test cost of the respective designs as obtained by the algorithm proposed in Section 7. The later group of columns depict the minimized cost as obtained by Simulated Annealing. In both group of three columns, the first column shows the number of TDRs required by the obtained test schedule, followed by the test time for the same schedule. The third column is the cost obtained by applying (15). The final column shows the percentage difference in the cost obtained by the proposed heuristic to that of that cost obtained by Simulated Annealing. It can be observed that, on an average the proposed approach arrives at a cost which is 1.3 % greater than the cost of the test plan obtained by Simulated Annealing, however, with considerably lesser computation time.

In Table 7, the package test cost for various SIC designs constructed by stacking the six benchmark designs in Table 6 are shown. The number of chips that have been stacked to realize the SIC is shown in the leftmost column. The top group of five rows have SICs with two chips in the stack, followed by a group of three rows having three chips in the stack. The second column from left shows the benchmark designs that have been used for the stack, which correspond to the serial numbers used in Table 6. For instance, the first SIC design contains two chips in the stack, 1 and 2, which refers to p22810 and p93791 respectively. The third column lists the total number of cores in the SIC. The groups of three columns on the left list the number of TDRs, the test time and the test cost respectively for the proposed approach. Similarly, the second group of three columns list the number of TDRs, the test time and the test cost respectively for Simulated Annealing. The rightmost column enlists the percentage reduction in the test cost obtained by using Simulated Annealing over heuristic tests. In Table 7, it can be seen that in case of SICs with 2 chips, the average reduction in test cost obtained is 2.28 % and that with 3 chips in the stack is 1.99 %. Furthermore, it was found that the test cost obtained by the proposed heuristic was higher than the test cost obtained by Simulated Annealing by a maximum of 5.50 %. However, since the Simulated Annealing algorithm in this case is required to nearly exhaust the search space, the CPU time required for the algorithm rises exponentially with increasing number of cores. On the other hand, the CPU time required by the heuristic increases linearly with the number of cores in the SIC. Table 8 shows the rounded values of the time taken to execute the heuristic and Simulated Annealing on all designs listed in Tables 6 and 7. It can be seen that Simulated Annealing arrives at the desired test plan with considerably longer computation time as compared to the heuristic. The heuristic arrives at test plan within a matter of seconds in case of non-stacked ICs, as well as for SICs. In case of non-stacked ICs, Simulated Annealing arrives at a test plan for all six designs in just over 6 minutes. However, in case of SICs with two and three chips in the stack, Simulated Annealing requires almost 6 hours and 5 hours respectively, to determine all test plans, whereas heuristics does it in seconds/minutes.

Table 8 Comparison of total execution times of the proposed heuristic against Simulated Annealing

No. of chips	Heuristic T_1 *(Seconds)*	Simulated Annealing T_2 *(Seconds)*	Ratio T_2/T_1
1	3	376	1.3×10^2
2	10	20000	2.0×10^4
3	6	17000	2.9×10^3

9 Conclusion

The cost of testing is a major contributor to the total production cost of SICs. The test cost, in turn, depends on both test time and DfT hardware, represented by the number of TDRs. In this paper, we have minimized the test cost of SICs, defining it as a function of test time and DfT hardware. We have considered core based non-stacked ICs and SICs based on the IEEE 1149.1 test architecture standard as systems for testing. The test cost is minimized by co-optimizing test time and DfT hardware. A test planning algorithm was proposed for scheduling tests while meeting power constraints, which addressed the following three objectives:

1) For a non-stacked IC, where the same test schedule is applied during wafer sort and package tests, the tests of

all the cores are grouped in sessions such that the cost is minimized by co-optimizing test time and the number of required TDRs.

2) For a SIC, where each chip is tested individually during wafer sort and jointly during package test, cost is minimized by forming sessions from different chips concurrently during the package test.
3) The algorithm for test scheduling of SICs with two chips is extended to SICs with any number of chips forming the stack.

The test planning algorithm was implemented on several ITC'02 SoC benchmarks, and the results were compared against the test costs obtained by Simulated Annealing. It was observed that the proposed test planning algorithm arrived at a solution with a considerably lower CPU time, and the test cost obtained was, to a maximum 5.50 % (Table 6, Case4 and Table 7, SIC with designs 3 and 4) higher than the test cost obtained by Simulated Annealing both for non-stacked ICs and SICs.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Andrews J (1994) An embedded JTAG, system test architecture. In: IEEE electro international conference, pp 691–695
2. Bleeker H, Eijnden PVD, Jong FD (1993) Boundary-scan test: a practical approach. Kluwer Academic Publishers
3. Caldwell T, Langford B (1992) Is IEEE 1149.1 boundary scan cost effective: a simple case study. In: IEEE international test conference (ITC), pp 106–109
4. Chakrabarty K (2003) Test access mechanism optimization, test scheduling, and tester data volume reduction for system-on-chip. IEEE Trans Comput 52(12):1619–1632
5. Chou RM, Saluja KK, Agrawal VD (1997) Scheduling tests for VLSI systems under power constraints. IEEE Trans VLSI Syst 5(2):175–185
6. Dahbura A, Uyar M, Yau C (1989) An optimal test sequence for the JTAG/IEEE P1149.1 test access port controller. In: IEEE international test conference (ITC), pp 55–62
7. Huang Y, Cheng W-T, Tsai C-C, Mukherjee N, Samman O, Zaidan Y, Reddy S (2001) Resource allocation and test scheduling for concurrent test of core-based SOC design. In: IEEE asian test symposium (ATS), pp 265–270
8. IEEE 1149.1: JTAG, http://grouper.ieee.org/groups/1149/1/, 1990
9. Iyengar V (2002) Test wrapper and test access mechanism co-optimization for system-on-chip. J Electron Test: Theory Appl 18:213–230
10. Iyengar V, Chakrabarty K, Marinissen EJ (2002) Wrapper/TAM co-optimization, constraint-driven test scheduling, and tester data volume reduction for SOCs. IEEE VLSI Test Symp (VTS) 44.3:685–690
11. Jiang L, Huang L, Qiang X (2009) Test architecture design and optimization for three-dimensional SoCs. In: Design, automation and test in Europe (DATE), pp 220–225
12. Jiang L, Qiang X, Chakrabarty K, Mak TM (2009) Layout-driven test-architecture design and optimization for 3D SoCs under pre-bond test-pin-count constraint. In: International conference on computer-aided design (ICCAD), pp 191–196
13. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
14. Lee H-HS, Chakrabarty K (2009) Test challenges for 3D integrated circuits. In: IEEE design and test of computers, special issue on 3D IC design and test, pp 26–35
15. Lee Y-J, Lim SK (2008) Co-optimization of signal, power, and thermal distribution networks for 3D ICs. In: Electrical design of advanced packaging and systems symposium, pp 163–166
16. Lewis DL, Lee H-HS (2007) A scan-island based design enabling pre-bond testability in die-stacked microprocessors. In: IEEE international test conference (ITC), pp 1–8
17. Marinissen EJ, Verbree J, Konijnenburg M (2010) A structured and scalable test access architecture for TSV-based 3D stacked ICs. In: IEEE VLSI test symposium (VTS), pp 1–6
18. Marinissen EJ, Zorian Y (2009) Testing 3D chips containing through-silicon vias. In: IEEE international test conference (ITC), pp 1–11
19. Muresan V, Wang X, Muresan V, Vladutiu M (2004) Greedy tree growing heuristics on block-test scheduling under power constraints. In: Journal of Electronic Testing: Theory and Applications, pp 61–78
20. Noia B, Goel SK, Chakrabarty K, Marinissen EJ, Verbree J (2010) Test-architecture optimization for TSV-based 3D stacked ICs. In: IEEE European test symposium (ETS), pp 24–29
21. Oakland S (2000) Considerations for implementing IEEE 1149.1 on system-on-a-chip integrated circuits. In: IEEE international test conference (ITC), pp 628–637
22. Parker KP (2003) The boundary-scan handbook. Kluwer Academic Publishers
23. Rosinger P, Al-Hashimi BM, Chakrabarty K (2006) Thermal-safe test scheduling for core-based system-on-chip integrated circuits. IEEE Trans Comput-Aided Des Integr Circ Syst (TCAD) 25(11):2502–2512
24. Samii S, Selkala M, Larsson E, Chakrabarty K, Peng Z (2008) Cycle-accurate test power modeling and its application to soc test architecture design and scheduling. IEEE Trans Comput-Aided Des Integr Circ Syst 27(5):973–977
25. Semiconductor Industry Association (SIA), In: International Technology Roadmap for Semiconductors (ITRS), 2011
26. SenGupta B, Ingelsson U, Larsson E (2012) Scheduling tests for 3D stacked chips under power constraints. J Electron Test: Theory Appl (JETTA) 28(1):121–135
27. Wang L-T, Wu C-W, Wu C-W, Wen X (2006) VLSI test principles and architectures: Design for testability. Academic
28. Whetsel L (1997) An IEEE 1149.1 based test access architecture for ICs with embedded cores. In: IEEE international test conference (ITC), pp 69–78
29. Whetsel L (1994) An approach to accelerate scan testing in IEEE 1149.1 architectures. In: IEEE international test conference (ITC), pp 314 – 322
30. Wu X, Falkenstern P, Xie Y (2007) Scan chain design for three-dimensional integrated circuits (3D ICs). In: International conference on computer design (ICCD), pp 208–214
31. Zhang X, Jiang Y, Ju J (2009) Specific design and optimization of JTAG IP core. In: IEEE circuits and systems international conference on testing and diagnosis (ICTD), pp 1–4
32. Zorian Y (1993) A distributed BIST control scheme for complex VLSI devices. In: IEEE VLSI test symposium (VTS), pp 6–11

Breeta SenGupta received her M.Sc. degree from Indian Institute of Technology, Kharagpur, India. At present she is pursuing her Ph.D. at the Department of Electrical and Information Technology, Lund University, Sweden. Her area of research includes testing of Integrated Circuits, with a focus on test optimization for 3-Dimensionally Integrated Circuits.

Dimitar Nikolov received his Diploma Engineering degree from SS. Cyril and Methodius University, Skopje, Macedonia, in 2008. In 2012, he received the Licenciate Degree from Linköping University, Sweden. In 2015, he received his PhD from Lund University, Sweden. His current research interests include fault tolerance, design and test of digital systems, computer architectures, reconfigurable computing, and digital signal processing.

Urban Ingelsson received the MSc degree from Linköping University, Sweden, and the PhD degree from the University of Southampton, U.K. He has been a post-doc at Linköping University before joining Semcon Sweden AB in 2011. His research interests include testing of digital electronic systems, fault management, test of 3D-stacked ICs, and functional safety.

Erik Larsson is Associate Professor at the Department of Electrical and Information Technology at Lund University since 2012. He received his M.Sc., Tech. Lic and Ph.D from Linköping University in 1994, 1998, 2000, respectively. After a Post Doc at Nara Institute of Science and Technology (NAIST), Japan, he returned to Linköping University as Assistant Professor (2002-2005) and Associate Professor (2006-2012). He did a research visit at NXP Semiconductors, The Netherlands (2008-2010). His current research interests include test planning for manufacturing test, test during operation (in-situ), scan-chain diagnosis, silicon debug and validation, IJTAG/SJTAG, stacked 3D chip test, fault-tolerance for MPSoCs (Multi-Processor System-on-Chip), and property checking in distributed systems (MPSOcS with Network-on- Chip (NoC)). He has more than 130 publications in these areas. He received the best paper award at the European Test Symposium (2016) and Asian Test Symposium (2002). In 2009 he received the Institution of Engineering and Technology (IET) Premium Award. He is Senior member of IEEE.