Efficient Utilization of Test Elevators to Reduce Test Time in 3D-ICs

Sreenivaas S. Muthyala and Nur A. Touba

Computer Engineering Research Center,
The University of Texas, Austin, Texas, USA
sreenivaas@utexas.edu, touba@ece.utexas.edu

Abstract. Three Dimensional Integrated Circuits are an important new paradigm in which different dies are stacked atop one another, and interconnected by Through Silicon Vias (TSVs). Testing 3D-ICs poses additional challenges because of the need to transfer data to the non-bottom layers and the limited number of TSVs available in the 3D-ICs for the data transfer. A novel test compression technique is proposed that introduces the ability to share tester data across layers using daisy-chained decompressors. This improves the encoding of test patterns substantially, thereby reducing the amount of test data required to be stored on the external tester. In addition, an inter-layer serialization technique is proposed, which further reduces the number of TSVs required, using simple hardware to serialize and deserialize the test data. Experimental results are presented demonstrating the efficiency of the technique proposed.

Keywords: 3D-IC, Testing, TSVs, Test Compression, Serialization

1 Introduction

Three-dimensional Integrated Circuits (3D-ICs) are stacked devices with low latency interconnections between adjacent dies, resulting in reduced power consumption in interconnects and higher bandwidth for communication across layers [Hamdioui 11]. 3D-IC designs are suitable for fabricating chips containing heterogeneous components like analog, digital, mixed signal, flash, DRAM components, which is usually the case in System-on-Chip designs. The different layers in a 3D-IC are connected using through-silicon-vias (TSVs) which are vertical interconnects across layers. There are different ways of stacking the dies, wafer-to-wafer (W2W), die-to-wafer (D2W) or die-to-die (D2D), each having its own set of pros and cons, described in [Patti 06].

3D-IC testing is more challenging than 2D-IC testing, since the components of the system are distributed across different layers and, as mentioned earlier, can have non-digital components. In addition, the dies comprising the different layers can be synthesizable designs (soft cores), in which the design-for-test (DFT) architecture can be

customized. The designs can also be layouts (hard cores) for which the DFT architecture is already fixed. It can also contain IP cores, in which it is possible to perform test pattern generation to obtain test patterns that are applied during test. Another challenge in 3D-IC testing is that the input tester channels from the automatic test equipment (ATE), i.e. the external tester, is fed only to the bottom layer in the 3D-IC, and to transfer the test data to the non-bottom layers, TSVs are used, like how the functional data is transferred. These TSVs used for testing purposes are also called test elevators [Marinissen 10]. This further constrains testing, the number of tester channels required for the non-bottom layers should be accommodated within the available number of TSVs, which is a challenge in hierarchical designs with numerous cores. Apart from these, it is also necessary to test the TSVs, since they are prone to defects such as insufficiently filled TSVs, development of micro-voids in the copper filling, and cracking, which is a result of having different coefficient of thermal expansion [Marinissen 09].

There are several stages in testing 3D-ICs; pre-bond testing is performed on each die individually, to ensure defective dies are not used in stacking and then post-bond testing is performed on the final stack, to ensure the 3D-IC as a whole is not defective. In some cases, testing is also done on partial stacks. In order to do pre-bond testing on the non-bottom layers, it is necessary to add probe pads for test purposes. This is because the TSV tips as well as the microbumps are too small to be probed and are very sensitive to scrub marks [Marinissen 09]. These probe pads are a test overhead that take up a lot of space and limit the locations where TSVs can be placed, which puts constraints on the design and floorplan of a die.

One important way to reduce test time is by test compression [Touba 06]. The conventional way of implementing test compression in core-based schemes is for each core to have its own local decompressor. This allows compressed test data to be brought over the Test Access Mechanism (TAM) lines to each decompressor in each core. In a 3D-IC, if a decompressor is shared across multiple cores that are in different layers, larger TAMs would be required, greatly increasing the routing complexity. In addition, this will also increase the number of test elevators required to transfer the uncompressed test data to the non-bottom layers, which is not desirable.

The existing methods for dynamically adjusting the number of free variables sent to each decompressor have the drawback that they increase the amount of routing (i.e. TAM width) required between the tester channels and the core decompressors. This is less suited for 3D-ICs because routing between layers requires additional test elevators (i.e., extra TSVs). This paper proposes a new architecture and methodology for test compression in core-based designs, which is better suited for 3D-ICs. It can be used to either provide greater compression for the same number of test elevators or reduce the number of test elevators while maintaining the same compression compared to existing methods.

This paper proposes a simple, elegant and scalable test compression architecture that reduces the required number of tester channels by sharing tester data amongst the different layers, which in turn, can be accommodated by using very few test elevators. In addition, further reduction in the number of test elevators can be obtained by using

"Inter-layer serialization", a simple technique to send data serially across layers. Preliminary results were presented in [Muthyala 14].

2 Related Work

Testing a core based design typically involves designing wrappers around the cores, providing a test access mechanism, which transfers test data from the pins of the tester to the cores, and developing a test schedule, which decides the sequence in which the cores are tested, the set of cores that are tested concurrently and so on. There are several techniques to design wrappers, TAMs and test schedules, see [Xu 05] for a survey.

In the 3D-IC paradigm, an early work addressing the testability issues of 3D-ICs uses the concept of "scan island" for pre-bond testing [Lewis 09], which uses IEEE standards 1500 and 1149.1 to design wrappers for pre-bond testing. [Wu 07] proposes different optimization strategies for the number of scan chains and the length of TSV based interconnects for 3D-ICs. Heuristic methods for designing core wrappers is proposed in [Noia 09], however it does not involve the reuse of die level TAMs [Noia 10]. [Jiang 09] proposes using heuristics to reduce weighted test cost, while taking into account constraints on the widths of test pins for both pre-bond and post-bond testing. However, the above methods involve die-level optimization of TAMs. [Jiang 09] addresses TAM optimization for post bond testing by placing a constraint on the number of extra probe pads for pre-bond testing when compared to the post bond testing. [Lo 10] proposes reusing the test wrappers of individual cores embedded in the 3D-IC, for test optimization. Recently, the optimization procedures for minimizing test time under a constraint on the number of test elevators has been proposed in [Wu 08] and [Noia 10]. A feature of 3D-IC testing is that the number of probe pads on the non-bottom layers (which are added for test purposes, as explained in Sec. 1) is very limited, which results in limited bandwidth available from the tester for pre-bond testing. [Jiang 12] proposes having separate TAMs for pre-bond testing and post-bond testing. [Lee 13] uses a single TAM with a bandwidth adapter to manage bandwidth for pre-bond and post-bond testing. However, all the above works target test time reduction by optimization of test architectures.

Some recent work has looked at ways to improve encoding efficiency by dynamically adjusting the number of free variables sent to the decompressor in each core on a per test cube basis to try to better match the number of free variables sent to the decompressor with the number of care bits. This can be done by dynamically allocating the number of tester channels used to feed each decompressor [Janicki 12] or by selectively loading decompressors in each clock cycle [Kinsman 10], [Muthyala 13]. [Larsson 08] presents selective encoding of test data for testing SOCs using codewords to represent the care bit profile of the test cubes. The conventional approach is to have independent decompressors for each core such that each of them has their own set of free variables so that the linear equations for each decompressor can be solved independently. This helps to reduce computational complexity. The drawback is that the unused free variables are wasted when the decompressor is reset be-

tween test cubes. This happens in all of the existing methods except [Muthyala 13], which allows limited sharing of free variables between decompressors during a few cycles in such a way that the linear equations are mostly independent and can be solved with only a small increase in computational complexity.

In conventional testing, the scan shift frequency is typically much slower than the functional clock frequency and the maximum ATE (automatic test equipment) clock frequency, due to power dissipation limitations as well as the fact that the test clock may not be buffered and optimized for high speed operation. In [Khoche 02], the ATE shifts in test data *n* times faster than the scan shift. An *n*-bit serial-to-parallel converter is used to take in serial data at the fast ATE shift rate and convert it to *n*-bits in parallel at the slower scan shift frequency. This allows a single ATE channel to fill *n* scan chains in parallel. This idea was combined with a test compression scheme in [Wang 05], where faster ATE channels are sent to a time division demultiplexing (TDDM) circuit which feeds the inputs of a VirtualScan decompressor [Wang 04], and the compacted output response is fed through a time division multiplexing (TDM) circuit to go to the ATE.

This paper proposes using a similar concept for test elevators when transferring test data from one layer to another layer in a 3D stack. The idea is to use a serializer in the layer sending the test data that accepts data in parallel at the scan shift frequency, and generates a serial output at the functional clock frequency which is sent over the test elevator to a deserializer on the receiving layer which converts the serial data coming in at the functional clock frequency into parallel outputs at the scan shift frequency. This approach is described in detail in Sec. 5.

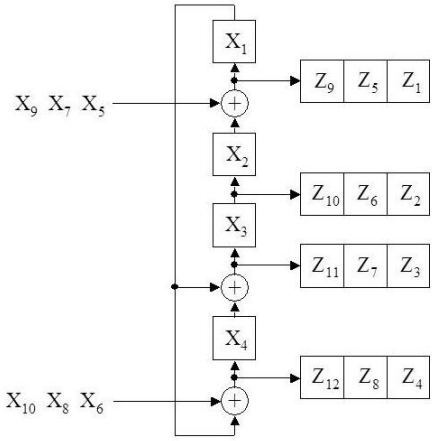
3 Sequential Linear Decompression

Sequential linear decompressors are a major class of decompressors used for test compression, and are inherently attractive while encoding test cubes, i.e. test vectors with unassigned inputs represented as don't cares. The compressed test data, which is stored on the external tester (Automatic Test Equipment, ATE), is obtained using a two-step procedure as follows:

1. The test patterns applied to the scan cells are obtained from automatic test pattern generation (ATPG).
2. Since the decompressor architecture is known, the test patterns can be encoded to determine the compressed tester data.

Fig. 1 shows an example of how the tester data is obtained by encoding using a 4-bit linear feedback shift register (LFSR) as the decompressor. To encode the test patterns, the tester data coming from the external tester are considered *free variables* and the dependence of scan cells on these free variables is obtained using symbolic simulation, which gives the linear equations for the free variable dependence of each scan cell. These equations are solved for each test pattern (*Z* in Fig. 2) by assigning values to the free variables, making them *pivots* (free variables that are not assigned any value are called *non-pivots*). These values of free variables (both pivots and non-

pivots) constitute the tester data is stored on the tester to obtain the test pattern in the scan cells.



$Z_9 = X_1 \oplus X_4 \oplus X_9$	$Z_5 = X_3 \oplus X_7$	$Z_1 = X_2 \oplus X_5$
$Z_{10} = X_1 \oplus X_2 \oplus X_5 \oplus X_6$	$Z_6 = X_1 \oplus X_4$	$Z_2 = X_3$
$Z_{11} = X_2 \oplus X_3 \oplus X_5 \oplus X_7 \oplus X_8$	$Z_7 = X_1 \oplus X_2 \oplus X_5 \oplus X_6$	$Z_3 = X_1 \oplus X_4$
$Z_{12} = X_3 \oplus X_7 \oplus X_{10}$	$Z_8 = X_2 \oplus X_5 \oplus X_8$	$Z_4 = X_1 \oplus X_6$

Fig. 1. Example of Symbolic Simulation of Sequential Linear Decompressor

More details about this procedure can be found in [Könemann 91], [Krishna 01] and [Wang 06], which is used in Mentor Graphics' TestKompress tool [Rajski 04]. An important thing to notice is that the LFSR that is fed by the free variables is reset to zero after decompressing each test cube, to keep the complexity of solving equations within manageable limits. This means that the *non-pivot* free variables, which are not required to have any value to solve the equations, are not used and simply thrown away. This source of inefficiency in compression has been dealt with in several previous works.

The amount of compression achieved with sequential linear decompressors depends on the *encoding efficiency*, defined as the ratio of the number of free variables (bits stored on the tester) versus the number of care bits in the test data. The encoding efficiency achieved using the conventional approach of having fixed size TAMs feeding multiple independent core decompressors is limited by the fact that the TAM bandwidth to each core decompressor needs to be large enough to bring enough free variables to encode the worst-case test cube with the largest number of care bits. To maximize encoding efficiency, a special ATPG procedure is used to generate test cubes in a way that limits the number of care bits in each test cube. This typically comes at the cost of more test cubes being generated. Moreover, there can still be a considerable amount of variance in the number of care bits per test cube (profiles for industrial circuits showing how the percentage of care bits varies can be found in

[Janicki 12]). The inherent drawback of the conventional approach is that the number of free variables brought in for decompressing each test cube is fixed by the worst-case test cube with the most care bits, and so, many free variables are wasted for test cubes with fewer care bits.

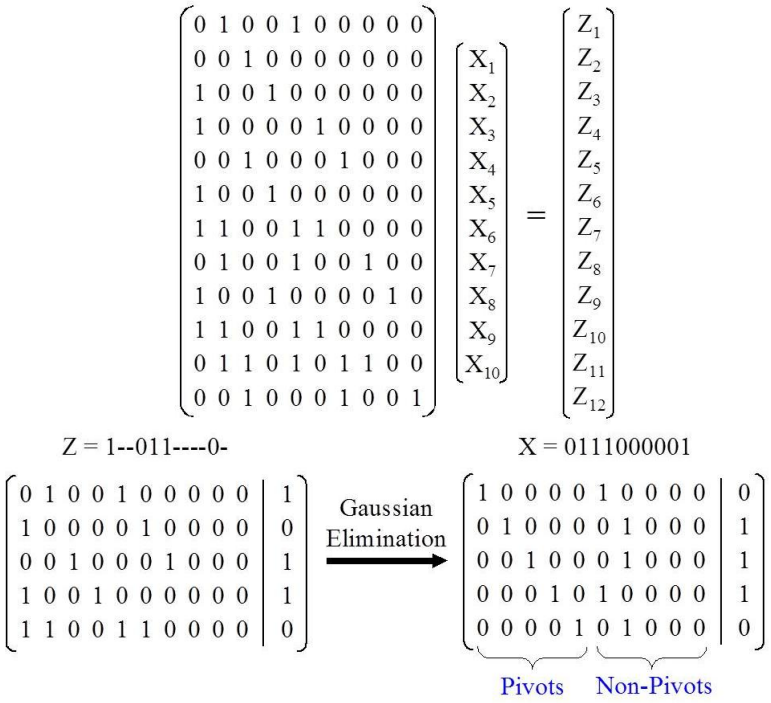


Fig. 2. Gaussian Elimination to Obtain Tester Data for Example in Fig. 1

Several improvements to the encoding procedure for sequential linear decompressors have been proposed, including reusing unused free variables [Muthyala 12]. In addition, a test scheduling procedure for hierarchical SoC based designs was proposed in [Muthyala 13], which enables free variable reuse.

To implement test compression in a core-based 3D-IC, it is necessary to design Test Access Mechanisms (TAMs) for each core. The IEEE 1500 standard, which was introduced to address the challenges in SoC testing, would be a very suitable solution for 3D-IC testing. In addition, the provision of a test wrapper, which standardizes the test interface, further tilts the scales towards the IEEE 1500 standard for 3D-IC testing, since the same interface can be used for pre-bond and post-bond testing. However, the implementation of a test wrapper should also consider the fact that the number of test elevators used for testing should be minimized. There are several ways to design the connection between the test wrappers of each core, one of them, similar to the one shown in Fig. 3, is used in the test architecture proposed in this paper.

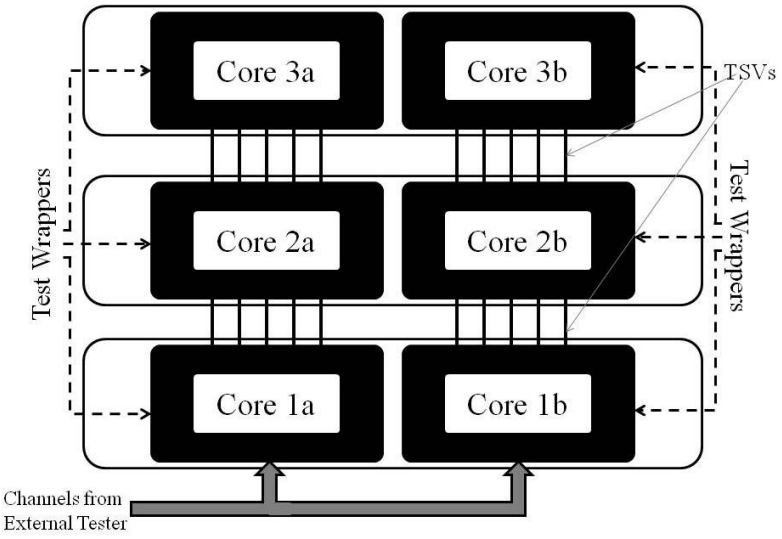


Fig. 3. Test Architecture for a 3D-IC with Wrappers for Each Core

However, it is not necessary to design wrappers in the way shown in Fig. 3; several other ways of designing core wrappers are possible, any design can be selected based on the need. Another application for wrappers is for bypassing cores that cannot be tested in tandem with other cores. For instance, if analog cores are present, the test wrapper for the analog core can be bypassed. These wrappers can be designed using the IEEE 1500 standard for testing SoCs, since the logic between the wrappers, namely the TSVs, are also required to be tested.

4 Proposed Architecture

The conventional approach for test compression in a core-based 3D-IC design, with each layer having several cores operating independently of other cores, is shown in Fig. 4. In this architecture, which uses static allocation of tester channels, the input test data bandwidth from the Automatic Test Equipment (ATE) is distributed to core decompressors using Test Access Mechanisms (TAMs) according to the free variable demand of each core. This distribution is done in a way to minimize the total time required to test the entire 3D-IC with no additional control information. However, the distribution of the tester channels among the decompressors is static and cannot be changed for every test cube. In addition, the tester channels allocated to each core should be large enough in number to ensure sufficient free variables are provided in symbolic simulation to encode the test cube that requires the largest number of free variables (in other words, the most difficult to encode test cube). However, there is a disadvantage if the decompressor is encoding a test cube that requires fewer free variables. Since the tester channels feed the same number of free variables for encoding all the test cubes, many free variables are wasted while encoding “easier-to-encode”

test cubes, which results in increase in tester storage, as even the free variables not assigned any value (non-pivots) have to be stored on the tester. This extra tester storage, which is not used, is necessary to support the test architecture implemented in the 3D-IC.

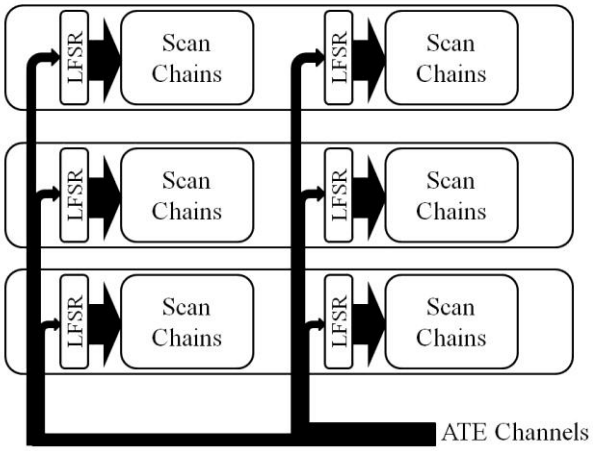


Fig. 4. Test Architecture of 3D-ICs with Static Allocation of Tester Channels

To overcome this disadvantage, the tester channel allocation can be dynamic (i.e. changed for every test cube); however, to do so it is necessary to route the entire ATE bandwidth to all the layers, and control the number of channels allocated by providing additional control signals, which need more test elevators compared to static channel allocation. This presents a significant advantage that, because the number of channels allocated while decompressing every test cube can be changed, the number of wasted free variables from the tester can be reduced. In the context of 3D-ICs, the number of tester channels to each core also translates into the number of TSVs or test elevators required. Thus, dynamic allocation of tester channels requires lot more test elevators than static allocation, which may not be feasible for designs with large number of cores, as the number of TSVs available is limited.

The proposed architecture is shown in Fig. 5; with the decompressors of the cores daisy-chained together. This architecture uses the same number of test elevators as the static channel allocation, while providing an advantage almost equivalent to the dynamic channel allocation. In the architecture shown in Fig. 5, the decompressors of the cores are daisy-chained together, with a tapering bandwidth between successive higher layers of the 3D-IC. In other words, the number of channels between the highest layer and the middle layer is less than the number of channels between the bottom layer and the middle layer, which in turn is less than the input bandwidth to the decompressors in the bottom layer, i.e. total bandwidth from the ATE. Using this architecture, some free variables from the decompressor in the bottom layer is sent to the decompressor in the middle layer and so on. This provides some flexibility in encoding the test cubes. Free variables unused while encoding a core in a layer are

available to encode other cores in the layers above with which they are shared, hence reducing free variable wastage and improving encoding efficiency.

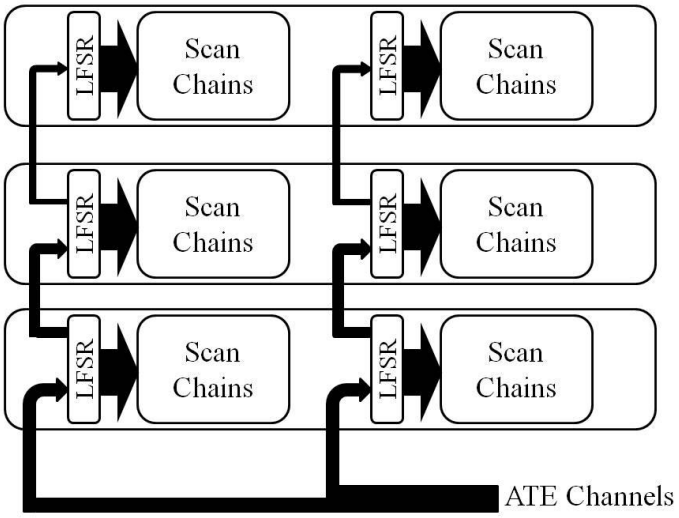


Fig. 5. Proposed Architecture with Daisy-Chained Decompressors

The rationale behind such bandwidth allocation is that the bandwidth feeding each layer should be sufficient to supply enough free variables to encode the test cubes for the core in that layer and the test cubes for the layers above, as the decompressor in each layer is fed via taps from the decompressor of the layer below. Hence, the bandwidth supplied to the bottom layer should be big enough to supply free variables to encode test cubes of the cores in the bottom layer as well as cores in the middle layer and the top layer. Similarly, the bandwidth supplied to the middle layer should be big enough to supply free variables to encode test cubes of the cores in the middle layer and the top layer. In other words, the number of free variables supplied to decompressor in each layer decreases as we move from the bottom layer to the top layer. Hence, a tapering bandwidth is used, as shown in Fig. 5, proportional to the number of free variables required to be supplied to the decompressor in each layer. An example comparing the bandwidth allocation for the two architectures is shown in Fig. 6. It should be noted that the number of test elevators required to implement both the architectures is the same.

The proposed architecture provides flexibility in utilization of free variables; any free variable coming from the tester can be used in any of the layers, provided it is distributed across all the layers.

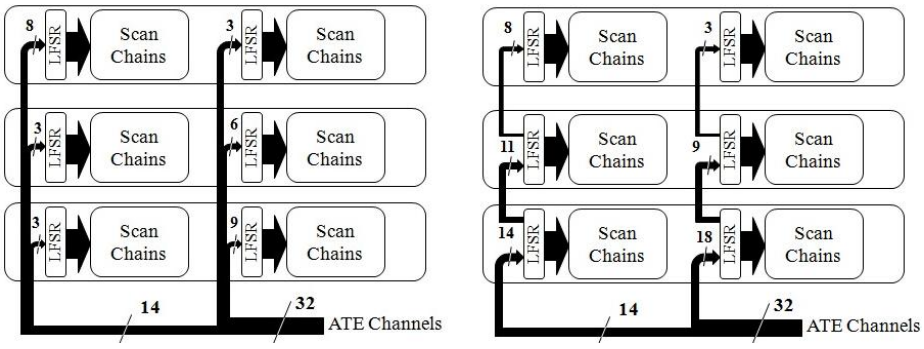


Fig. 6. Example for Comparing Channel Allocation between Conventional (left) and Proposed Architectures (right).

Consider the decompressor in the top layer is encoding an easy-to-encode test cube with few specified bits, which needs less than average free variables, while the test cube being encoded in the middle layer is hard-to-encode with a larger number of specified bits, needing more free variables than average. In the proposed architecture, the free variables which are not required in the top layer can be used in the middle layer in help encode the hard-to-encode test cube. This reduces the number of free variables required to encode the entire test set for the 3D-IC, without additional hardware and control, with the same number of TSVs as used in the conventional architecture shown in Fig. 4. However, to accomplish this, the test cubes for the cores should be reordered and reorganized such that all the cores are not being tested by hard-to-encode test cubes at the same time. In other words, the number of free variables required to encode test cubes that are decompressed together should be reduced as much as possible. This reduces the tester storage and thereby, the test time. This reordering can be accomplished heuristically by a test scheduling algorithm similar to the one explained in [Muthyala 13], with a few modifications to suit the 3D-IC testing paradigm. However, additional control data is not required, unlike [Muthyala 13], unless a subset of cores from those daisy-chained together have to be selected.

Table 1. Example - Care Bit Profile of Test Cubes for the Three Cores of the Example 3D-IC

Test Cube	# Care bits in Test cube		
	Core 1	Core 2	Core 3
1	13	11	12
2	12	11	10
3	10	10	9
4	9	7	8
5	8	6	7
6	7	5	5
7	7	5	4
8	6	4	3

To illustrate the encoding advantage of the proposed approach, consider a small example in which a 3D-IC has three layers with each layer having one core. Let the set of test cubes for the cores have care-bit profiles as shown in Table 1. Consider each care bit in a test cube needs one free variable to encode, as a first order approximation. To encode the entire 3D-IC using the conventional architecture shown in Fig. 1, core 1 would need a minimum of 13 free variables per test cube, core 2 would need 11 free variables and core 3 would need 12 free variables per test cube. Since there are 8 test cubes for each core, a total of (13 + 11 + 12) x 8 = 288 free variables are required to encode the 3D-IC.

Consider the same 3D-IC, now using the proposed architecture. In this case, encoding is done in groups of three test cubes, with one test cube from each core. Optimizing the test cubes in the group for minimizing the total number of care bits in the group, the test cubes are grouped as shown in Table 2.

Table 2. Care Bit Profile of Test Cube Groups for Encoding Using Proposed Architecture

Test Cube Group	# Care bits in Test cube			Total Care Bits in Test Cube Group
	Core 1	Core 2	Core 3	
1	13	4	7	24
2	12	5	8	25
3	10	5	12	27
4	9	6	10	25
5	8	7	9	24
6	7	10	5	22
7	7	11	3	21
8	6	11	4	21

By encoding in groups according the above table, the maximum number of care bits in any group is 27, and hence, to encode the entire set of eight test cube groups, 27 x 8 = 216 free variables are required using the proposed architecture, which is less than the number of free variables required to encode using the conventional architecture shown in Fig. 4. Thus, it is seen that the proposed architecture gives better compression and hence reduces the tester storage requirements and test time.

The drawback in the proposed architecture is that the linear equations for scan cells driven by daisy-chained decompressors cannot be solved independently. In order to make sure the shared free variables are used in only one of the decompressors, the equations for all the scan cells fed by the connected decompressors have to be solved concurrently. Consider one test cube from each layer being encoded. The total number of free variables brought in from the tester has to be sufficiently large enough to encode all three test cubes. Hence, creating pivots for care bits in all the three test cubes involves a lot of computation. The total number of XOR operations required to create pivots is $9n^3$ as compared to $3n^3$ XOR operations required to encode using the conventional approach, where n is the number of free variables. Hence, this method is

reasonable when the additional computation is a reasonable price to pay for the test time reduction achieved.

5 Optimizing Number of Test Elevators by Inter-layer Serialization of Test Data

In this section, an implementation is proposed using a serializer-deserializer structure to further reduce the number of test elevators required to implement the proposed architecture. The scan shift frequency is usually lower than the functional frequency, since the scan clock tree is generally not buffered up for high speeds. Another reason is that during scan shift, a large percentage of flip-flops toggle, and hence a large amount of power is drawn from the power grid of the chip. This causes a voltage drop in the power lines. To avoid these problems, scan shifting is generally considerably slower than functional frequency.

By using the proposed implementation, the difference between the slower scan shift frequency and the faster functional frequency can be exploited to further reduce the number of test elevators. The idea is to use a serializer at the layer sending test data to serialize the test data from the decompressor taps. The test elevators are driven in the sending layer by this serial test data. At the receiving layer, the data from the test elevators are restored in parallel and sent to the decompressor in the same format as sent by the LFSR in the sending layer. (Fig. 7). This is possible because the test elevators are optimized for the functional frequency. However, even if the transfer of data cannot be done at the functional frequency, it can be done at a frequency higher than the scan shift rate, as long as it is a multiple of the scan shift frequency.

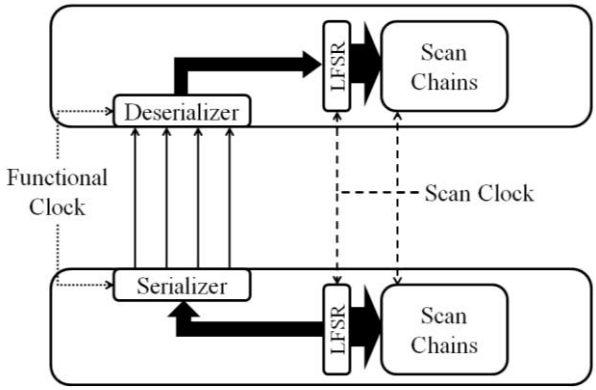


Fig. 7. Proposed Implementation of Inter-layer Serialization of Test Data

If the test elevators can be operated at n times the scan shift frequency, then instead of having m test elevators to transfer data across layers, $\lceil m/n \rceil$ test elevators are required. On the other hand, it is also possible to have the same number of test elevators and increase the effective bandwidth by n times, i.e. m x n bits of data can be shifted in one shift cycle using m test elevators implementing inter-layer serialization, as

compared to *m* bits of data shifted in one shift cycle using *m* test elevators without serialization. Hence, depending on the constraints on the number of test elevators, this architecture can be used at an advantage to either increase the effective bandwidth or reduce the number of test elevators required in the design.

Consider a serializer driven by *m* taps from the LFSR in the sending layer. Let the functional clock be *n* times faster than the scan clock. Therefore, as explained previously, the number of test elevators required would be *m/n*. Let the number of test elevators required be represented as *t*, which, here, is equal to *m/n*. Inter-layer serialization would require an *m* x *t* serializer in the sending layer driving the test elevators between the layers and a *t* x *m* deserializer driving the LFSR in the receiving layer. The simplest way of implementing an *m* x *t* serializer in the sending layer is by using a *m:t* multiplexer controlled by a modulo *m* counter driven by the faster functional clock (Fig. 8).

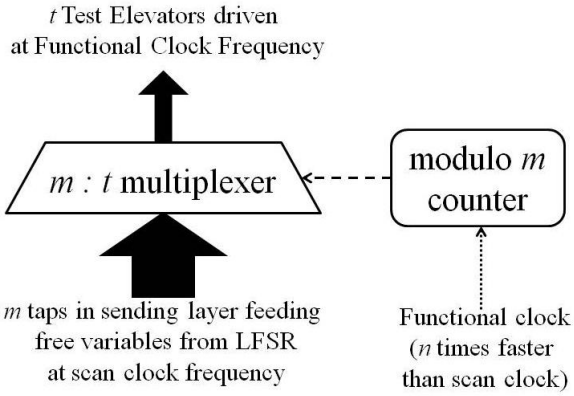


Fig. 8. Serializer Implementation for Functional Clock Operating *n* Times Faster than Scan Clock, Where Number of Test Elevators, *t = m/n*

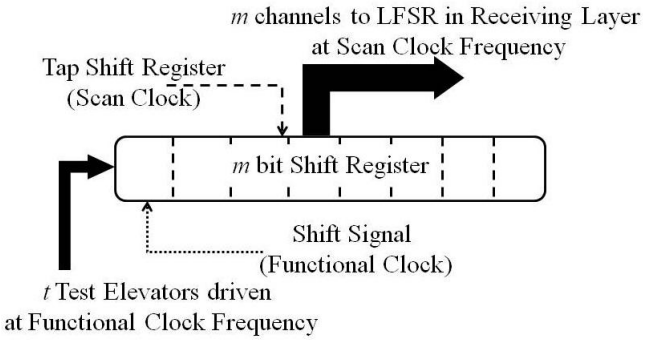


Fig. 9. Deserializer Implementation for Functional Clock Operating n Times Faster Than Scan Clock, Where Number of Test Elevators, *t = m/n*

This ensures that the test data coming in at scan shift frequency from the LFSR is coupled to the test elevators at the faster functional clock frequency. Similarly, the

deserializer can be implemented as an *m* bit shift register driven by the faster functional clock, whereas the data in the shift register is sampled at the slower scan clock rate (Fig. 9).

Alternatively, if the LFSRs can be operated at the functional frequency, or at any other faster frequency, the need for additional serializer and deserializer can be avoided. The LFSRs can just transfer the data at the faster frequency, while the scan chains shift in data at the scan frequency. However, this needs the time domains of the different cores to be synchronized or the frequency has to be reduced in such a way that the receiving LFSR can sample the data from the test elevators without any loss in data or introducing metastability concerns.

As discussed above, inter-layer serialization has a small area overhead; hence, it can be used in cases where the advantage of using it outweighs the additional cost of implementing the required architecture in the design.

6 Experimental Results

Experiments were conducted on six different 3D-IC designs and the results are presented in this section. These 3D-ICs have three layers, with each layer having one core. Three different test compression architectures were experimented using these test chips. In the first test architecture (*arch1*), each core has a 64-bit LFSR, acting as a decompressor. The input tester channels from the ATE are allocated to the decompressors statically as shown in Fig. 4. In this architecture, the output cone of the each decompressor is confined to the layer in which the decompressor is present, i.e. test elevators are required to transfer the compressed test data to the decompressors in the non-bottom layers. However, all scan cells driven by a decompressor are localized to the layer in which the decompressor is present. Hence, it is enough to have sufficient test elevators to transfer the compressed test data, which is generally less in number compared to using test elevators to transfer the uncompressed test data.

In the second architecture (*arch2*), the 64-bit LFSRs that were local to each layer in *arch1*, are interconnected and reconfigured to form a big primitive LFSR with number of flip-flops in this big LFSR being the sum of the number of flip-flops in the *arch1* LFSRs in all the three layers. It should be noted that the LFSR in *arch2* is distributed across the three layers, i.e. sections of LFSR are present in each of the three layers and these sections are interconnected using test elevators in such a way that a primitive LFSR is formed and this 192-bit LFSR drives scan chains in all three layers of 3D-IC. In this case also, the scan chains are confined within the layer, i.e. test elevators are required to transfer compressed test data to the sections of LFSR in the non-bottom layers.In addition, test elevators are also required to interconnect the sections of LFSR that are in different layers. Hence, more test elevators are required in *arch2* compared to *arch1*. Using this architecture, the equations for the scan cells of the three layers are solved together and since the pivots for the test cubes are created in common, the free variables are used more efficiently and results in better compression and increased encoding efficiency.

The third architecture (*arch3*) is the proposed architecture using daisy-chained decompressors shown in Fig. 5. In this case, the decompressors used are similar to the ones used in *arch1*, a local 64-bit LFSR in each core acting as a decompressor for the core, with all the decompressors daisy-chained together. The tester channels are allocated to the decompressors as described in Sec. 4. This method combines the advantages of *arch1* and *arch2* at the cost of increased computational complexity of encoding the test cubes together. By using this architecture, some of the free variables are distributed to the decompressors in the other layers and the encoding of the test cubes of the three layers are done together, thereby the free variables are used more efficiently and results in better compression and increase in encoding efficiency similar to *arch2*. However, in this method, test elevators are required only to transfer the compressed test data to the non-bottom layers, similar to *arch1*, while providing an encoding advantage similar to arch2. In addition, reconfiguration of the local 64-bit LFSRs into a big LFSR for post bond testing of the 3D-IC is also not necessary when using *arch3*.

Experiments were run on six different designs of 3D-ICs, each containing three layers. The test cubes used provided 100% coverage of detectable faults. Static encoding was used to encode the test cubes. The compressed test data, i.e. tester storage required for the three architectures explained above and the number of test elevators (TSVs) required to implement the test architecture is presented in Table 3. As shown in Table 3, there is reduction in the amount of tester data while using *arch2* when compared to *arch1*.

Table 3. Comparison of Tester Data for the Three Architectures

	Test Vectors	Scan Cells	Local Independent Decompressors (*arch1)*		Global Decompressor (*arch2)*		Proposed Daisy-chained Decompressors (*arch3)*		Percentage Reduction in Tester Data
			Tester Data (# of Bits)	TSVs	Tester Data (# of Bits)	TSVs	Tester Data (# of Bits)	TSVs	
A	838	2578	8272	13	6016	21	6016	13	27.27%
B	606	6747	18245	15	11070	21	11070	15	39.33%
C	686	5662	9512	13	6560	21	6560	13	31.03%
D	751	8724	13314	15	9193	21	9193	15	30.95%
E	803	9432	13583	15	10144	21	10144	15	25.32%
F	807	10538	17538	15	12046	21	12046	15	31.31%

As explained earlier, similar benefit is obtained by using *arch3* as well. This is because both *arch2* and *arch3* provide flexible use of free variables across layers and the free variables that are not used in encoding test cube of one layer can be used to encode test cubes of other layers. In addition, by using daisy-chained decompressors (*arch3*), the number of test elevators required is less compared to *arch2*, since the decompressors are local to each layer.

7 Conclusion and Future Work

The proposed daisy-chain test architecture for 3D-ICs implements a free-variable sharing technique, which allows test patterns to be encoded more efficiently thereby reducing the amount of tester data that needs to be stored on the tester. Given the larger transistor count of 3D-ICs, reducing tester data is of a great importance. In addition, this architecture can use inter-layer serialization, where the number of test elevators required is reduced even further, by utilizing the faster functional frequency to transfer test data across layers. Experimental results are presented comparing the tester storage requirements of the conventional architecture and the proposed architecture, and it is shown that the proposed daisy-chain architecture presents a significant reduction in the amount of tester storage required, for the same set of test patterns.

This work opens up avenues of possible future work. The method proposed is an aggressive way of minimizing the number of test elevators at the cost of additional computation in the linear equation solver. One direction for reducing the amount of computation would be to develop intelligent partitioning methods for which cores are tested together that reduce the total amount of free variable sharing, but still retain most of the encoding flexibility where it is most needed. Another direction for further research would be to consider ways to reduce power dissipation during decompression by using the added encoding flexibility to reduce transitions in the decompressed scan vectors.

References

[Hamdioui 11] Hamdioui, S.; Taouil, M.. "Yield Improvement and Test Cost Optimization for 3D Stacked ICs", *Proc. of Asian Test Symposium (ATS)*, pp.480-485, 2011

[Janicki 12] Janicki, J.; Kassab, M.; Mrugalski, G.; Mukherjee, N.; Rajski, J.; Tyszer, J., "EDT Bandwidth Management in SoC Designs", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 31, No. 12, pp. 1894-1907, Dec. 2012.

[Khoche 02] Khoche, A.; Volkerink, E.; Rivoir, J.; Mitra, S, "Test vector compression using EDA-ATE synergies", *Proc. of VLSI Test Symposium*, pp. 97-102, 2002.

[Kinsman 10] Kinsman, A.B.; Nicolici, N., "Time-Multiplexed Compressed Test of SOC Designs", *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 18, No. 8, pp. 1159-1172, Aug. 2010.

[Könemann 91] Könemann, B., "LFSR-Coded Test Patterns for Scan Designs", *Proc. of European Test Conference*, pp. 237-242, 1991.

[Könemann 01] Könemann, B.; Barnhart, C.; Keller, B.; Snethen, T.; Farnsworth, O.; Wheater, D., "A SmartBIST Variant with Guaranteed Encoding," *Proc. of Asian Test Symposium*, pp. 325-330, 2001.

[Krishna 01] Krishna, C.V.; Touba, N.A., "Test Vector Encoding Using Partial LFSR Reseeding", *Proc. of International Test Conference*, pp. 885-893, 2001

[Jiang 09] Jiang. L.; Xu. Q.; Chakrabarty, K.; Mak, T.M., "Layout-driven test-architecture design and optimization for 3D SoCs under pre-bond test-pin-count contraint", *Proc. IEEE International Conference on Computer-Aided Design*, pp. 191-196, 2009.

[Jiang 12] Jiang. L.; Xu. Q.; Chakrabarty, K.; Mak, T.M., "Integrated Test-Architecture Optimization and Thermal-Aware Test Scheduling for 3-D SoCs Under Pre-Bond Test-Pin-Count Constraint", *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 20, No. 9, pp. 1621-1633, Sep. 2012.

[Larsson 08] Larsson, A.; Larsson, E.; Chakrabarty, K; Eles, P.; Zebo Peng, "Test Architecture Optimization and Test Scheduling for SOCs with Core-Level Expansion of Compressed Test Patterns", *Proc. of Design, Automation and Test in Europe*, pp. 188-193, 2008.

[Lee 13] Lee, Y.-W.; Touba, N.A., "Unified 3D test architecture for variable test data bandwidth across pre-bond, partial stack, and post-bond test", *Proc. of Defect and Fault Tolerance Symposium*, pp. 184-189, 2013.

[Lewis 09] Lewis, D. L.; Lee, H.-H. S., "Testing Circuit-Partitioned 3D IC Designs", *Proc. of IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 139-144, May 2009.

[Lo 10] Lo, C.-Y.; Hsing Y.-T.; Denq, L.-M.; Wu, C.-W., "SOC Test Architecture and Method for 3-D ICs", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 29, No. 10, pp. 1645-1649, Oct. 2010.

[Marinissen 09] Marinissen, E.J.; Zorian, Y., "Testing 3D Chips Containing Through-Silicon Vias", *Proc. of International Test Conference*, pp. 1-6, 2009.

[Marinissen 10] Marinissen, E.J.; Verbree, J.; Konijnenburg, M., "A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs", *Proc. of VLSI Test Symposium*, 2010

[Muthyala 12] Muthyala, S.S.; Touba, N.A.; "Improving Test Compression by Retaining Non-Pivot Free Variables in Sequential Linear Decompressors", *Proc. of International Test Conference*, Paper 9.1, 2012

[Muthyala 13] Muthyala, S.S.; Touba, N.A.; "SOC test compression scheme using sequential linear decompressors with retained free variables", *Proc. of VLSI Test Symposium*, 2013.

[Muthyala 14] Muthyala, S.S.; Touba, N.A.; "Reducing Test Time for 3D-ICs by Improved Utilization of Test Elevators", *Proc. of International Conference on Very Large Scale Integration (VLSI-SoC)*, 2014

[Noia 09] Noia, B.; Chakrabarty, K.; "Test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs", *International Conference on Computer Design*, pp. 70-77, 2009.

[Noia 10] Noia, B.; Goel, S.K.; Chakrabarty, K.; Marinissen, E.J.; Verbree, J., "Test-architecture optimization for TSV-based 3D stacked ICs", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 11, pp. 1705-1718, Nov. 2011.

[Patti 06] Patti, R.S.; "Three Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proc. of the IEEE*, Vol. 94, No. 6, pp. 1214-1224, June 2006.

[Rajski 04] Rajski, J.; Tyszer, J.; Kassab, M.; Mukherjee, N., "Embedded Deterministic Test", *IEEE Trans. on Computer-Aided Design*, Vol. 23, Issue 5, pp. 1306-1320, May 2004.

[Touba 06] Touba, N.A., "Survey of Test Vector Compression Techniques", *IEEE Design & Test Magazine*, Vol. 23, Issue 4, pp. 294-303, Jul. 2006.

[Wang 04] Wang, L.-T.; Wen, X.; Furukawa, H.; Hsu, Fei-Sheng; Lin, Shyh-Horng; Tsai, Sen-Wei; Abdel-Hafez, K.S.; Wu, S., "VirtualScan: a new compressed scan technology for test cost reduction", *Proc. of International Test Conference*, pp. 916-925, 2004.

[Wang 05] Wang, L.-T.; Abdel-Hafez, K.S.; Wen, X.; Sheu, B.; Wu, Shianling; Lin, Shyh-Horng; Chang, Ming-Tung, "UltraScan: using time-division demultiplexing/multiplexing (TDDM/TDM) with VirtualScan for test cost reduction", *Proc. of International Test Conference*, pp. 946-953, 2005.

[Wang 06] Wang, L.-T.; Wu, C.-W.; Wen, X., *VLSI Test Principles and Architectures: Design for Testability*, Morgan Kaufmann, 2006.

[Wu 07] Wu. X; Falkenstern, P; Xie, Y., "Scan Chain Design for Three-dimensional Integrated Circuits (3D ICs)", *Proc. International Conference on Computer Design (ICCD)*, pp. 212-218, Oct. 2008.
[Wu 08] Wu, X.; Chen, Y.; Chakrabarty, K.; Xie, Y., "Test-access mechanism optimization for core-based three-dimensional SOCs", *Proc. of International Conference on Computer Design*, pp. 212-218, 2008.
[Xu 05] Xu, Q.; Nicolici, N., "Resource-Constrained System-on-a-Chip Test: A Survey", *IEE Proc. Computers & Digital Techniques*, Vol. 152, Issue 1, pp. 67-81, Jan. 2005