INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

TSV based 3D-SIC Testing

Sudeep Ghosh
sudeep.ghosh@gnit.ac.in
Gurunanak Institute of Technology, Kolkata, West Bengal

Mandira Banik
mandira.banik@gnit.ac.in
Gurunanak Institute of Technology, Kolkata, West Bengal

Tridib Chakraborty
tridib.chakraborty@gnit.ac.in
Gurunanak Institute of Technology, Kolkata, West Bengal

Niraj Pandey
nirajpandey034@gmail.com
Gurunanak Institute of Technology, Kolkata, West Bengal

ABSTRACT

Three Dimensional Integrated Circuits (3D IC) have been getting attention among researchers and IC designers as an emerging technology to help overcome the interconnect delay and power limitations. Tight integration of multiple silicon tiers using vertical 3D vias as interconnect offers a better functioning of 3D-ICs. In this work, we focus on 3D-SICs implemented using Through- Silicon Via (TSV) vertical interconnects. Through strategic modification of the architectures to take advantage of 3D, significant improvement in the functioning can be achieved.

Keywords— *3D IC, TSV, SICs*

1. INTRODUCTION

The semiconductor industry is pushing relentlessly for high-performance and low-power chips. Recent advances in semiconductor manufacturing technology have enabled the creation of complete systems with direct stacking and bonding of die-on-die. These system chips are commonly referred to as Three-Dimensional (3D) stacked ICs (SICs). In this work, we focus on 3D-SICs implemented using Through-Silicon Via (TSV) vertical interconnects. Using this technology, 3D-SICs are created by attaching multiple device layers to each other through wafer or die stacking, and connecting metal layers between dies using vertical TSVs. Compared to traditional two-dimensional SOCs, 3D-SICs provide greater design flexibility, higher on-chip data bandwidth, reduction in average interconnect length, and alleviation of problems associated with long global interconnects.

Testing core-based dies in 3D-SICs brings forward new challenges. In order to test the dies and associated cores, a Test Access Mechanism (TAM) must be included on the dies to transport test data to the cores, and a 3D TAM is needed to transfer test data to the dies from the stack input/output pins. TAM design in 3D-SICs involves additional challenges compared to TAM design for 2D SOCs. In a 3D-SIC, a test architecture must be able to support testing of individual dies as well as testing of partial and complete stacks. Furthermore, test architecture optimization must not only minimize the test time (test length), but it also needs to minimize the number of TSVs used to route the 3D TAM; as each TSV has area costs associated with it and is a potential source of defects in a 3D-SIC.

2. PREVIOUS WORK

There has been quite a number of experiments being conducted in the field of 3D IC. The results that have been obtained is a result of studying these various works and hence an improvement over the former.

Testing of 3D Stacked ICs (SICs) is becoming increasingly important in the semiconductor industry. In this paper, the problem of test architecture optimization for 3D stacked ICs implemented using Through-Silicon Vias (TSVs) technology. The paper considers 3D-SICs with both fixed, given and yet-to-be-designed test architectures on each die and shows that both corresponding problem variants are NP-hard [1].

Advancement of VLSI technology helps the semiconductor industry to manufacture Through-Silicon-Via (TSV) based 3D Stacked ICs (SICs). During 3D assembly, multiple partial stack tests are necessary. In the paper, test architecture optimization for 3D stacked ICs is implemented with hard dies. Two different test sets derive optimal solutions to minimize all test times when complete stack and multiple partial stacks, need to be tested. Results are performed for two handcrafted 3D SICs comprising of various SOCs from ITC'02 SOC test benchmarks. In this work, the test architecture optimization for 3D SIC where the die level test architecture is fixed and each die consists of one SOC. The decrease in total test length with the increasing number of TSVs is more than the increase in the number of test TSVs. Furthermore, there is a presentation of test schedules and corresponding test lengths for every multiple insertion [3].

The paper has addressed the test infrastructure design for TSV based 3D stacked IC (3D SIC). Each of the dies consists of one or more hard SOCs. The main objective of this work is to design the test architecture for the 3D SIC so that overall test time can be optimized. To prove the efficiency of the proposed algorithm it has considered a 3D stacked IC (SIC) using 5 standard SOCs. Obtained test results show that the proposed solution can achieve up to 59 % reduction in test time compared to the baseline method of sequentially testing all the dies in the stack. It also shows that increasing the number of Test Access Mechanism (TAM) and Through-Silicon Vias.

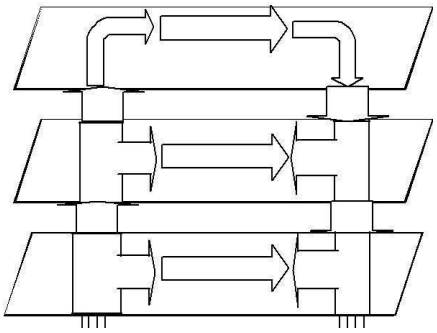
(TSVs) help in the reduction of test time but the increase in the number of TAM is unnecessary after a certain limit. In this work, it has been assumed that the different dies in different layers may consist of two SOCs as opposed to previous work, where each die consists of single SOC [2].

3. OUR WORK

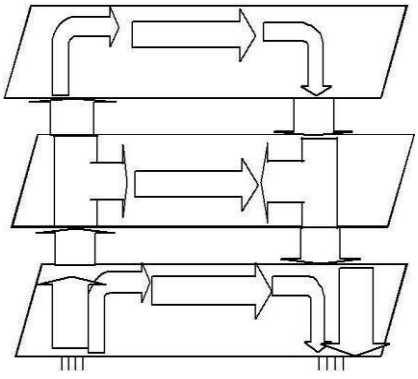
3.1 Problem definition

The lowest die is connected to chip I/O pins. To test the non-bottom dies in the stack, test data must enter through TSVs. To test other dies in the stack, the TAM must be extended to all dies in the stack. "Through-Silicon-Vias" is included on each die except for highest die. An example of scheduling architecture for 3D stacked IC is given with serial test architecture figure (a) and parallel test architecture figure (b).

The problem that has been considered in this paper is as follows: Given a stack with a set M of dies, the total number of TSVs (TSVmax). For each die m Є M, die's number corresponds to its tier. We are given the number of TSVs on each die Tm (TSVm ≤ TSVmax), the associated test time (tm) and a maximum number of TSVs (TSVmax) that can be used for TAM design between dying m-1 and m (m>1). The goal is to determine an optimal TAM design and test schedule for each stage of stacking such that the total test time is minimized and the number of TSVs used per die does not exceed TSVmax.



(a)



(b)

Fig. 1: (a) Serial architecture and (b) parallel architecture of 3D SIC

3.2 Proposed algorithm

In our work, we have assumed that all the SOCs are hard. The principal idea of the proposed algorithm given below is to enable the parallel testing of SOCs as much as possible. For a given test insertion, DN is the set of dies to be tested. First, we have sorted all the DN dies according to their test cycles tried to test all dies of DN in a parallel fashion. It is possible when Σ m for DN dies is less than or equal to TSVmax. Then we can assign a schedule of dies as parallel and test length of this insertion is the maximum test length among DN dies. If it is not possible to test all the DN dies in parallel, we then find all the next set of dies from DN dies such that ΣTSVm ≤ TSVmax. Then when one layer is satisfied we create a new layer and insert the next die there to be run in serial with the other layer. Then assign corresponding dies as a parallel schedule and the test length for this schedule is the maximum test length of DN dies. The remaining dies of DN are scheduled in a similar fashion. So, the test length of the test insertion is found by adding the test length of all the combination. Final stack test time is found by adding the test length of all the insertions.

Dies DN
Total Test Cycle T TSV of Die DTSV Test Cycle of Die DTC

Maximum TSV denoted as TSV_MAX BEGIN

SORT DNin ASCENDING ORDER on the basis of Test Cycle (DTC)

tempTSV = 0, P = 0

WHILE (DNnot empty AND tempTSV + DN[i]TSV≤ TSV_MAX)

tempTSV = tempTSV + DN[i]TSV ADD DN[i] to first layer
i = i + 1

END WHILE

T = getMaxCycle(first layer) Add first layer to 3D Stack
WHILE (DN not empty) Create a new layer L

tempTSV = 0

WHILE (DN not empty AND tempTSV + DN[i]TSV≤ TSV_MAX)

tempTSV = tempTSV + DN[i]TSV ADD DN[i] to L

T = getMaxCycle(L) + Σprevious layers test cycle
i = i + 1

END WHILE
ADD L to 3D Stack

END WHILE END
Function getMaxCycle(Layer L)
maxCycle = -1;
FOR each Die d in L

IF (DTC>= maxCycle)
maxCycle = DTC
END IF

END FOR
RETURN maxCycle
END Function

3.3 Experiment

We execute this algorithm using Java programming language. Now we explain the proposed algorithm with an example as shown above. Consider a SIC with five dies, each die has TSV from bottom to top 40, 30, 25, 20, 10 and associated test length for individual dies are 500, 400, 300, 200 and 100 cycles. So, tier numbers are first sorted on the basis of associated test cycles and then placed on the stack. So TSV 10 are placed first then 20, 30, 40 and 50 and their test lengths are 100,200,300,400,500 respectively. Let TSVmax is considered as 70. When the second die is inserted on the first die then insert as many dies as possible in parallel until ΣTSVm<= ΣTSVmaxbecause ΣTSVm for dies 1 and 2 and 3 are55 and TSVs requirement within layer 1 is less than TSVmax. So the test length is 300 cycles. Then fourth die is inserted on dying 1-2-3-4 and the total number of TSV of die 1, die2 die 3 and die 4 is greater than TSVmax. After the insertion of die 4 whose TSV is 30, ΣTSVm> ΣTSVmaxhence a new layer is created, layer

So they are tested in serial with the parallel execution of dies 1-2-3. The test length is (300+400) =700 cycles and test schedule is 1||2||3, 4. Similarly, after the insertion of fifth die on die 1-2-3-4, then there are again two possibilities, either add die 5 on die 4 in parallel in layer 2 and ΣTSVm=70.Hence ΣTSVm= ΣTSVmaxand now the test schedule becomes 1||2|3,4||5 and the test cycle now becomes (300+500) = 800 test cycles. Finally, after all the completion of insertion of dies, the total testing time becomes (300+700+800) test cycles = 1800 test cycles.

3.4 Experiment result

Experimental results are simulated for one hand crafted 3D SIC. Using several SOCs from the SOC test benchmarks as dies inside SICs. A number of layers for the SICs are five. For simulation, the algorithm is implemented in Java language and run on (Intel® Core™ i5-4440 CPU @ 3.10Ghz, 8.00 GB RAM, x64-based processor) in Windows OS. In SICs the dies are arranged in the ascending order of test cycle.

Table 1: SICs the dies are arranged in the ascending order

Die Number	01	02	03	04	05
TSV	30	25	25	20	15
Test Cycle	2608870	2743317	1333098	700665	106391

For a range of TSVmax and a range of test cycle, we present the result of one SIC using our proposed algorithm

Table 2: Range of TSVmax and a range of test cycle

TSVmax	Total Test Cycles
49	14763011
50	14763011
69	9351481
70	9351981

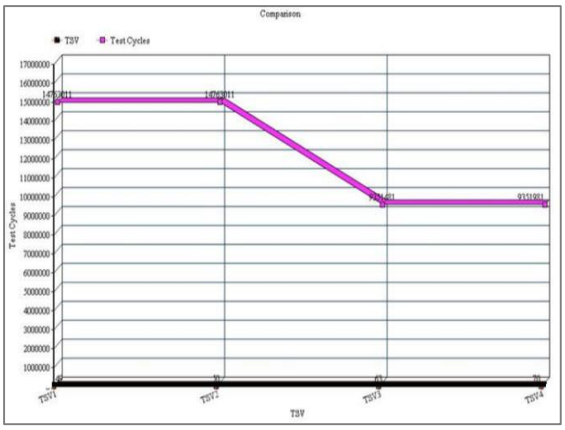


Fig. 2: Variation in total test cycles

The graph shows the variation in total test cycles with an increase in the number of TSVmax for a single SIC. It can be observed thatTSVmax determine which dies should be tested in parallel.

In relation to the previously proposed papers on the aforesaid topic, we have sketched a comparative graph relating the previous result with our experimental result. The table below is a summary of the comparison and shows the optimum result.

Table 3: Comparison with other work

SIC1	SIC2	TSVmax	Experiment al Result
25408042	12901854	49	14763011
21515139	11042714	50	14763011
14972562	9554160	69	9351481
14972562	8959886	70	9351981

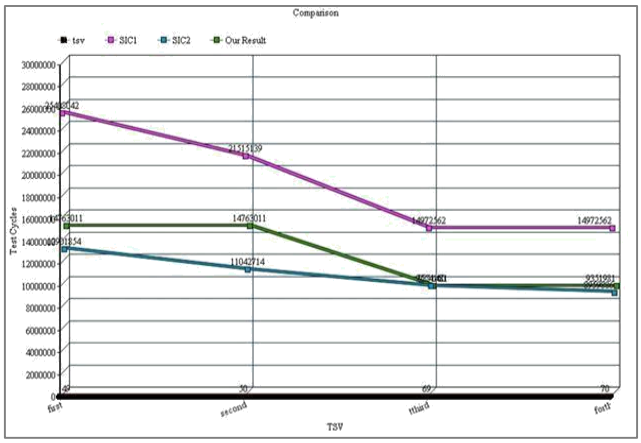


Fig. 3: Comparison with another result

The above line graph shows a comparison between our work and previous works. The green line shows the value of our work. It is evident from the graph that the total test cycles required for testing using our proposed algorithm is less than the previously required test cycles. Our result is way better compared to SIC1 using the sorted method of test cycles, but in case of SIC2 initially, their experiment yields better results but as the number of TSVs increase our values yield better results due to the decrease in complexity of the algorithm. Thus we can state that our result is better than the previous work which is indicated by blue and purple lines respectively.

4. CONCLUSION

This work presented an optimization method that provides test schedules under the constraints of the maximum number of TSVs and also minimizes test time for the final stack test. Results have been presented in comparison with previous works which shows a significant improvement in the testing time as well as reducing the general complexity of the overall setup since it attributed to reducing the number of combination of dies needed to be taken by sorting up the overall TSVs and test cycles. The comparison charts and graphs are an indication of our effectiveness. Hence it can be said with the esteemed belief that this paper has the potential to be taken up for future work to set up the standard for post bond testing of stacked 3D ICs.

5. REFERENCES

[1] Noia, B., Dept. Electr. and Comput. Eng., Duke Univ., Durham, NC, USA Goel, S.K.; Chakrabarty, K.; Marinissen, E.J.; Verbree, J. "Test architecture optimization for TSV based 3D stacked ICs". Published in: Test Symposium (ETS), 2010 15th IEEE European.

[2] Roy, S.K. Dept. Inf. Technol., Bengal Eng. and Sci. Univ., Shibpur, India Giri, C.; Chakraborty, A.; Mukherjee, S.; Das,

D.K.; Rahaman, H. "Optimizing Test Architecture for TSV Based 3D Stacked ICs Using

[3] Hard SOCs".Published in: Electronic System Design (ISED), 2011 International Symposium.

[4] Roy, S.K. Dept. of Inf. Technol., Bengal Eng. and Sci. Univ., Howrah, India Roy, D.; Giri, C.; Rahman, H."Testing 3D stacked ICs for post - bond partial/complete stack". Published in: Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium.

[5] Hsien-Hsin S. Lee and Krishnendu Chakrabarty, "Test Challenges for 3D integrated circuits", IEEE Design and Test of Computers, Special issue on 3D IC Design and Test, vol. 26, no. 5, pp. 26–35, Sep/Oct 2009.

