Faulty TSVs Identification in 3D IC Using Pre-bond Testing

Dilip Kumar Maity[1(✉)], Surajit Kumar Roy[2], and Chandan Giri[2]

[1] Department of Computer Science and Engineering, Academy of Technology,
Hooghly 712121, India
dilip.maity@aot.edu.in

[2] Department of Information Technology, IIEST, Shibpur, Howrah 711103, India
suraroy@gmail.com, chandangiri@gmail.com

Abstract. Through-silicon via (TSV) based three-dimensional integrated circuit (3D IC) is gaining remarkable attention in semiconductor industry. The design of 3D IC goes through a complex manufacturing process and testing of TSVs is a critical issue to the researchers. This paper presents an efficient solution for pre-bond TSV testing. The proposed method generates the sequence of test sessions for identifying defective TSVs in a TSV network in reduced test time. Simulation results show the effectiveness of proposed method in terms of test time reduction than the prior works.

Keywords: 3D IC · Pre-bond testing · TSV test

1 Introduction

The idea of three dimensional integrated circuit (3D IC) is alleviated to overcome the scaling bottleneck of 2D IC. 3D ICs are manufactured by stacking multiple dies vertically and interconnecting the dies using through-silicon vias (TSVs). This promises to achieve performance improvements in terms of reduced interconnection length, reduced power consumption and smaller footprint than conventional 2D IC [1]. But there are several challenges related to testing of 3D IC [2].

TSVs are considered as the path for transporting signal to the upper layer dies of 3D IC. Several types of defects like pinhole or microvoid may arise during manufacturing of 3D ICs and these faults are identified by pre-bond testing [3]. The resistance of a TSV increases due to microvoid whereas the capacitance between TSV and the substrate increases due to pinhole. Capacitance measurement is used for pinhole fault, but resistance measurement is required for microvoid fault [4]. Faults evolved due to alignment, bonding or stress can be identified by post-bond testing. A single faulty TSV can paralyze the whole chip. So, testing of TSVs is necessary for proper functionality of the chip. In this paper, we have considered only pre-bond TSV testing.

It is a challenge to test the TSVs before bonding of different dies due to the small pitch of TSV. A pre-bond TSV probing method is presented in [4] where multiple TSVs are shorted together with a probe needle to form a TSV network. The probing method is modeled by adding an active driver in the probe needle and forming a charge sharing

B. K. Kaushik et al. (Eds.): VDAT 2017, CCIS 711, pp. 805–812, 2017.
https://doi.org/10.1007/978-981-10-7470-7_75

circuit between single (multiple) TSV(s) and the probe needle. Capacitor charging time of the charge-sharing circuit is considered as the test time of TSV for resistance measurement of TSV. The charging time with respect to the different parallel active TSVs is detailed in [4].

Table 1. Capacitor charging time of parallel TSV test [8]

Number of TSVs tested in parallel (q)	Charging time t(q) (μs)
1	0.80
2	0.53
3	0.42
4	0.38

Recently several research works are being undertaken on pre-bond TSV testing. For instance, pre-bond TSV test method is presented in [5] where multiple TSVs can be tested simultaneously. An integer linear programming (ILP) based method to detect faulty TSVs is proposed in [6]. The authors present heuristic methods to identify defective TSVs in reduce the test time in [7, 8]. A session based pre-bond TSV test method is presented in [9]. In this paper we have proposed a fast heuristic method that identifies the defective TSVs efficiently in terms of test set generation and significantly reduces the overall test time.

The rest of this paper is organized as follows: Sect. 2 describes the problem formulation of the proposed methodology. Proposed algorithm for identifying the defective TSVs uniquely is presented in Sect. 3. Proposed methodology is explained using an illustrative example in Sect. 4. Section 5 presents the experimental results and finally Sect. 6 concludes the paper.

2 Problem Formulation

Pre-bond TSV testing helps to identify defective dies early i.e. before bonding. Objective of TSV probing is to identify the defective TSVs within a TSV network. The test time can be reduced if the TSVs are tested in parallel. The proposed pre-bond TSV test method generates sequence of test sessions to detect faulty TSVs in reduced test time. A test session is formed by TSVs that are tested simultaneously and number of TSVs within a session indicates the session size (q). The difference in capacitor charging time between faulty and non-faulty TSVs decreases when size of the session increases which affect the resolution [4]. So the size of the session cannot be increased beyond a certain limit. Resolution constraint (number of probe pins) p indicates the upper bound of session size. Now, formally the problem can be stated as follows: *Given n identical TSVs within a n-TSV network, p number of probe pins and the test time t(q), $1 \leq q \leq p$, for the test session containing q number of TSVs, determine the sequence of test sessions to identify faulty TSVs of the n-TSV network such that the test time is minimized as much as possible.*

3 Proposed Methodology

We have proposed a heuristic based algorithm to solve the above mentioned problem.

The proposed algorithm uses the concept of parallel testing. Parallel TSV testing reduces the test time compared to the sequential TSV testing. Each TSV network has limited number of redundant TSVs (*r*) to repair the network. Finding $r + 1$ number of defective TSVs implies that the network is not repairable; so further test is useless. The proposed algorithm can identify *m* number of defective TSVs, where $0 \leq m \leq r + 1$. Also the test process can be done without knowing any prior information about the number of faults or number of redundant TSVs. The algorithm starts with the procedure *Test_session_generation(),* which generates a session by picking *p* number of TSVs from the set of n untested TSVs. Each generated session is tested and the corresponding test time is recorded. If a session is identified as fault-free then all the TSVs of the session are considered as good TSV. But if the session is faulty, then the procedure *Defective_TSV_identification()* is invoked to find the exact position of the first occurrence of the defective TSV within the current session.

Algorithm 1. Test_session_generation(n, p)

```
Input: number of TSVs (n), number of test pins (p).
Output: Set of test sessions, and total test time.
if count=r+1or n=0 then return;
if n>0 then
    Create a session s by taking all n (if n<p) or p(if n≥p) number of TSVs;
    Modify the session s;                          //by padding non-faulty tested TSVs
    total_test_time:=total_test_time+t(length(s));  //test time accumulation
    if session s is tested as being faulty then
        j:= Defective_TSV_identification(length(s), p);
        n:=n-length(s)+(j+1);  count:=count+1;       //defective TSV is identified at location j
        Test_session_generation (n, p);
    else                                             // the session s is fault free
        n= n-length(s);  Test_session_generation(n, p);
end.
```

Algorithm 2. Defective_TSV_identification(s, p)

```
Input: The faulty session (s), number of probe pins (p).
Output: Position of defective TSV.
if length(s)=1 then return the position of  the defective TSV;    //a defective TSV is identified
else
   Bi-partition the session s into two new sessions s1 and s2;
   Modify the session s1 and s2;                              //by padding non-faulty tested TSVs
   total_test_time:=total_test_time+t(length($s_1$));          //test time accumulation
   if session s1 is tested as being faulty then Defective_TSV_identification($s_1$,p);
   else  Defective_TSV_identification($s_2$,p);      // session $s_1$ is fault free  so s2 must be faulty
end.
```

The procedure *Defective_TSV_identification()* divides a faulty session of length *p* into two smaller sessions (as left half and right half). First the left half is tested. If the left half is found to be faulty, subsequent testing is done on this half itself. On the other hand, if the left half is found as fault free, the right half is bi-partitioned and tested. In this fashion a faulty session is bi-partitioned until the length of session is one. Finally the position of the first defective TSV is identified by the algorithm. When the left half is identified as faulty, we have no concrete information about the right half. Therefore, this right half is considered as untested TSVs. During the test process, non-faulty tested TSVs are padded with the smaller sessions to utilize the maximum number of probe pins for reducing the test time.

4 Illustrative Eample

This section elaborates the proposed method with an example. Consider a problem instance where number of TSVs (n) is16 and number of test pins (p) is 4. TSVs are represented with the numbers 1, 2…16 and initially all are untested. According to the proposed method, procedure *Test_session_generation()* starts with the formation of a session {1, 2, 3, 4} of length 4 and they can be tested simultaneously. Now the session is tested to check whether there is any faulty TSV or not. If there is no fault, then the next session will be formed from the remaining untested TSVs. Assume that the defective TSVs are 1, 8 and 12. So the session {1, 2, 3, 4} is identified as faulty. Hence, *Defective_TSV_identification()* procedure is invoked to detect the position of faulty TSV. The procedure *Defective_TSV_identification()* starts with bi-partitioning the session and generates two new sets {1, 2}, and {3, 4}. The session {1, 2} is tested and detected as faulty. So it is decomposed as {1} and {2}. After testing {1}, it is detected as defective. As the sets {2} and {3,4} are not tested, we cannot make any decision about the TSVs 2, 3,4 that they are faulty or not i.e. they are considered as untested. Now the next session {2, 3, 4, 5} is created from the remaining untested TSVs (2 to 16) and is identified as fault-free.

Similarly, session {6, 7, 8, 9} is identified as faulty and decomposed as {6, 7} and {8, 9}. Now non-faulty tested TSVs can be included randomly with these sets for reducing test time further and the test session {6, 7, 2, 3} is formed. The session {6, 7, 2, 3} is detected as non-faulty and TSVs 6 and 7 are non-faulty. So the fault is within the set {8, 9} which is to be bi-partitioned next. Similarly the session {8, 4, 5, 2} is tested and identified as faulty. Hence, TSV 8 is defective as other TSVs of this session are non-faulty. In a similar fashion, remaining untested TSVs are tested and all the faulty TSVs are detected. At the end of test process following test sessions are generated: {1,2,3,4}, {1,2}, {1}, {2,3,4,5}, {6,7,8,9}, {6,7,2,3}, {8,4,5,2}, {9,10,11,12}, {9,10,3,6}, {11,7,4,5}, {13,14,15,16}.

Here, the underlined numbers indicate that padding is done using these TSVs. The total test time is calculated using Table 1 as $9 \times t(4) + 1 \times t(2) + 1 \times t(1) = 9 \times 0.38 + 0.53 + 0.8 = 4.75$ micro second.

5 Experimental Results

The proposed algorithm is coded and compiled in *gcc* compiler and executed on a Intel Core i5 processor with 3 GB RAM. Simulation results are presented for different TSV networks with varying number of probe pins. HSPICE simulations are considered to find the test time for different test sessions. The resistance and capacitance of each TSV are considered as 1Ω and 20 fF respectively as in [4].

Figure 1 shows the variation of reduction in test time with the number of faulty TSVs identified for 20-TSV network. It is seen from the figures that for a given value of test pins the reduction in test time decreases with increasing value of number of faulty TSVs. 13 faults can be identified with reduced test time for 20-TSV network. The reduction of test time will be large for greater number of test pins. Because the number of test sessions will be less for large value of p. There is 15% reduction of test time than serial testing for 20-TSV network such that 12 faulty TSVs can be detected uniquely. From the Fig. 1 it is also observed that, for small m, the differences of reduction in test time with p = 2, p = 3 and p = 4 are significant, but for large value of m, the test time reduction is quite small. This observation also indicates that for large m, sequential testing is better.

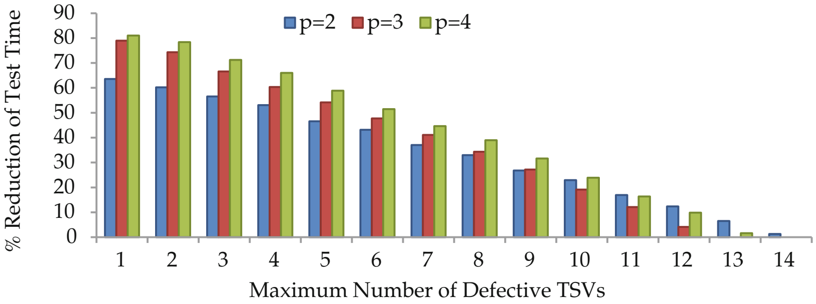

Fig. 1. Percentage reduction of test time for 20-TSV network

In Fig. 2, for 20 TSV networks and test pin = 3, our experimental result is compared with [6, 8]. It is seen that the proposed methodology can identify four 4 with reduced test time of 60% & above and 8 faults with reduced test time of 35% & above. Interestingly to detect seven faults our proposed heuristic approach takes slightly more time than [8]. However, this single anomaly can be overlooked when compared with the reduction of time achieved in all other cases. So it can be concluded that the proposed heuristic approach provides an efficient solution for identifying defective TSVs for 3D ICs for larger TSV network.

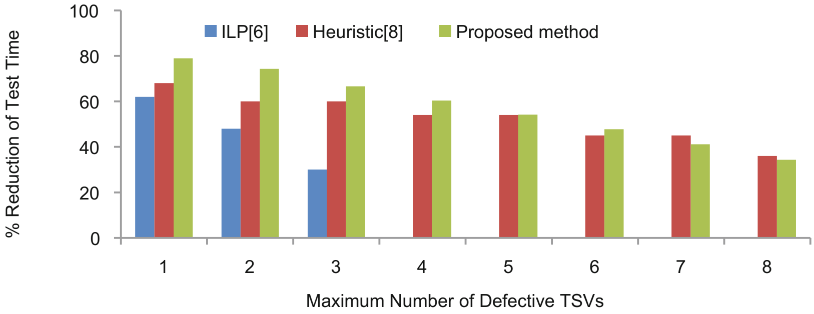

Fig. 2. Comparison in test time for 20-TSV network for p = 3

Table 2 shows the comparisons between the dynamically optimized test in [9] and the proposed method for various TSV networks. From Table 2, the following observations can be made. First, it is seen from the table that obtained results are better than [9] in almost all cases. Second, as expected, number tested session increases as number of fault increases. Third, as m increases the average percentage reduction decreases. This is expected as large number of sessions are tested to identify more defective TSVs within a TSV network.

Table 2. Comparative study of test sessions and test time constructed by [9] and Heuristic Algorithm

Number of TSVs, Number of redundant TSVs, Number of probe pin (n, r, p)	Number of defective TSVs identified (m)	Average Case of Dynamically optimized test in [9] (Number of test sessions, time in µs)	Average Case of Heuristic Method (Number of test sessions, time in µs)	Average case reduction by Proposed Heuristic Algorithm over Dynamically optimized test in [9] (sessions, time)
(8, 3, 3)	0	(5.0, 2.10)	(3.0, 1.26)	(40.00%, 40.00%)
	1	(5.3, 2.25)	(4.0,1.72)	(24.53%, 23.78%)
	2	(6.4, 2.71)	(5.4, 2.55)	(15.63%, 5.90%)
	3	(7.5, 3.17)	(6.71, 3.09)	(10.53%, 2.52%)
(12, 4, 3)	0	(7.0, 2.94)	(4.0, 1.68)	(42.86%, 42.86%)
	1	(7.5, 3.14)	(5.0, 2.12)	(33.33%, 32.52%)
	2	(8.7, 3.65)	(7.7, 3.38)	(11.49%, 7.51%)
	3	(10.3, 4.32)	(9.3,4.19)	(9.71%, 3.01%)
	4	(11.8, 4.97)	(10.45, 4.75)	(11.44%, 4.43%)
(15, 5, 3)	0	(8.0, 3.36)	(5.0, 2.10)	(37.50%, 37.50%)
	1	(9.6, 4.03)	(6.0, 2.55)	(37.50%, 36.77%)
	2	(11.1, 4.68)	(8.7, 3.67)	(21.62%, 21.52%)
	3	(12.6, 5.33)	(10.2,4.53)	(19.05%, 15.08%)
	4	(14.3, 6.03)	(12.5, 5.53)	(12.59%, 8.29%)
	5	(15.8, 6.66)	(14.1, 6.32)	(10.70%, 5.11%)
(20, 5, 4)	0	(9.0, 3.42)	(5.0, 1.90)	(44.44%, 44.44%)
	1	(10.8, 4.10)	(7.8, 3.02)	(27.78%, 26.34%)
	2	(12.3, 4.68)	(8.6, 3.37)	(29.92%, 27.93%)
	3	(13.9, 5.31)	(11.6, 4.64)	(16.55%, 12.67%)
	4	(15.1, 5.76)	(13.5, 5.52)	(10.60%, 4.17%)
	5	(18.0, 6.85)	(14.95, 6.11)	(16.94%, 10.80%)

6 Conclusion

3D ICs with TSVs address a major challenge in the semiconductor industry. Defects in TSVs decrease the yield and reliability of 3D ICs. Pre-bond testing ensures the yield of each die before it is stacked. Thus, designing a faster test model is important. In this paper, we have proposed a faster heuristic model to uniquely identify the defective TSVs. As we have seen, the proposed algorithm showed a better result compared to [5, 6, 8, 9] in terms of the percentage of reduction of test time. Besides this, the proposed model was able to identify more than 50% defective TSVs in the reduced test time.

References

1. Davis, W.R., et al.: Demystifying 3DICs: the pros and cons of going vertical. IEEE Des. Test Comput. **22**(6), 498–510 (2005)
2. Lee, H., Chakrabarty, K.: Test challenges for 3D integrated circuits. IEEE Des. Test Comput. **26**(6), 26–35 (2009)

3. Chen, H., Shih, J., Li, S.W., Lin, H.C., Wang, M., Peng, C.: Electrical tests for Three-Dimensional ICs (3DICs) with TSVs. In: International Test Conference 3D-Test Workshop, pp. 1–6 (2010)
4. Noia, B., Chakrabarty, K.: Pre-bond probing of TSVs in 3D stacked ICs. In: Proceedings of IEEE International Test Conference, pp. 1–10 (2011)
5. Noia, B., Chakrabarty, K.: Identification of defective TSVs in pre-bond testing of 3D ICs. In: Proceedings of IEEE Asian Test Symposium (ATS), pp. 187–194 (2011)
6. Zhang, B., Agrawal, V.D.: Diagnostic tests for pre-bond TSV defects. In: Proceedings of IEEE International Conference on VLSI Design, pp. 387–392 (2014)
7. Roy, S.K., Chatterjee, S., Giri, C.: Identifying faulty TSVs in 3D stacked IC during pre-bond testing. In: Proceedings of IEEE International Symposium on Electronic System Design (ISED), pp. 162–166 (2012)
8. Roy, S.K., Chatterjee, S., Giri, C., Rahaman, H.: Faulty TSVs identification and recovery in 3D stacked ICs during pre-bond testing. In: Proceedings of International 3D Systems Integration Conference (3DIC), pp. 1–6 (2013)
9. Zhang, B., Agrawal, V.D.: An optimal probing method of pre-bond TSV fault identification for 3D stacked ICs. In: IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S) (2014)