A Reliable Massively Parallel Testing Method for Wafer Testing

Haksong Kim, Yong Lee, Sungyoul Seo, and Sungho Kang
Yonsei University, Seoul Korea
hskim86, daiginda, sungyoul @soc.yonsei.ac.kr
shkang@yonsei.ac.kr

Abstract

For wafer testing, high parallelism is important in order to reduce test costs. However, increasing parallelism is becoming more difficult because there are limitations to how much the parallelism can be increased with design changes. To solve the limitations, wafer level test schemes based on the network-on-chip (NOC) have been developed. However, connecting to one device under test (DUT) has a reliability problem in which fault distribution causes good-DUT to be improperly tested is becoming a concern. In this paper, a new MPT method is proposed which achieves both its high parallelism and reliability with BT algorithm. The experiment results show that the proposed method achieves the nearly 100% reliability without affecting yield.

Keywords: wafer testing, test cost reduction, massively parallel testing (MPT)

1. Introduction

As the progress of wafer testing technology, today's mass-production testing requires more parallelism in system on chip (SOC) test development for test cost reduction. For increased parallelism, reduced pin count testing (RPCT) techniques have been proposed [1-2]. These works focus on reducing the pin count using a specific design in a chip such as IEEE Standard 1149.1 (Joint Test Action Group, JTAG), IEEE Standard 1500, built-in self test (BIST) and test data compression (TDC). However, there are limitations to how much the parallelism can be increased with design changes. In addition, the chip gets smaller and physical challenges increase. Therefore, other solutions are required for massive parallelization. As part of these efforts, wafer level test schemes based on the NOC have been developed which can be tested by one DUT connection only [3]. However, connecting to one root-DUT (which is directly connected to ATE) has a reliability problem. For example, three cases of a single root-DUT are shown in Figure 1. A faulty-DUT signifies not a failure in the functionality of the DUT but a failure on the path or test router logic in the DUT. A rectangle signifies a good-DUT which can be tested. An isolated-DUT cannot be tested by the faulty-DUTs in spite of the good-DUT. In Figure 1, the first case shows the failure of the root-DUT and the second cases show that the root-DUT is surrounded by the faulty-DUTs. Therefore, the two cases that show the good-DUTs cannot be tested because the path does not exist. In addition, the third case shows that the good-DUTs are isolated by faulty-DUTs. In this case, the isolated-DUTs cannot be tested because the path from the root-DUT to the isolated-DUTs does not exist. As a result, test failures show the limitations of a single root-DUT method. The significance of reliability is evaluated by how many good-DUTs are tested according to the yield of the faulty-DUT.

2. Proposed MPT method

For wafer testing, all DUTs are connected to the root-DUT which is connected directly to the ATE and the wafer testing time is dependent on that connection. In order to reduce the testing time, a balanced tree (BT) algorithm is proposed. It is possible to create a path with a balanced tree with various failure distributions. The balanced tree can obtain the lowest depth at about the same DUT count and reduces testing time because the ATE simultaneously sends test data via the root-DUT in four directions during each clock cycle. Therefore, the wafer testing time is determined by the depth of the tree and DUTs with equal depth have the same test time. If a failure occurs in the specific locations, the BT algorithm generates a new path. At this time, if ATE does not have to set a new path to the destination-DUT, it is isolated or the root-
DUT is isolated by the failure-DUTs. Therefore, to improve the testability and testing time, multiple root-DUTs are used in this paper. The process for path search using BT algorithm is shown in Figure 2.

**Step 1: Initial information setting**
**Step 2: Search path using Route R**
  - Step 2-1: Backtrack path search in Route R
  - Step 2-2: Search path using Route L
  - Step 2-3: Backtrack path search in Route L
**Step 3: Decide Pass or Fail**
  - Step 3-1: Save the destination-DUT information
  - Step 3-2: Change the destination-DUT
**Step 4: Post-processing of the stored DUTs**

![Figure 2: Process for BT algorithm](image)

In the first step, ATE sets the location information for the start-DUT and destination-DUT. Subsequently, the path search is performed in step 2. To determine the priority of the path search, this algorithm uses two variables which are called PX and PY such that there are gained according to value of the ID (X, Y coordinate based on the center) difference of destination-DUT and start-DUT. Path search steadily progresses in order of priority shown in Figure 3. For example, the destination-DUT is assumed to be a rectangle '7' in Route R. The rectangle '7' has 'Left, Up, Right, Down' priority. The priority of the path is changed when the path meets a faulty-DUT or does not exist. The relative position of destination-DUT from start-DUT is frequently changed. In other words, it can be changed when the path cannot go in the direction. If a specific condition (search count is exceeded or the path does not exist anymore) occurs, backtrack progresses. This action makes progress until finding the path that is not taken and not faulted. After finding the path, Route L begins. This transition operation (Route R \(\leftrightarrow\) Route L) is performed repeatedly whenever the above-mentioned situation is generated. In step 3, ATE determines that the path set has been completed or it does not exist with the failure-DUTs. If the coordinates of the start-DUT are equal to the first position, the path with Route R first does not exist or the DUT (or root-DUT) has been isolated. In this case, the ATE attempts to search the path using Route L first in order to set a path. If the search also fails, step 3-1 and 4 are in progress. On the other hand, if the path search is successful, the path setting of the DUT is completed and step 3-2 is in progress. Finally, in the final step, the stored-DUT information from step 3-1 is handled by next root-DUT. Then, stored-DUTs are performed in steps 2 to 3 by the changed root-DUT. If the path connection of some DUTs does not succeed, it is the isolation by the failure-DUTs or the failure-DUT. As a result, all DUTs can be connected to the root-DUTs except the failure-DUTs and isolated-DUTs.

### 3. Experiment results

The reliability is dependent on the number of isolated-DUTs and the distribution of failure-DUTs and how many DUTs can be handled by the root-DUT. The results are obtained by sampling 100 times and the failure-DUT locations are chosen by a Gaussian random distribution method. The results mean that some DUTs which are determined to be an isolated-DUT cannot be tested using the single or two root-DUTs. However, most of them are available with the test using the four root-DUTs with 10,000 DUTs.

![Figure 4: Reliability according to the yield](image)
4. Conclusion

For wafer testing, the proposed MPT method is considered for both its high parallelism and reliability with BT algorithm. The experiment result shows the nearly 100% reliability without affecting yield. According to the result, more root-DUTs will increase the reliability. However, whenever the number of root-DUTs increases the probe card overhead increases. In other words, the number of needles on the probe card causes a rise in manufacturing costs and comprises the largest portion of the probe card overhead. As a result, a reasonable decision about the root-DUT count is needed considering the probe card overhead and reliability. Current SOC testing performs 4-para (4 DUTs) at the same time. Therefore, it is reasonable to use the four root-DUTs in the proposed MPT method.

Acknowledgement
This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2012R1A2A1A03006255).

References
