Clustering-Based Production-Line Binning of ICs Based on I_{DDQ}

Ritesh P. Turakhia
Department of Electrical and Computer Engineering
ritesh@engr.colostate.edu

Anura P. Jayasumana
Department of Electrical and Computer Engineering
anura@engr.colostate.edu

Yashwant K. Malaiya
Department of Computer Science
malaiya@cs.colostate.edu

Colorado State University, Fort Collins, CO 80523

Abstract

A clustering-based technique is proposed for production line testing and real time binning of ICs. This paper presents a two-phase approach. The first phase involves off-line clustering and cluster characterization based on prior data. In the second phase, each device is sorted based on its quality attributes, into bins associated with specific quality and cost parameters. This allows fast real-time sorting of ICs on production line use of I_{DDQ} test methodology. Use of clustering algorithm provides a high-resolution technique for identifying groups of devices with similar characteristics. Proposed technique is tested on production test data and results are presented.

1. Introduction

I_{DDQ} testing has been shown to be an efficient and practical testing procedure to identify physical defect in the devices [1,8,9,12,19]. Research is now being pursued to overcome limitations associated with threshold setting [2,15,17,20,24]. With the background current of devices elevated due to scaling, resolution of quiescent currents for defective and good devices has decreased. Not all devices with I_{DDQ} higher than the normal are necessarily bad. In addition, devices with higher operating frequency tend to have higher I_{DDQ} [13]. Thus, it is not possible to classify devices just as defective or non-defective based on a hard threshold value or high current variation. Methods like Current Signatures [6,7,16] and ΔI_{DDQ} Testing [18, 23] compare the quiescent current values against a threshold limit or look for significant variation in quiescent current pattern of a device to identify it as defective. Statistical Post Processing technique like nearest neighbor residual (NNR) reduces variance of good and faulty I_{DDQ} distributions by systematic use of die location and wafer or lot-level patterns for improved identification of die outliers [3,4]. A non-defective device can be identified based on the quality and performance parameters such as leakage, power and speed. A high-resolution technique is needed to make such a classification that can be a part of the production line. By associating a quality and cost factor with each classification, the yield can be optimized.

In this paper, we present a new technique based on clustering. This technique will allow the clustering-based solution of I_{DDQ} test data to be used in production line testing. As a result, the device under test (DUT) will be sorted and binned based on their analog characteristic being used as a parameter for classification.

Clustering techniques have been used for a long time in the fields of Image Processing, Pattern Recognition and Pattern Matching. Cluster Analysis techniques are used wherever data from a large data set is to be sorted into smaller groups such that members of each group display a high degree of similarity within the group. Use of clustering technique for I_{DDQ} testing was first suggested in [10,11]. Several benefits were observed by this technique for I_{DDQ} testing. The technique was found to be effective in the presence of high background current. It provides an effective way of binning devices and cost savings in yield improvement. Savings in time and effort required for Failure Analysis (FA) is another significant advantage of this technique. The technique was also found effective when compared against SEMATECH test methods, single threshold approach and delta-I_{DDQ} approach.

The approach was further optimized in [21] which proposed criteria for setting the number of clusters needed that would give best grouping for maximum defect identification. They also provided a test decision methodology to assess the quality of the groups formed by clustering technique. Introducing the quality factor give finer control over the quality of good devices obtained from the test and thus affects the yield of the testing technique. The technique is also able to overcome certain disadvantages in current signature approach [6,7]. The applicability of clustering-based solutions in production environment has not been investigated yet. This paper further extends the clustering technique for use in production line.

The crux of this new method is to match the analog characteristic of the (DUT) with a group of devices having similar characteristics. Based on this similarity of parameters, the DUT is binned into corresponding group of devices. As a result, we will have various bins, each with similar characteristics. In addition, each bin will have a quality and cost factor already attached with it.

Binning has been used previously to study the analog attributes related to IC testing and to predict the performance of device based on these attributes [14]. In addition, clustering applies self-binning to form these groups from given test data. Previous studies have proposed ways to maximize the test yield based on quality factor [10,11,21]. This paper adopts clustering-based techniques for testing and binning of IC's in a production environment. Proposed technique can improve the efficiency of testing. It can also assist in identifying suitable devices for reliability and failure analysis. Moreover, it will reduce the overall test time while preserving the advantages offered by the clustering-based test methods.

The paper is organized as follows. Section 2 explains the two clustering algorithm investigated. The new technique is presented in Section 3. Section 4 presents the results and discussion about the results obtained by applying the technique to the production data. Section 5 concludes the paper.

2. Clustering algorithms

The final grouping or clusters obtained depends on type of clustering algorithm and criterion used to identify the groupings from given data set. The criterion can be customized for given application and type of data. Generally clustering algorithms can be divided into two types: hierarchical and iterative.

2.1. Hierarchical clustering algorithm

Hierarchical clustering is a method to find grouping in the data, simultaneously over several scales, by creating a cluster tree. The tree is a multi-level hierarchy, where clusters at one level are joined to form clusters at the next higher level. This process continues until we reach the top of the hierarchy. In this kind of clustering, we have more control over the criterion and method of grouping the data into dissimilar groups of similar data sets.

For the hierarchical algorithm that is used for clustering the test data set, 'Euclidean' distance measure is used and 'centroid' algorithm is used for clustering the devices into groups. For a data set having N data points with each data point having P variables, the algorithm initially assumes that there are N groups. The data points nearest to each other, based on Euclidean distance between these data points, are merged to make groups. Every set of data points being merged is considered as a single entry represented by its centroid. After each merger the centroid of the groups are updated before a decision for new entry is made. Thus, hierarchical clustering identifies the natural groups within the data set.

The hierarchical clustering algorithm offers several advantages. The most evident advantage is that the algorithm identifies the natural sets of similar data points existing in a large data set, as opposed to k-means which forces k groups irrespective of existence of k groups in the data set. Moreover, the number of groups to be formed need not be specified before initiating the clustering algorithm. In addition, depending on the type of data and the kind of solution required the type of distance measure used and algorithm used to form the grouping can be decided.

2.2. K-means algorithm

Unlike the hierarchical clustering method, k-means does not create a tree structure to describe the groupings in the data; rather it iteratively creates a single level of clusters. K-means clustering can best be described as a partitioning method.

K-means uses an iterative algorithm that minimizes the sum of distances from each data point to its cluster centroid, over all clusters. This algorithm starts with K initial seed points that can be predefined or randomly chosen from the data set. It then starts adding other data points to these groups in such a fashion that the average sum of all points of a given group from the centroid of that group remains same or decreases by this new entry. In other words, the algorithm moves data points between all the clusters until the average sum within any cluster cannot be decreased further. This iterative process ends, when moving any data point from one group to any other group, will not decrease the average sum within that group or of the group to which this point is being moved to. The result is a set of clusters that are compact and well separated.

K-means algorithm suffers from some basic drawbacks. By defining, the number of groups (K) beforehand, the algorithm makes an underlying

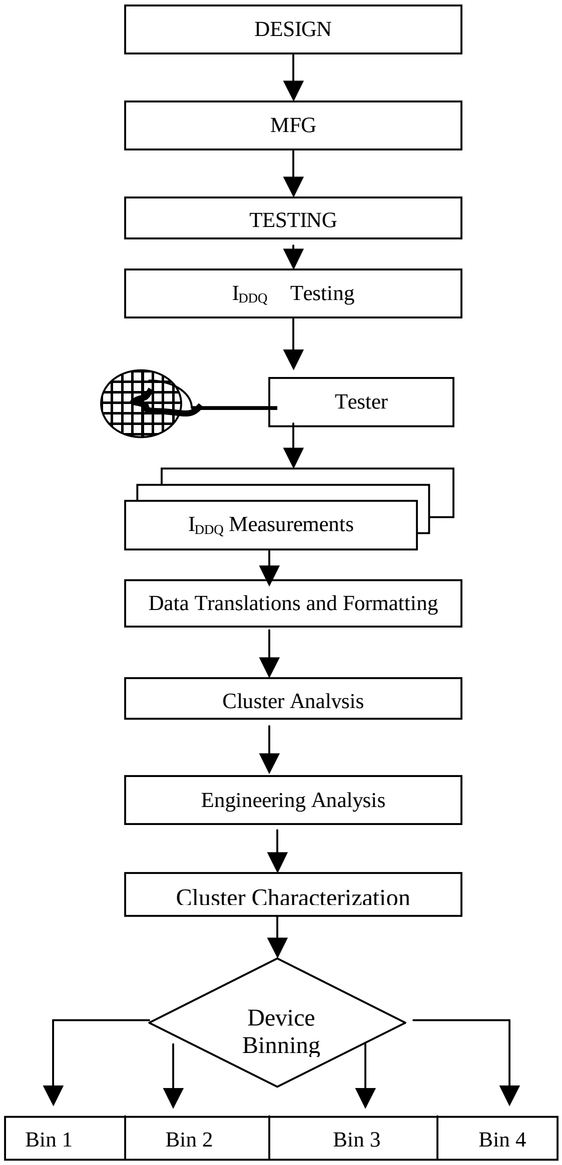

Figure 1. Phase 1: Cluster Pre-forming

assumption that there already exist K groups in the given data, which is not always true for real application data. It is also very dependent on starting (initial seeds) data points. Being an iterative method with a predefined number of groups, its computational complexity increases while solving massive clustering problems.

3. Clustering-based I_{DDQ} binning

The technique we propose has two different phases. The first phase identifies the clusters and the second

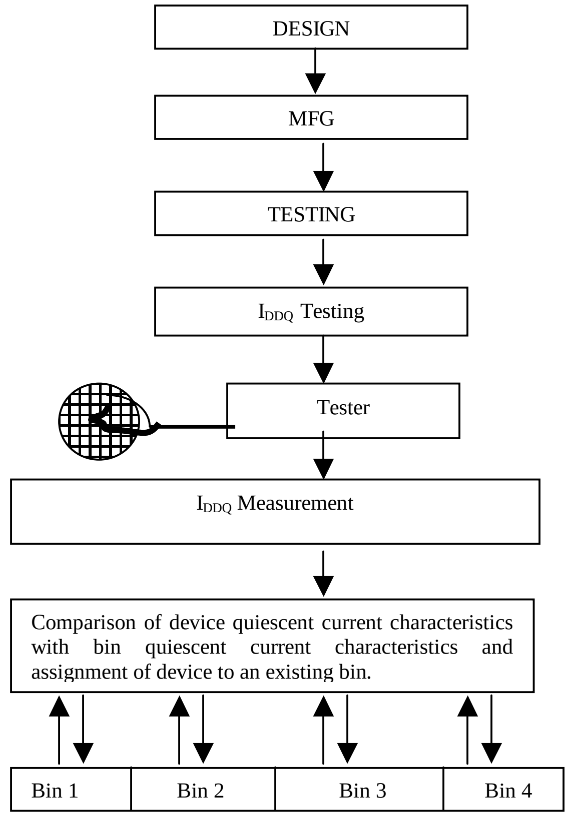

Figure 2. Phase 2: The Binning Process

phase actually does the binning. While the first part will use a sophisticated clustering algorithm, only limited computations need to be done during actual binning.

This paper addresses the second phase in detail. The existing scheme, given in [10], is represented here in Figure 1 with modifications to accomplish phase 1 of proposed technique. Figure 2 represents the steps in phase 2, which corresponds to the functions that are carried out during production line testing of DUTs.

3.1. The Two-Phase binning approach

3.1.1. Cluster pre-forming phase. In this phase, test data samples are collected and analyzed comprehensively. An analysis is done using the clustering process to obtain the best clusters in the collected data. These clusters are then characterized as defective, non-defective, high leakage, low power etc. Such a characterization takes into consideration, parameters like highest operating frequency, power requirement, performance, reliability and cost. These preformed clusters would be based on initial devices produced. However, it is also possible to modify these using certain desirable or undesirable analog and digital attributes. As this data analysis leads to identification of initial bins that have to be used for the next phase and will be responsible for the final yield, it is imperative that the clusters be identified and characterized very carefully. Once we get the ideal bin characteristics for a particular type of device it can be saved for use in future production of lots.

3.1.2. Binning phase. From the first phase, we have initial bins with their characteristics available. As each IC comes off the production line, it will be assumed to the closest cluster. As soon as the test parameters become available for each IC (Figure 2), its characteristics are compared with characteristics of all the bins and it is assigned to the bin that is the nearest match. Since this process involves simple evaluation of distance of the DUT to the existing clusters, it can be done in real-time, in a fraction of a second, thus not affecting the time and efficiency in a production environment.

Precaution should be taken to perform the binning process in the same fashion as the actual clustering algorithm being used to form the model bins. This precaution is necessary, as we want to preserve the efficiency and accuracy of clustering. Thus we maintain the characterization of the bins and just keep sorting the devices according to best match for a bin with the aim of optimizing the cost and quality factor to get the maximum yield.

Sections 3.2 and 3.3 explain the binning phase in detail for the two clustering algorithms described above.

3.2. Hierarchical clustering-based binning

We implement the proposed technique as follows:

1. The hierarchal clustering algorithm is applied on the first lot to obtain the initial clusters.
2. The test vector for the new DUT from the second lot is obtained and the Euclidean distances between this vector and centroid of each group obtained from the first phase is calculated. The device is assigned to the group to which it is near most.
3. The centroid of the group to which the device is added, is recalculated and updated for that group.
4. This procedure is carried out for all the DUTs of the new lot.

In step 3, by updating the centroid each time a new device is added to that group, the functionality of the algorithm is preserved.

3.3. K-means clustering-based binning

For the k-means clustering, we implement the proposed technique in following steps:

1. Initial groups are obtained using the k-means algorithm on the first lot.
2. The data vector for the first device from the second lot is obtained and the Euclidean distances between this vector and centroid of each group obtained from the first phase is calculated. The device is assigned to the group to which it is nearest.
3. To follow the k-means algorithm in this case, the average sum of distance of this point and each point of that group to the centroid should be calculated. This process will be repeated for every group and the device should be assigned to the group, which gives the minimum average sum. Clearly, such an approach will increase the complexity of binning process to a high degree and make it unusable in production environment. Thus, the next best approximation is followed and the device is assigned to whichever groups centroid it is closest.
4. The centroid of the group to which the device is added, is recalculated and updated for that group.
5. This procedure is carried out for all the DUTs of the new lot.

The next section presents the results obtained from the two methods described in sections 3.2 and 3.3.

4. Results and discussion

MATLAB (R13) was used as a computation tool for all the results and plots presented in this section and throughout the paper. While MATLAB is considerably slower than a general-purpose language like C, it provides a convenient higher-level platform for computation.

4.1. Test data

The test data being used to confirm the feasibility of the proposed technique was collected from a high volume device manufactured in deep sub-micron process at Texas Instrument [10]. The device has approximately 650K gates along with extensive DFT features including full scan. The test vectors were generated using commercial ATPG tool. Thirty I_{DDQ} measurements were taken on four lots containing 627, 724, 716 and 798 devices. We used the first lot for cluster pre-forming (phase 1) and lots 2, 3, 4 to evaluate the binning process. Fault coverage of 95% was obtained with these 30 vectors. Due to the proprietary nature of the data, it is shown in the normalized form.

4.2. Silhouette values and plots

Silhouette value is a very useful parameter to assess the quality of clusters obtained from any given cluster algorithm. Silhouette plots make it easier to visualize this quality. The silhouette value for each point is a measure of how similar that point is to points in its own cluster compared to points in other clusters. It is defined as [22]:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where, a(i) is the average distance from the ith point to all the other points in its cluster, and b(i) is the average distance from the ith point to all the points in the nearest neighbor cluster.

Clearly, the value of S(i) would range from +1 to –1. Greater the value of S(i) in each clusters better is the quality of cluster, as higher value of S(i) reflects that the points in that cluster are nearer to other points in the same cluster than to the points in the nearest neighbor to that cluster. For plotting the silhouette plots, all the values of S(i) for every cluster are sorted in decreasing order and then plotted. X-Axis represents the silhouette values and Y-Axis represents clusters. The number of devices in each cluster is mentioned on the right side of each plot.

4.3. Results for hierarchical clustering-based binning

Figure 3 presents the results for first phase using hierarchical clustering. This figure represents the characteristics of each cluster for all the 30 test vectors. There are thus 150 horizontal values (vectors 1 through 30 of each cluster for five clusters). These initial clusters were obtained from first lot containing 627 devices. In an industrial setting, the initial clusters may be obtained, as a result of the knowledge of design parameters, observation over several lots, evaluation of such observations based on parameter of interest, characterization of clusters based on parameter evaluation. The characterization is evaluated as per the customer requirement. A high degree of dissimilarity can be observed between all the clusters and it can be observed that clusters 1 and 3 are nearest neighbors while clusters 2 and 4 are nearest neighbors. Due to its distinct characteristics, the device in cluster 5 was not grouped with any other device. In addition, cluster 1 which seems to have devices with low I_{DDQ} current vectors is the largest. Due to the lack of details about design parameters and exact functionality of this device we were not able characterize the clusters.

One way to assess the quality of clusters is from its silhouette plot, Figure 4. Almost all the devices in cluster 1 are well correlated to other devices in the same group. The devices of interest for the test engineer would be the one displaying negative silhouette values (circled in figure 4). We call these devices as *weaker members* of that cluster. Due to the fact that these 31 devices are not close members to their cluster and nor are they assigned to any other cluster during the clustering process, they are most likely to be misinterpreted as displaying abnormal behavior (defective). This can be explained by looking at the mean values of clusters 1 and 3. Since clusters 1 and 3 are very close neighbors, these weaker members will display higher correlation to devices in 3.

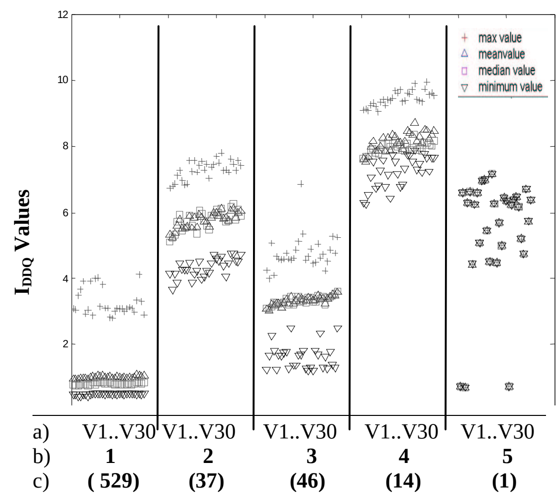

Figure 3. Cluster characteristics for all 30-test vectors using centroid method
a) vectors; b)cluster number; c)number of devices in the cluster.

Observing the characteristics of these weaker members show that their characteristics are normal (figure 5). Device 11 in figure 5 can be considered wrong assignment, as its characteristics are similar to devices of cluster 2.

For simulation purposes, we have assumed that figure 3 represents the model characteristics. The objective is to extract all the devices that have similar characteristics, from other lots too. The steps explained in section 3.2 were followed to bin the devices, from other three lots, based on characteristics in figure 3. Figure 6, shows the quality of clusters after all the 2238 device from lots 2, 3 and 4 were binned using the proposed technique. Again, the majority (2193) of devices were identified with cluster

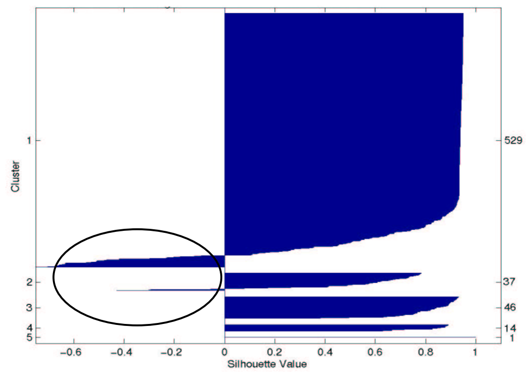

Figure 4. Silhouette plot for clusters from lot 1 using hierarchical clustering

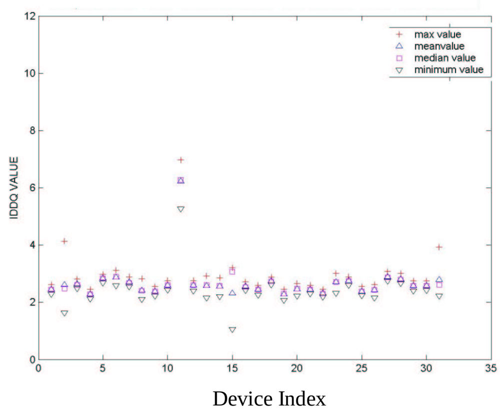

Figure 5. Characteristics of *weaker members* from cluster one

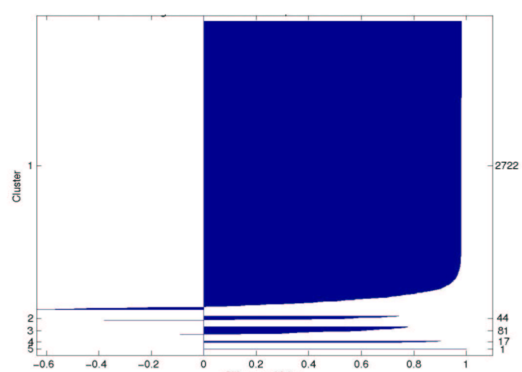

Figure 6. Silhouette plot for clusters obtained after binning devices from lots 2,3 and 4 using hierarchical clustering

1 and the remaining 45 devices are matched and binned with other devices. Moreover, the number of weaker members in the cluster did not increase. Of the total 2238 devices, none of the device from the other lots was a weaker member for cluster 1. Only one device from cluster 3 displayed negative silhouette.

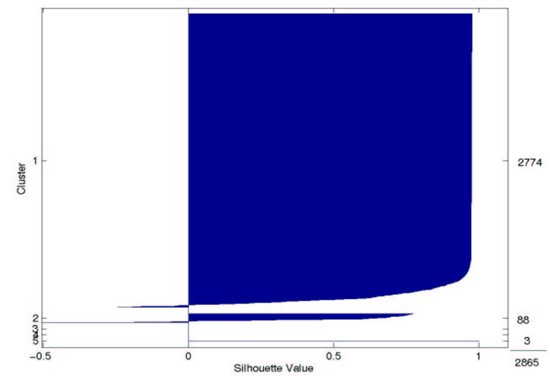

Figure 7. Silhouette plot for clusters obtained from clustering all the lots together using hierarchical clustering

In addition, introducing such a scheme has simplified the identification and screening of weaker members, as compared to previous approaches in [10,11,21] which applied complex techniques like re-clustering. Also binning has improved on accuracy offered by clustering technique. Comparing figure 6 with figure 7, it is evident that in the former case devices were binned accurately in one of the five clusters while for the latter case only three clusters were obtained from all the lots. If the clustering would have been performed, using the traditional technique, low dissimilarity is compromised and very close neighbors might merge. This fact is observed in Figure 7, where the devices of all the lots are clustered together producing only three clusters.

4.4. Results for k-means clustering-based binning

The analysis was repeated using K-means clustering method. Initial observation of figure 8 would reflect more accuracy in cluster identification. This might seem to be a better solution, but it suffers the drawbacks attached with k-means clustering. Since k-means forces k groups to be formed in the given data set, some of the defects might be suppressed due to merger of doubtful devices with good clusters. One evident example can be observed by comparing cluster three of figure 8 with cluster five of figure 3. The abnormal distribution (circled in figure 8) was included with cluster 3 in case of k-means while it was distinctly identified using hierarchical technique. Silhouette profile for figure 8 can be observed in figure 9. Although k-means algorithm tries to increase the cluster-to-cluster dissimilarity, in this case it is observed that due to very low dissimilarity in the device characteristics, k-means is able to increase the distance of weaker members from nearest neighbors but is not able to eliminate the negative silhouettes.

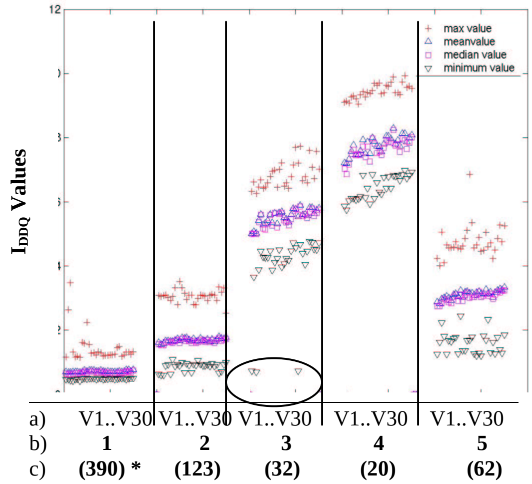

Figure 8. Cluster characteristics for all 30-test vectors using k-means method
a) vectors; b)cluster number; c)number of devices in the cluster.

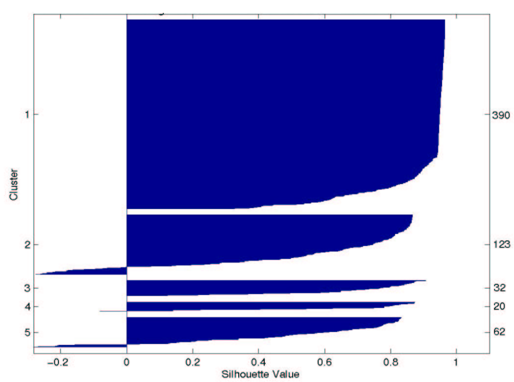

Figure 9. Silhouette plot for clusters from lot 1 using k-means clustering.

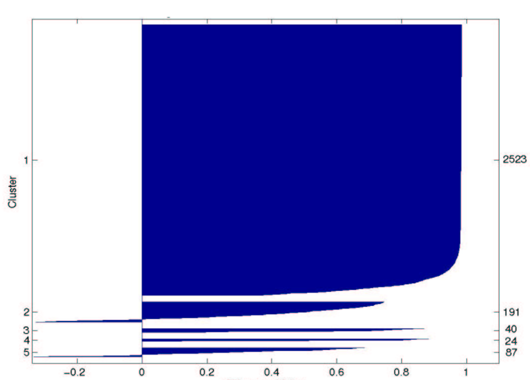

Figure 10. Silhouette plot for clusters obtained after binning devices from lots 2,3 and 4 using k-means Clustering.

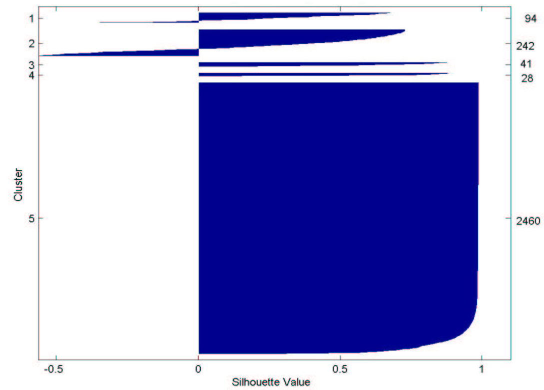

Figure 11. Silhouette plot for clusters obtained from clustering all the lots together using k-means clustering.

Binning of lots 2, 3 and 4 based on cluster characteristics obtained from phase one applied on lot 1 proved effective in the case of k-means too. Surprisingly, even after compromising on the actual k-means algorithm during the binning phase, the technique was able to bin the other lots efficiently. The clusters generated by k-means had 23 weaker members from two clusters. The result from the binning phase for k-means is presented in figure 10. Although the result of binning was good, the effect of compromise in binning phase shows up in the number of weaker members introduced in that phase. From the first, there were 23 weak members but after binning all the other lots, 22 weaker members were introduced. The results of binning are compared with the traditional technique using k-means in figure 11.

4.5. Discussion

The term *uncertainty factor* is introduced in this section and is defined as the ratio of number of weaker members in the final clustering solution to the total number of devices being clustered (or binned).

The results for the binning process using two clustering methods are summarized in Table 1. The number of devices in each cluster is unique for each method used in cluster pre-forming phase. Cluster 1 is largest with same characteristics for both the approaches. All the 390 devices in cluster 1 for K-means method are subset of 529 devices obtained with hierarchical approach. Similar observation can be made for other clusters from pre-forming phase. Difference in distribution is due to the random selection of initial seed point in case of K-means. Moreover, since the characteristics of devices obtained using each method

Table 1. Statistics for the proposed technique

Method	Phases	# of Devices	CLUSTER 1	2	3	4	5	# of Weak Members	Uncert-ainty	Time for binning	Complexity
Hierarchical Method	Pre-forming Lot 1	627	529	37	46	14	1	31	4.90%	26 sec	$O(N^2 \log^2 N)$ [5]
	Binning lots 2,3,4	2238	2193	7	35	3	0	1	0.04%	4 sec	O(N)
K-means Method	Pre-forming Lot 1	627	390	123	32	20	62	23	3.60%	0.69 sec	O(IKN) [25]
	Binning lots 2,3,4	2238	2133	68	8	4	25	22	0.98%	4 sec	O(N)

also differ, the characterizations obtained for final bins will also differ for the two algorithms. The same observation can be made for the bins obtained after all the devices from lots 2, 3 and 4. The number of weak members for hierarchical phase was 31 making the initial uncertainty 4.9% which seems to be a high value. Before the binning process, if these weak members are not screened as explained in section 4.3, the other lots add one weaker member during the binning process making the uncertainty 0.04% for lots 2,3 and 4.

Another parameter of interest is the time it takes for the whole process. For the cluster pre-forming phase, hierarchical clustering takes 26 seconds as compared to 0.69 seconds in k-means. This can be mainly attributed to the complexity of the process (last column, table 1). Hierarchical clustering has more than quadratic complexity while K-means has linear complexity depending mainly on I number of iterations to achieve K groups from N data points. However, the cluster pre-forming phase is performed offline and it will not affect the binning process, which is implemented on the production line. Moreover, the binning process for both the methods take same time (4.0 seconds total, or approximately 2mseconds per DUT) due to same order of complexity O (N). The computation time was measured on an UltraSPARC processor with clock speed of 400 MHz.

Neither of the two algorithms was found superior for the present implementation. The choice of algorithm can be based on spread of characteristics in the lots examined for the first phase. In addition, computation complexities can play important role in this decision. Hierarchical clustering algorithm is bound to have increased complexities if the data set is very large. In contrast, the K-means algorithms' complexity depends on number of groups being formed and the size of data set. If the test data set has a more variability in characteristics of devices, K-means algorithm can be more complex compared to hierarchical method. For the present case, based on uncertainty and characteristics obtained hierarchical method provides better solution as compared to K-means. As far as the binning process is concerned, the number of iterations is same for both implementations.

5. Conclusion

Clustering-based methods can use a number of analog measurements to classify ICs into bins with different attributes. The bins can correspond not only to acceptable/bad classification, but also to devices suitable for a specific market segment. Because faulty IDDQ values can be comparable to normal values, multiple measurements need to be considered to enhance resolution. Statistical clustering is a powerful method for assigning ICs to specific bins. However clustering can be computation intensive if a large number of devices need to be binned. Here a technique is proposed that identifies clusters using off-line computation. Once characteristics of the clusters are recognized, a device coming off the production can be quickly assigned to a specific bin. The feasibility of such an implementation was studied on industrial test data. Results show that the proposed technique will require only minimal computation time per device and thus it can be used in a production line environment. We have compared two clustering algorithms in terms of devices that result in uncertainty.

Even though we have used IDDQ for the devices, it is possible to extend the method for other analog measurements that may be available. It is possible to minimize the impact of vector-to-vector or lot-to-lot variability by suitably normalizing the data. Further research is needed to evaluate the impact of such normalization.

Acknowledgments

Authors thank Sri Jandhyala and Texas Instruments for providing data for this work.

References

[1] J.Acken, "Testing for Bridging Faults (Shorts) in CMOS Circuits", Proc 20th DAC, pp. 717-718, June 1983.
[2] K. Baker and B. Verhelst, "I_{DDQ} testing because Zero Defects isn't enough: A Philips Perspective", Proc of IEEE Int. Test Conf. 1990, pp. 253-254.
[3] W.R. Daasch, J.McNames, D.Bockelman, K.Cota, R. Madge, "Variance Reduction using Wafer Patterns in IDDQ Data", Proceedings ITC, pp. 189-198, October 2000.
[4] R. Daasch, K.Cota, J. McNames, R.Madge, "Neighbor Selection for Variance Reduction in IDDQ and Other Parametric Data", ITC, 2001. pp 1240-1248
[5] D. Eppstein, "Fast Hierarchical Clustering and Other Applications of dynamically closest pairs". The ACM Journal of Experimental Algorithms, June 2000, 5(1), 1-23.
[6] A.Gattiker and W.Maly, "Current Signature", Proc. of IEEE VLSI Test Symp. 1996, pp. 112-117.
[7] A. Gattiker, et al., "Current Signatures for Production Testing", Proc of IEEE Int. Test Conf. 1996, pp.724-729.
[8] C.F. Hawkins, et al., "Quiescent Power Supply Current Measurement for CMOS IC Defect Detection", Proc of IEEE Trans. On Industrial Electronics, 1989, pp 211-218.
[9] T. R. Henry, Thomas Soo, "Burn-in Elimination of a High Volume Microprocessor Using I_{DDQ}", Proc of IEEE Int. Test Conf. 1996, pp.242- 249.
[10] S.Jandhyala, H. Balachandran and A. Jayasumana, "Clustering-based Techniques for I_{DDQ} Testing", Proc of IEEE Int. Test Conf. 1999, pp. 724-729.
[11] S.Jandhyala, H. Balachandran, M. Sengupta and A. Jayasumana, "Clustering-based Evaluation of I_{DDQ} Measurements: Application and Classification of IC's", Proc of IEEE VLSI Test Symposium, April 2000, pp. 444-449.
[12] D.Josephson, Mark Storey, Dan Dixon, "Microprocessor IDDQ Testing: A Case Study", IEEE Design & Test of Computers, summer 1995, pp. 42-52.
[13] A.Keshavrazi, K.Roy, M.Sachdev, C. Hawkins, K. Soumyanath, V. De, "Multiple parameter CMOS IC testing with increases sensitivity for I_{DDQ}" Proc International Test Conference, 2000, pp. 1051-1059.
[14] J.Lee, D.M.H. Walker, L. Milor, Y.Peng and G.Hill, "IC Performance Prediction for Test Cost Reduction", 1999 Int. Test Symp. On Semiconductor Manufacturing, Nov. 1999.
[15] Chin-Wen Lu, Chauchin Su, Chung Len Lee and Jwu-E Chen, "Is IDDQ Testing not Applicable for Deep Sub micron VLSI in Year 2011?", Proc of the Ninth Asian Test Symp., 2000, (ATS 2000), pp. 338 –343.
[16] P. Maxwell, et al., "Current Ratios: A Self-Scaling Technique for Production I_{DDQ} Testing" Proc of IEEE Int. Test Conf. 2000, pp 1148-1156.
[17] S. McEuen, "I_{DDQ} Benefits", Proc of IEEE VLSI Test Symp. 1991, pp.285-290.
[18] C. Miller, "I_{DDQ} Testing in Deep Sub Micron Integrated Circuits", Proc of IEEE Int. Test Conf. 1999, pp724-729.
[19] L.K.Mourning, et al., "Measurement of Quiescent Power Supply Current for CMOS ICs in Production Testing", Proc of IEEE Int. Test Conf. 1987, pp 300-309.
[20] R. Perry, "I_{DDQ} testing in CMOS Digital ASIC's- Putting it altogether", Proc. of IEEE Int. Test Conf.1996, pp. 151-157.
[21] A.Rao, A.P.Jayasumana and Y.K.Malaiya,"Optimal Clustering and Statistical Identification of Defective ICs using I_{DDQ} Testing", Proc of IEEE Int. Workshop on Defect Based Testing, 2000, pp. 30-35.
[22] P.J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.", *Journal of Computational & Applied Mathematics 20*, 1987. pp 53-65.
[23] C. Thibeault, " A Histogram Base Procedure for Current testing of Active Defects", Proc of IEEE Int. Test Conf. 1999, pp. 714-723.
[24] T.W. Williams, et al., "I_{DDQ} Test: Sensitivity analysis of Scaling", Proc. of IEEE Int. Test Conf. 1996, pp. 786-792
[25] O.Zamir, O.Etzioni, "Web Document Clustering: A Feasibility demonstration", Proceedings of the 21st Annual ACM SIGIR conference of Research and Development in Information Retrieval, 1998, pp 45-54.