3rd NASA Symposium on VLSI Design 1991

Measurement Selection For Parametric IC Fault Diagnosis [1]

A. Wu and J. Meador
School of Electrical Engineering and Computer Science
Washington State University
Pullman, Washington 99164-2752
meador@eecs.wsu.edu awu@eecs.wsu.edu
(509) 335-5363 FAX: (509) 335-3818

Abstract- **This paper describes experimental results obtained with the use of measurement reduction for statistical IC fault diagnosis. The reduction method used involves data pre-processing in a fashion consistent with a specific definition of parametric faults. The effects of this preprocessing are examined.**

1 Introduction

An integrated circuit test is specified by a combination of input and output signals which characterizes some attribute of ideal circuit function. The presence of faults in a fabricated circuit will cause observed output signals to deviate from the simulated ideal. A fault diagnostic is a decision rule combining what is known about an ideal circuit test response with information about how the response is distorted by fabrication variations and measurement noise. The rule is used to detect fault existence in fabricated circuits using real test equipment.

The IC failure diagnosis problem can be viewed as a statistical pattern recognition problem. Instead of extracting output response parameters explicitly and comparing with the specification, the output responses can be identified into faulty or non-faulty according to some classification decision rules. It has been positively demonstrated that pattern classification technique can be used in IC diagnosis [Mea90].

Recent experiments [Mea91] have showed that feedforward network classifier (FFN) generally perform as well as or even better either than the traditional statistical parametric classifier, Gaussian Maximum Likelihood Classifier (GML) or the non-parametric classifier, the K-nearest Neighbors classifier (KNN). However, it usually needs more computational efforts for FFN in the training phase to establish the discriminant function. To be more effective, there is a need to find ways to consistently reduce this training overhead, while simultaneously retaining prediction accuracy.

Nevertheless, performance of a classifier depends on the data presented in the training, the discriminant function established in the training phase as well as the classification algorithm of the classifier. To ensure high performance accuracy, essential information has to be presented in the training data for the establishment of the discriminant function of the classifier.

[1] This work supported by NSF-UIC CDADIC Project 90-1.

In IC diagnosis, determination of a circuit fault is to classify the circuit from the input-output measurement according to a decision rule which is built upon the estimated prior probability distribution in the performance space of the circuit. Judgment is made according to the decision rule of the classifier established in training, which defines the decision boundaries for the classification. For accurate classification, decision boundaries of the classifier has to be coincided with or close to the performance specification criteria or boundaries. Such decision boundaries has to be captured by the classifier to set up the discriminant function or decision rules in training. The acceptance region of the fabricated circuits lines between the upper and lower performance specification limits. For highest accuracy, in the training phase, the decision boundaries have to be built around the specification transition space. In this case, they are the upper and lower specification limits instead of the mean of the performance distribution.

It is a well known fact that, in back-propagation training algorithm, input values are multiplied by the derivative of the logistic function, such that, a window is placed on the current estimated decision boundaries, not the mean [Lip88]. This character is very important in IC diagnostic problem especially the go/no go testing. If the discriminant function is established around the specification boundaries, it will improve the performance of the classifier. Besides, training on these boundaries, it will improve the training computational load.

To improve the training effort, it is therefore logical to train a FFN based on the decision boundaries. If data used in the training is collected around these boundaries, the discriminant function computed by the trained network will be more accurate in these regions. It will improve the training computational load as well, since fewer epochs are required to converge to a given accuracy. This paper reports on experiments conducted to help verify this idea.

2 Data Reduction Method: Boundary Band Data Preprocessing

In contrast to the design task, the concern of IC fault diagnosis is mainly on whether the circuit performance fall within the acceptance region instead of the performance mean. In other words, the specification transition boundaries are the most concerns in IC diagnosis. If the decision rules or decision boundaries of any diagnosis algorithm are based on these transition boundaries, it is reasonable to expect a highly accurate and effective diagnostic capability.

As discussed in the preceding section, there is a need to improve the computational load of training FFN classifier even though it has a better diagnostic capability than the other traditional statistical classifiers. Here, we proposed a Boundary Band Data (BBD) training method for FFN training to improve the computational load in the training phase. The essence of the proposed method is based on the characteristic of FFN. In back-propagation training algorithm of FFN, input values are multiplied by the derivative of the logistic function, such that, a window is placed on the current estimated decision boundaries, not

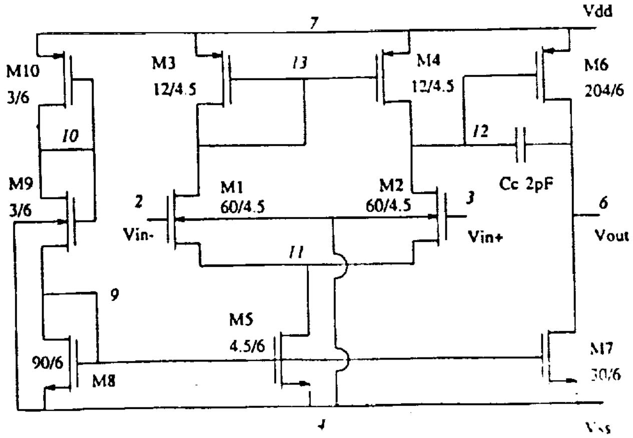

Figure 1: Operational amplifier circuit diagram

the mean. If the discriminant function or decision boundary of the trained network is set up around the specification transition boundaries, it will improve the performance of the classifier. Besides, training on these boundaries, it will improve the training computational load. Making use of this distinctive characteristic of FFN, the proposed Boundary Band Data training method is to train a FFN with those data gather from the proximity of performance specification transition boundaries.

3 Experiment Set Up

To investigate the feasibility of using BBD in FFN training, experiments were conducted in this study. The transient response and frequency response of the operational amplifier shown in Figure 1 were used for the experiments. For frequency response experiments, the open loop frequency response of the operational amplifier were used. For transient response experiments, the step response of an inverting amplifier with the same operational amplifier for the frequency response experiments are used. The circuit configuration for the transient response experiment is shown in Figure 2.

4 Fault Definition

All experiments were designed to detect parametric faults in an operational amplifier. Monte Carlo simulation of MOSFET model parameters was used. Only those statistical independent model parameters were used so that the correlation effect among model parameters was eliminated. In each of the experiments, circuit fault was defined as a large variation in one of the independent model parameter. In our experiments, three types of parametric faults were used. They were variations in MOSFET oxide thickness (of all the transistor in the circuit), zero bias threshold voltage, and junction depth.

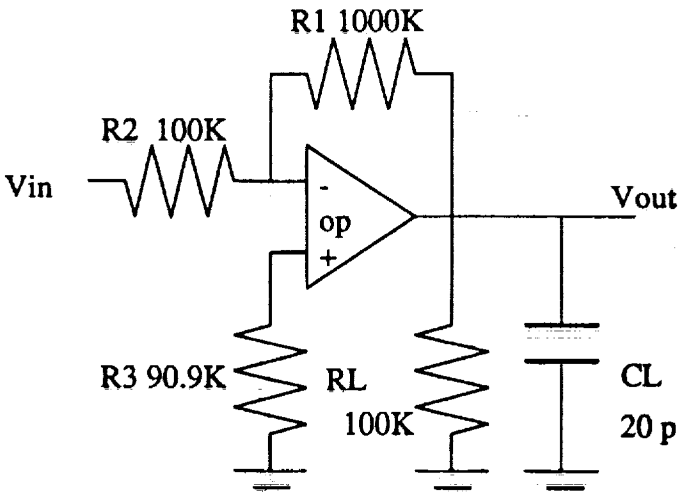

Figure 2: Inverting amplifier circuit configuration

Monte Carlo simulation using SPICE with a large variation around the mean value in the chosen model parameter was used. With a pre-selected performance criteria, the appropriate upper and lower limits of the model parameter which defined the fault and normal transition boundaries could be determined. For example, we were interested to define the fault and normal boundaries for the experiment with the inverting amplifier circuits. The fault was chosen to be variation in oxide thickness. So the SPICE model parameter, *tox*, was chosen with mean value equal to 600Å. The related circuit performance criteria were the step response overshoot and the slope of the stem response. Using the above method, the transition boundaries were set at 400Å and 800Å.The acceptance region was set between these limits. Any circuit fell within this region was defined to be normal; otherwise, it was defined as faulty.

In this study, three types of parametric faults were studied. They were circuit faults in oxide thickness, junction depth, and zero biased threshold voltage. For each of the experiments, there was only a single parametric fault existing in the circuit. The mean model parameter values for normal circuit and transition boundaries for circuit faults are listed in Table 1.

5 Experiment Description

Eight experiments which were divided into two categories were investigated for the BBD training methods. For the first category, it consisted of six experiments. To simplify the problem, in each of the experiments, only one SPICE model parameter was allowed to alter. It was under the assumption that there was no process existing in IC fabrication except the process fault. Even though such assumption might not be realistic for actual IC fabrication, the goal of these non-noisy experiments was to study the effect of the proposed BBD training method under the ideal condition. In these non-noisy experiments, the

Experiment	Parametric Fault	Performance parameters	Mean	Upper Limit	Lower Limit
1	oxide thickness	slew rate and overshoot	600Å	800Å	400Å
2	junction depth	slew rate	0.4μm	0.6μm	0.2μm
3	threshold voltage	slew rate	$\vert 0.7\text{V}\vert$	$\vert 0.5\text{V}\vert$	$\vert 0.9\text{V}\vert$
4	oxide thickness	slew rate	600Å	800Å	400Å
5	junction depth	slew rate	0.4μm	0.6μm	0.2μm
6	threshold voltage	slew rate	$\vert 0.7\text{V}\vert$	$\vert 0.5\text{V}\vert$	$\vert 0.9\text{V}\vert$
7	oxide thickness	slew rate and overshoot	600Å	800Å	400Å
8	junction depth	slew rate	0.4μm	0.6μm	0.2μm

Table 1: SPICE model parameter mean and transition boundary values

transient response of the amplifier in a closed loop inverting circuit under various nominal and faulty conditions was used to develop the experiment database for the experiment 1 to 3. For experiments 4 to 6, open loop frequency response of the amplifier under the same nominal and faulty conditions as in experiments 1 to 3 were used. The six non-noisy experiments were:

Exp.1: Detect a 33% variation in oxide thickness of all the transistor in the circuit by observing the circuit open-loop frequency response.

Exp.2: Detect a 50% variation in junction depth of all the transistor in the circuit by observing the circuit open-loop frequency response.

Exp.3: Detect a 30% variation in threshold voltage of all the transistor in the circuit by observing the circuit open-loop frequency response.

Exp.4: Detect a 33% variation in oxide thickness of all the transistor in the circuit by observing the circuit time-domain step response.

Exp.5: Detect a 50% variation in junction depth of all the transistor in the circuit by observing the circuit time-domain step response.

Exp.6: Detect a 30% variation in threshold voltage of all the transistor in the circuit by observing the circuit time-domain step response.

The second categories of the experiments consisted of two experiments which were similar to experiment 4 and 5 with the difference that there were process noise existed. It was under the assumption that there were process noises in the fabrication but not contributed to circuit faults. Such assumption was more realistic for actual IC fabrication. The goal of these experiments was to study the effect of BBD training of FFN under the non-ideal environment. Those process noises were generated by varying those statistical independent model parameters [She88] of lateral diffusion (LD), substrate doping density (NSUB), bulk threshold parameter (gamma), and channel-length modulation (lambda) at most one percent. In these two experiments, the transient response of the amplifier in a closed loop inverting circuit under various nominal and faulty conditions was used to develop the experiment database. For these noisy experiments, only one parametric fault were assumed but accompanied with all the process noises listed above. The noisy experiments were:

Exp.7: Detect a 33% variation in oxide thickness of all the transistor with process noise in the circuit by observing the circuit time-domain step response.

Exp.8: Detect a 50% variation in junction depth of all the transistor with process noise in the circuit by observing the circuit time-domain step response.

6 Data Generation

In each of the experiments, two data distributions namely normally distributed data and boundary band data as shown in Figure 3 and 4, were used to build up the experiment database. 120 simulated responses were obtained via a Monte Carlo simulation for each data distribution. Data for the boundary band distribution were generated around the transition boundaries. The sample data distribution of each of the experiment is similar to Figure 3 and 4 with difference in variation percentage of the corresponding model parameter. And the corresponding circuit performance distribution from the two data distributions were showed in Figure 5 and 6. 60 of the responses correspond to the fault free condition and 60 correspond to the faulty condition. In other words, there were four set of data consisting the experimental database for each of the experiments. The data sets were data for faulty circuit with normal distribution, data for normal circuit with normal distribution, data for faulty circuit with boundary band distribution, and data for normal circuit with boundary band distribution. For a particular data distribution, 30 responses from each class (normal/faulty) were used for classifier training. After training, classifier were tested on the unseen data from the trained data distribution as well as the data from the other type of distribution.

7 Classifier Training

As mentioned in the introduction, the objective of this study is to contrast the effectiveness of a feedforward network classifier trained on boundary band data against that of traditional statistical classifiers trained on normally distributed data and feedforward network as well in the context of IC fault diagnosis. Classifiers used in this study were Gaussian Maximum Likelihood Classifier, K-Nearest Neighbor Classifier and Feedforward Classifier.

Thirty patterns chosen from each the normal and faulty class of each of the experimental database for the training. For GML, training data was used to build the corresponding mean matrix, covariance matrix and the inverse of covariance matrix. For KNN, training data was used as the base for the classifier. For FFN, different types of training were used

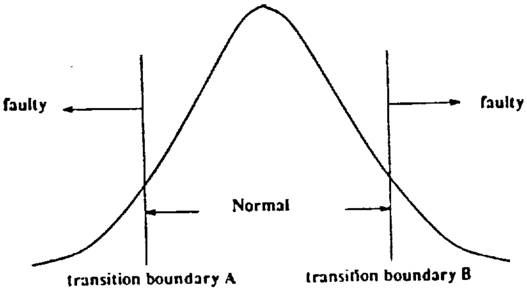

Figure 3: Data with normal distribution

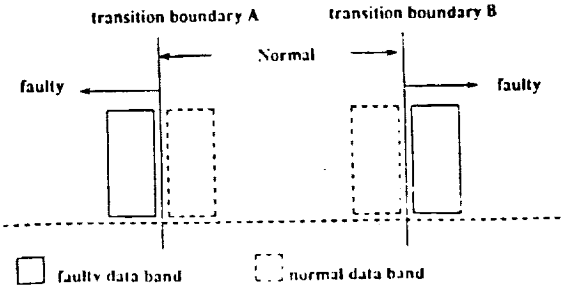

Figure 4: Data with boundary nand distribution

to establish the discriminant function for the corresponding FFN. There were FFN trained with non-noisy normal distributed data, non-noisy boundary band data, noisy normal distributed data, and noisy boundary band data. For each trained FFN, only one of the listed training method was used. Unlike traditional statistical classifiers, there are some training criteria can be chosen. We trained our FFN based on the total sum of square error of all the training data for a particular type of training or up to a preset training epoch limit.

8 Classifier Computational Load Calculation

The performance of each classifier was not only measured in terms of predictive accuracy on previously unseen data, but also the number of floating point operations (FLOPS) required to construct the classifier, and the number of FLOPS required to perform a diagnostic classification. Number of FLOPS computed for each of the classifier of each experiment is based on the implementation algorithm. It is not the actual computer operation. Since different software packages are used in the implementation of the classifier, it is not accurate if they are compared based on the real CPU time. In comparing computation requirement in testing , number of flops required per pattern are calculated with the equations $2y(n+1)+2z(y+1)$ for Feedforward network, $mn(3+2n)$ for Gaussian Maximum Likelihood classifier, $3mnp$ for K-Nearest Neighbor classifier. (m: no. of class, n: no. of measurement for each pattern, y: no. of hidden unit, z: no. of output unit, p: no. of pattern for the training set in each class)

9 Experiment Results

In each of the experiments, the performance of the classifiers were evaluated for the prediction accuracy of unseen data as well as the training and testing computational load. The results of non-noisy experiments 1 to 6 are summarized in Table 2 and Table 3. The results of noisy experiments 7 and 8 are summarized in Table 4.

Classifier	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Accuracy (unseen data - %)						
FFN(Norm)	82.2	88.8	97.7	88	95.5	98.8
FFN(Boun)	99.4	97.2	98.8	97.2	98.8	96.6
FFN(ReBo)	98.8	97.7	100	98.3	99.4	97.2
GML	N.W	N.W	N.W	50	N.W	N.W
1NN	85	96	98.3	50	96	100
3NN	81.6	100	98.3	53.2	100	100
5NN	80	96	98.3	50	96	100
Setup FLOPs (total)						
FFN(Norm)	3e8	3e8	3e8	2.7e8	2.7e8	2.7e8
FFN(Boun)	3e8	3e8	3e8	2.7e8	2.7e8	2.7e8
FFN(ReBo)	6e7	6e7	3e7	5.4e7	5.4e7	2.7e8
GML	N.A	N.A	N.A	5e4	N.A	N.A
KNN	none					
Diagnostic FLOPs (per pattern)						
FFN(Norm)	1.5e3	1.5e3	1.5e3	1.4e3	1.4e3	1.4e3
FFN(Boun)	1.5e3	1.5e3	1.5e3	1.4e3	1.4e3	1.4e3
FFN(Rebo)	1.5e3	1.5e3	1.5e3	1.4e3	1.4e3	1.4e3
GML	N.A	N.A	N.A	5.6e3	N.A	N.A
KNN	1.5e4	1.5e4	1.5e4	1.3e4	1.3e4	1.3e4

N.A: Not applicable

N.W: Not working for the case due to singular covariance matrix

GML: Gaussian Maximum Likelihood Classifier

KNN: K-Nearest Neighbors Classifier

FFN(Norm): Feedforward network trained with normal distributed data

FFN(Boun): Feedforward network trained with boundary band data

FFN(ReBo): Feedforward network trained with boundary band data and reduced training epoch

Table 2: Classifier Accuracy and Computational Overhead of Exp. 1 to 6

We trained three FFN for each of the experiments. There was FFN trained with normal distributed data. In this training, the FFN (Norm) was trained up to a point that the *tss* did not changed much with ongoing training. We determined the training stopping point to be 2e5 epochs. FFN (Boun) was a FFN trained with the boundary band data around the transition values. The same training stopping point was used as in FFN(Norm). The third type of training used in these six experiments was a FFN trained with boundary band data but with fewer training epochs. The stopping point of this training method depended on the prediction accuracy of the trained FFN. We stopped the training whenever the prediction accuracy of the trained FFN was similar to the FFN(Boun).

From the results of the experiments 1 to 6, summarized in Table 2, it shown that feedforward networks had better performance than GML and KNN. Besides, in general, FFN trained with boundary band data had better prediction accuracy than that of FFN trained with normally distributed data. These results were observed as predicted in the proposed method section. It was because the decision boundaries of the trained networks were expected to set around the transition boundaries in the performance space. Moreover, trained with fewer epoch, in general, it had a better prediction accuracy. It was because of the nature of neural network. With fewer training, it might eliminate the trained network from memorize the training data. In most of the cases, inverse covariance matrix of the training data for GML could not be computed without further data preprocessing.

To investigate the effectiveness of boundary band training, different types of training were used in our experiment. There were FFN trained with non-noisy normal distributed data, non-noisy boundary band data, noisy normal distributed data, and noisy boundary band data. For each trained FFN, only one of the listed training method was used. The results of FNN trained with these method of the non-noisy experiments are summarized in Table 3. Experiments with process noises are summarized in Table 4.

Results shown in Table 3 and 4 were the prediction accuracy of FFN with different training methods. Each trained FFN was tested on the unseen data from both of the normal distributed database and boundary band database. As showed in Table 3 and 4, there were two prediction accuracy for each of the trained FFN which tested on unseen data from normal distributed database (labeled Normal) and from boundary band database (labeled Boundary). It showed that the boundary band training did work on both non-noisy and noise cases. In general, with fewer training epochs, FFN trained with boundary band data performed as well as and even better in some case than FNN trained with normal distributed data. And, there were a large training epochs and prediction trade off of FNN trained with boundary band data. Using this method, there was very few prediction degradation but with a significant reduction in computation load spending on training.

Frequency Response	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Accuracy(unseen data–%) with various distribution						
FFN(Norm) trained with normally distributed data						
Normal	95	96.6	98.3	96.6	96.6	98.3
Boundary	75.8	85	97.5	85	95	99.1
FFN(Boun) trained with boundary band data						
Normal	99.1	98.3	99.1	96.6	98.3	96.6
Boundary	100	95	99.1	98.3	100	100
FFN(ReBo) trained with boundary band data and reduced training epoch						
Normal	98.3	97.5	100	98.3	99.1	97.5
Boundary	100	98.3	100	98.3	100	96.3
Setup Computation Load (Training)						
FFN(Norm) trained with normally distributed data						
Epoch	2e5	2e5	2e5	2e5	2e5	2e5
Flops	3e8	3e8	3e8	2.7e8	2.7e8	2.7e8
Load	100%	100%	100%	100%	100%	100%
FFN(Boun) trained with boundary band data						
Epoch	2e5	2e5	2e5	2e5	2e5	2e5
Flops	3e8	3e8	3e8	2.7e8	2.7e8	2.7e8
Load	100%	100%	100%	100%	100%	100%
FFN(ReBo) trained with boundary band data and reduced training epoch						
Epoch	4e4	4e4	2e4	4e4	2e4	2e4
Flops	6e7	6e7	3e7	5.4e7	2.7e7	2.7e7
Load	20%	20%	10%	20%	10%	10%

Table 3: Comparison of Feedforward Network with Different Training of Exp.1 to 6

Transient Response	Exp. 7	Exp. 8
Accuracy (unseen data–%) with various distribution		
FFN(Norm) trained with normally distributed data		
Normal	98.3	95
Boundary	94.1	94.1
FFN(Boun) trained with boundary band data		
Normal	97.5	99.1
Boundary	98.3	96.6
FFN(ReBo) trained with boundary band data and reduced training epoch		
Normal	99.1	96.6
Boundary	98.3	95
Setup Computation Load (Training)		
FFN(Norm) trained with normally distributed data		
Epoch	2e5	2e5
Flops	2.7e8	2.7e8
Load	100%	100%
FFN(Boun) trained with boundary band data		
Epoch	2e5	2e5
Flops	2.7e8	2.7e8
Load	100%	100%
FFN(ReBo) trained with boundary band data		
Epoch	4e3	4e3
Flops	5.4e6	5.4e6
Load	2%	2%

Table 4: Comparison of Feedforward Network with Different Training of Noisy Exp.7 and 8

10 Conclusion

We studied the effectiveness of a feedforward network classifier trained on boundary data against that of traditional statistical classifiers trained on normally distributed data and feedforward network as well in the context of IC fault diagnosis. Eight experiments with and without process noises were conducted. In this study, experiment results once again demonstrated, in general, that feedforward network out performed the traditional statistical classifiers namely Gaussian Maximum Likelihood classifier and K-Nearest Neighbor classifier. Feedforward networks trained with boundary band data, it reduced the training effort with only little prediction degradation. Experiment results showed that the proposed boundary band data did improve the computational load needed for feedforward network training.

References

[1] William Y. Huang and Richard P. Lippmann, Comparisons Between Neural Net and Conventional Classifiers, *Proc. International Joint Conference of Neural Network*, pp. IV-485-489, 1988.

[2] T. Lin, H. Tseng, A. Wu, N. Dogan and J. Meador, Neural Net Diagnostics for VLSI Test, *Proc. Second NASA SERC Symposium on VLSI Design 1990*, pp. 6.1.1-6.1.11, 1990.

[3] J.L Meador, A. Wu, C.T. Tseng and T.S. Lin, Mixed Signal IC Test, *NSF Center for Design Analog-Digital Integrated Circuit Technical Report 90-1*, Project # CDADIC 90-1, Jan 1991.

[4] Bing. J. Sheu, Chung-Ping Wah, Chih-Ching Shih, Wen-jay Hsu and Ming C. Hsu, Determination of Process-Dependent Critical SPICE Parameters for Application-Specific ICs, *IEEE Proc. International Conf. on Microelectronic Test Structure*, pp.73-78, 1988.