

Neural Net Diagnostics for VLSI Test

T. Lin, H. Tseng, A. Wu, N. Dogan and J. Meador
Department of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752

Abstract-

This paper discusses the application of neural network pattern analysis algorithms to the IC fault diagnosis problem. A fault diagnostic is a decision rule combining what is known about an ideal circuit test response with information about how it is distorted by fabrication variations and measurement noise. The rule is used to detect fault existence in fabricated circuits using real test equipment. Traditional statistical techniques may be used to achieve this goal, but they can employ unrealistic a priori assumptions about measurement data. Our approach to this problem employs an adaptive pattern analysis technique based on feedforward neural networks. During training, a feedforward network automatically captures unknown sample distributions. This is important because distributions arising from the nonlinear effects of process variation can be more complex than is typically assumed. A feedforward network is also able to extract measurement features which contribute significantly to making a correct decision. Traditional feature extraction techniques employ matrix manipulations which can be particularly costly for large measurement vectors. In this paper we discuss a software system which we are developing that uses this approach. We also provide a simple example illustrating the use of the technique for fault detection in an operational amplifier.

1 Introduction

An integrated circuit test is a combination of input and output signals which characterize some attribute of idealized circuit function. The presence of faults in a fabricated circuit will cause observed output signals to deviate from the simulated ideal. Unfortunately, variation of fabrication process and device parameters as well as measurement noise will also cause a deviation from ideal circuit performance, so something is needed which helps distinguish signal deviations due to fault existence from those due to these other sources. A diagnostic is a decision rule combining what is known about an ideal circuit test with information about how it is distorted by fabrication variations and measurement noise. The rule is used to detect fault existence in fabricated circuits using real test equipment. In this paper we discuss the application of neural network algorithms to the automatic synthesis of diagnostics for integrated circuit test.

Diagnostic synthesis is less concerned with specific aspects of test design than it is with the generation of a decision rule for a given test and process specifications. The focus of most test generation techniques is upon finding an appropriate combination of signals which will properly excite a circuit to reveal the existence of a potential fault. In diagnostic synthesis, a specific test has already been designed (typically under the assumption of a deterministic measurement process), and the job is to find some decision function which accurately reflects the outcome of that test in the real world (where measurements are random). In the event that a good diagnostic cannot be found for a given test, that information can provide feedback to the test designer so that a more robust variation can be created.

2 Statistical IC Diagnostic Synthesis

Diagnostic synthesis can be formulated as a statistical pattern recognition problem. This involves the generation of sample data and the analysis of that data using statistical tools. A priori assumptions about the measurement distribution can be made to simplify the mechanics of the data analysis, or a large number of samples can be used to approximate actual distributions. Feature extraction and pattern clustering techniques can also be used to simplify the discrimination task.

One way to obtain sample data for IC test measurements is simply to fabricate ICs. This is clearly the most accurate way to characterize process dependent performance variations, but it is also an expensive alternative. Monte Carlo simulation of process and device parameter variation is more economical provided circuit simulation requirements do not exceed the capacity of available computational resources. This approach will be less accurate since process and device model limitations may not fully reflect actual circuit performance.

Given a sample of noisy test data, one approach is to assume some measurement distribution, then use sample moments to estimate the joint probability density functions (jpdf) of operational and faulted test results. A threshold discriminant can then be employed as a diagnostic. This method suffers from the inaccuracy of the distribution assumption as well as from the need for some separate feature extraction technique to decide which measurements are useful. Even if fabrication process disturbances are normally distributed, the nonlinear relationship between process variation and measurement means that the measurement data cannot be expected to be distributed in some easily predictable manner. The strongest a priori assumption that can be justifiably made about measurement distribution due to process variation is that it is probably unimodal and asymmetric [11]. Monte Carlo generation of a large number of samples better approximates measurement jpdfs [1], but still suffers from the need for separate measurement feature extraction.

Feature extraction involves the selection of measurement combinations which provide for a more efficient representation of the raw data. A more efficient representation emphasizes combinations which exhibit better discrimination properties. Feature extraction implies data preprocessing which reduces measurement dimensionality as well as clustering

similar measurements. Previously used algebraic methods [11] require the manipulation of large matrices when there are many measurements and provide no solution guarantee. Sometimes human inspection of scatter plots is used to discover correlations between sample data measurements [1] and reduce the number of required measurements.

3 Statistical Properties of Feedforward Neural Networks

A multi-layer feedforward network of the kind currently popular in the neural network literature can be viewed as a statistical pattern recognition algorithm. When trained on random sample data, neural network connection weights effectively form a vector-valued statistic of that data [12]. Feedforward neural networks also serve as universal vector function approximators provided sufficiently many hidden units are available [2] [4]. These properties suggest that a feedforward network can be used to approximate a discriminant function based upon random sample data without the need for a priori knowledge of the sample distribution or an excessive number of samples.

The popular backpropagation training method [9] arrives at feedforward connection weights in a fashion which encourages the automatic extraction of important data features. The gradient descent algorithm associated with backpropagation strengthens connections which contribute to the reduction of error in the approximation of the sample mapping. A feedforward network trained this way will emphasize input combinations which contribute the most to a good approximation, automatically performing feature extraction without the need for data preprocessing. Clustering of similar inputs and dimensionality reduction can both be observed to occur automatically. These characteristics help eliminate the need for cumbersome scatter plot inspection and numerically unwieldy algebraic data preprocessing.

4 Neural Network Based IC Diagnostic Synthesis

The IC diagnostics which we are investigating take advantage of the properties associated with feedforward neural networks. Monte Carlo simulation is used to generate a sample data set modeling ideal test conditions in the presence of process and measurement noise. Part of this sample data set is used to train a feedforward neural network using the backpropagation algorithm. The resulting connection weights define a discriminant function which is then tested for fault coverage performance using the remaining portion of the sample data set. Once an acceptable level of coverage is determined, the connection weights are available for transfer to the automatic test equipment.

The main advantages of this approach are that no measurement distribution assumption is needed to form a discriminant and that features are automatically extracted without complex numerical manipulations. The principle disadvantage is that iterative gradient descent training techniques like backpropagation are subject to solution convergence difficulties which can lead to excessive training times and nonoptimal solutions. It is notable

that contemporary research toward finding ways to overcome such difficulties is in progress.

Recent work in both electronic circuit test and automobile diagnostics lends additional support to this approach. Neural networks have been demonstrated which approximate the relationship between a node voltage measurement space and a six resistor circuit element space, with the goal of detecting out-of-tolerance device parameters [10]. They have also been used to discriminate between automobile engine faults given control CPU signals [6]. Our work includes an additional dimension to these previous results by specifically incorporating the effects of production variations and measurement noise.

Figure 1 shows a diagnostic generation system in its intended context within an overall IC test design strategy. The diagnostic generator combines circuit layout and test specifications with process specifications to generate a diagnostic decision rule. The test specification is expressed as a fault dictionary which relates ideal stimuli and responses to various fault conditions for a given circuit specification. The diagnostic is expressed in terms of a specific feedforward neural network configuration and its associated connection weights. The diagnostic processor corresponds to the hardware which executes the neural network algorithm in conjunction with automatic test equipment. The diagnostic generator also provides some confidence measure which indicates fault coverage. This can help guide potential revisions of the test specification or even the tested circuit.

The internals of the diagnostic generator are shown in Figure 2. Process specifications are translated into a device characteristic sample via Monte Carlo process simulation. The layout specification is translated to a netlist by a circuit extractor, and both are used as input to a circuit simulator. The test stimulus completes the specification of a circuit simulation which when executed, provides a random sample of test results. This sample simulates the measurements which would be made on a batch of fabricated ICs. A measurement simulation then distorts the fabricated response sample, modeling the imperfections of the targeted test equipment. The resulting simulated response sample is then combined with additional information from the original test specification during the training of the pattern recognition algorithms. The results of this training process are then made available to the designer in the way of a diagnostic decision rule and feedback regarding its effectiveness.

A fabrication process simulator has been implemented using the SUPREM-III process simulator [3] and the PISCES-II device characteristic extractor [8] configured with a Monte Carlo process parameters generator. The FABRICS fabrication process simulator [7] is another tool available for implementing the statistical simulation of the fabrication process. In our experimental setup, we are using various Berkeley tools for circuit specification with PSPICE and MCNC CAZM as our circuit simulators. The specific choice of process and circuit simulators is relatively independent of our diagnostic synthesis goal, and is considered a matter of designer preference.

The measured response sample obtained from these simulation steps is then used as input to a feedforward neural network training algorithm. The sample is partitioned into two smaller pieces: one for training and one for evaluation. The network is trained on equal numbers of faulty and fault-free exemplars from one of these sets. The quality of the acquired discriminant is then tested using the previously unseen sample. If a sufficient

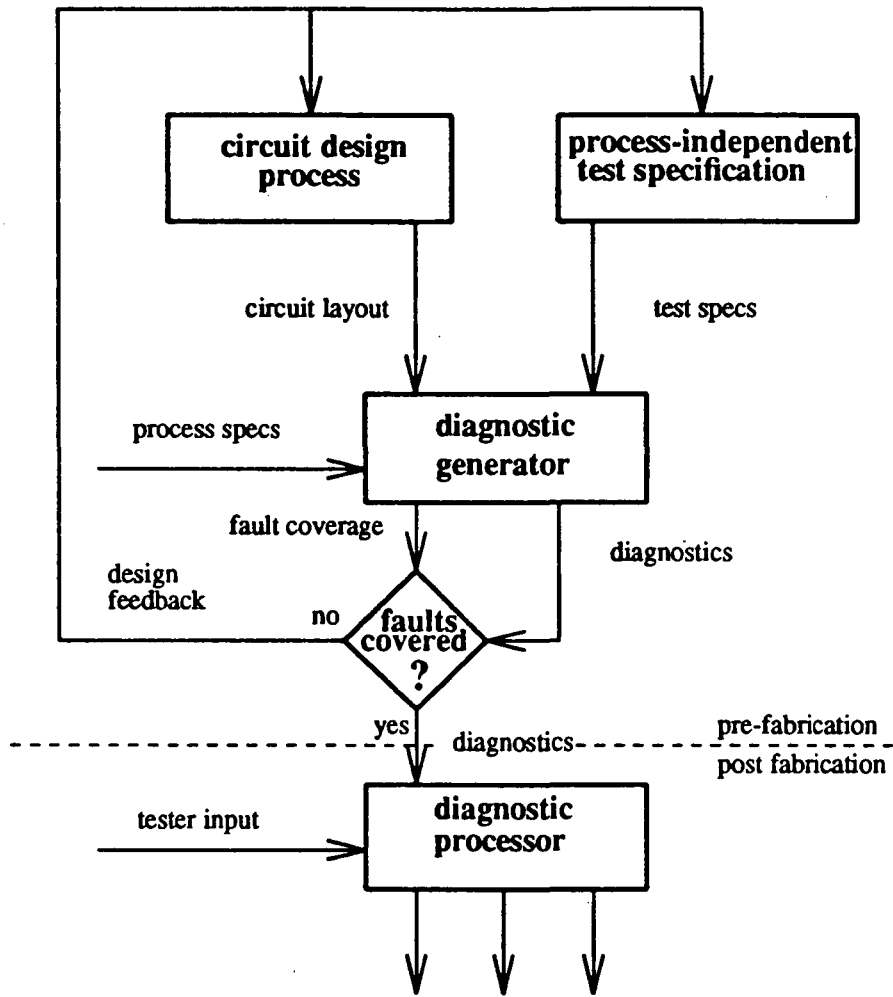


Figure 1: Diagnostic synthesis in a test design process

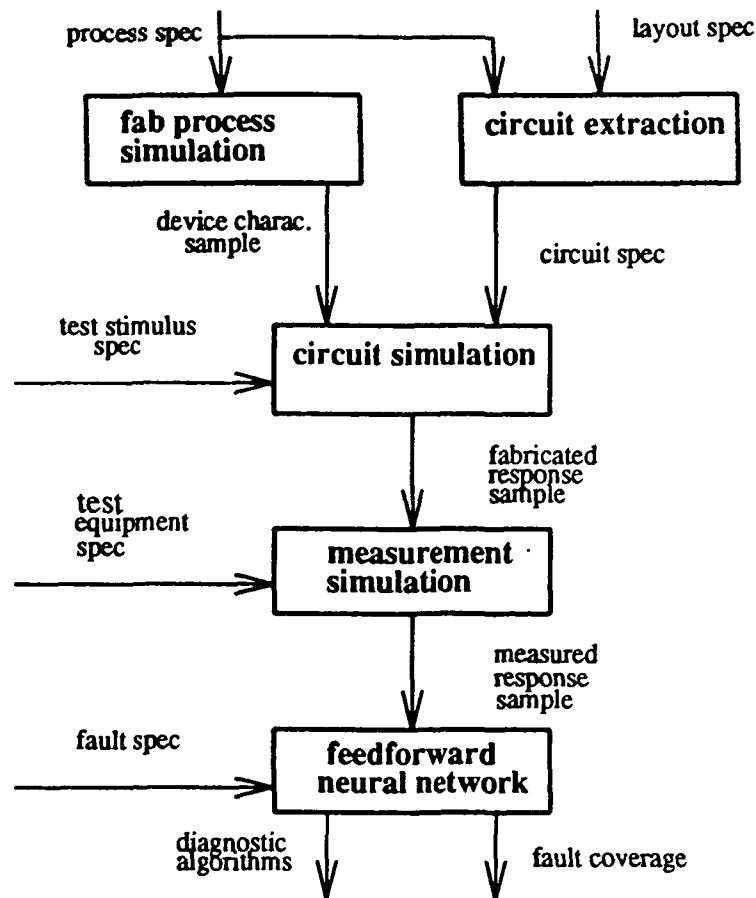


Figure 2: Neural network based diagnostic synthesis method

percentage of these test exemplars is properly classified, then the connection weights corresponding to the generated diagnostic are made available for transfer to the automatic test equipment. A poor discriminant can arise for many reasons however, ranging from an inappropriate test specification to an overly constrained network architecture. We are currently investigating variations of the backpropagation training algorithm which automatically add hidden units as a function of convergence rate. It is expected that a poor discriminant is less likely to result from an insufficient network architecture using such an algorithm.

5 Experimental Approach and the Results

The quality of the pattern classification performance strongly depends on the number of training samples the network is exposed to during the learning stage and how closely the training patterns resemble the actual data with which the network will be confronted during normal operation. Therefore, it is essential to find efficient techniques for fault simulation. In general, fault simulation can be carried out either in a real IC fabrication process, or using computer simulation. The first method has two severe drawbacks:

1. Such experiments are expensive and time-consuming.

- The disturbances introduced in the fabrication process cannot be controlled with sufficient accuracy.

We propose using a statistical process simulator SUPREM-III, a semiconductor device modeling program PISCES-II, and a circuit simulator, e.g., SPICE or CAZM for fault simulation. The relationship among SUPREM-III, PISCES-II, and SPICE or CAZM are as shown in Figure 3. SUPREM-III takes care of the process parameters, the layout parameters, and the process disturbances. The FAB process simulation outputs come from SUPREM-III are fed to PISCES-II for electrical characteristics analysis. The output of PISCES-II is then fed to SPICE or CAZM through an interface software called PICA. PICA is currently being designed at Washington State University. The circuit performances output from SPICE or CAZM will be used for pattern classifications using neural networks.

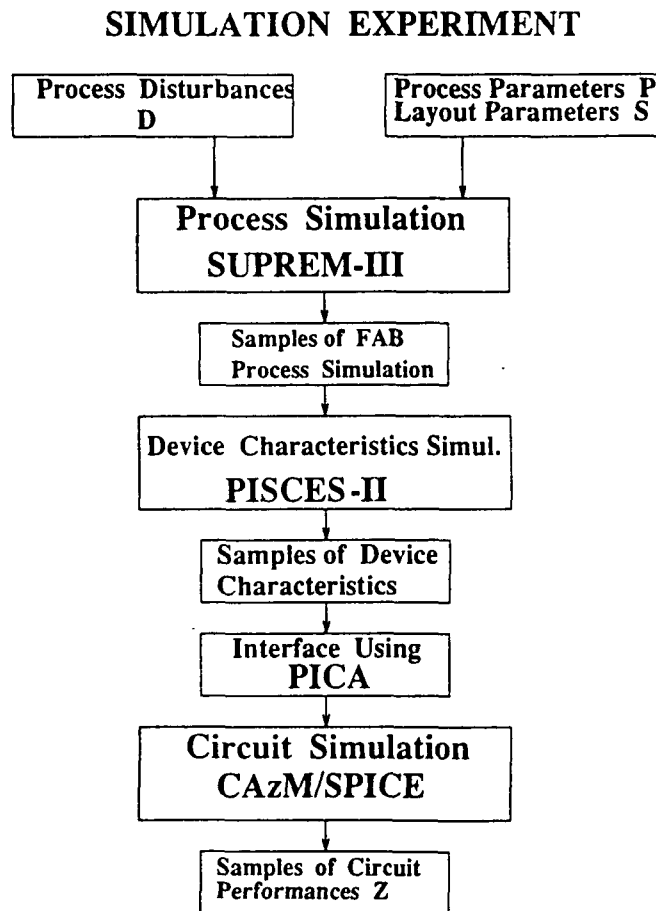


Figure 3: The simulation experiment flow chart for IC tests

The Monte Carlo Method [5] will be used to generate data which resemble the process disturbances. Monte Carlo method is one that involves deliberate use of random numbers in a calculation that has the structure of a stochastic process. By stochastic process we mean

a sequence of states whose evolution is determined by random events. In a computer, these are generated by random numbers. This particular experiment consists of the following steps: creating the fault dictionary using PSPICE, preprocessing the data of the fault dictionary, selecting the neural network architecture and training algorithm, and finally training the neural network for classifying IC failures.

An operational amplifier, shown in Figure 4 and 5, was used in our experiment. Two fault dictionaries of transient analysis were created by using PSPICE with Monte Carlo method. A normal data set was collected by varying device parameters within operational limits. While a faulty data set was collected by making the variation of the junction depth of one of the MOSFETs (M1 in Figure 5) exceeding the functional specification. Input stimulus is a 0.1 volt pulse of 5 micro seconds duration. Each data set has sixty patterns. Each pattern has 101 sampling points in 10 micro seconds. One of our experiments chose 32 samples from each pattern with equal spaces. The other one sparsely chose 11 samples and also evenly spaced. Data in the fault dictionary were normalized to be in the range between 0.1 and 0.9. Figure 6 shows these densely sampled signal waveforms in four sets a, b, c and d.

Using the backpropagation learning algorithm, a two-layer feedforward neural network (one hidden layer) was then trained as pattern classifiers for those data. For the data with 32 samples, we used a 32:5:1 network. For the one with 11 samples, we used 11:5:1 network. Data set used for training has 30 patterns each from both the normal data set and the faulty data set. After the network has been trained, all four data sets were used for testing. The 32:5:1 neural network can classify all 120 patterns after 15000 epochs of training. The 11:5:1 neural network, trained with sparsely sampled data, cannot finish the training phase after 73000 epochs.

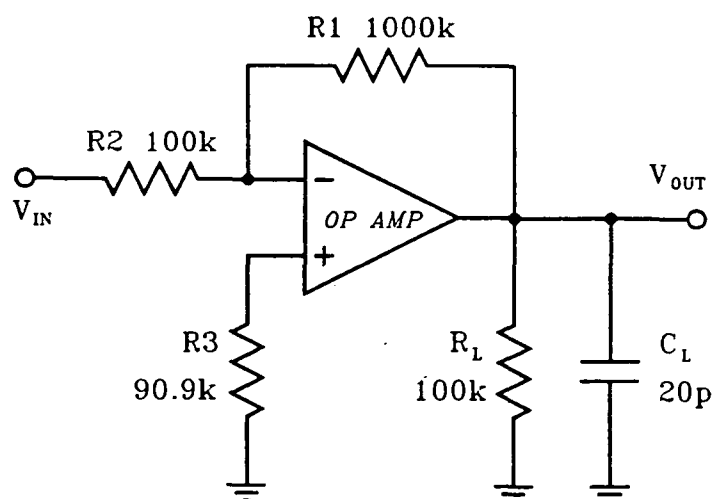


Figure 4: Configuration for SPICE simulation. The OP AMP is shown in Figure 5.

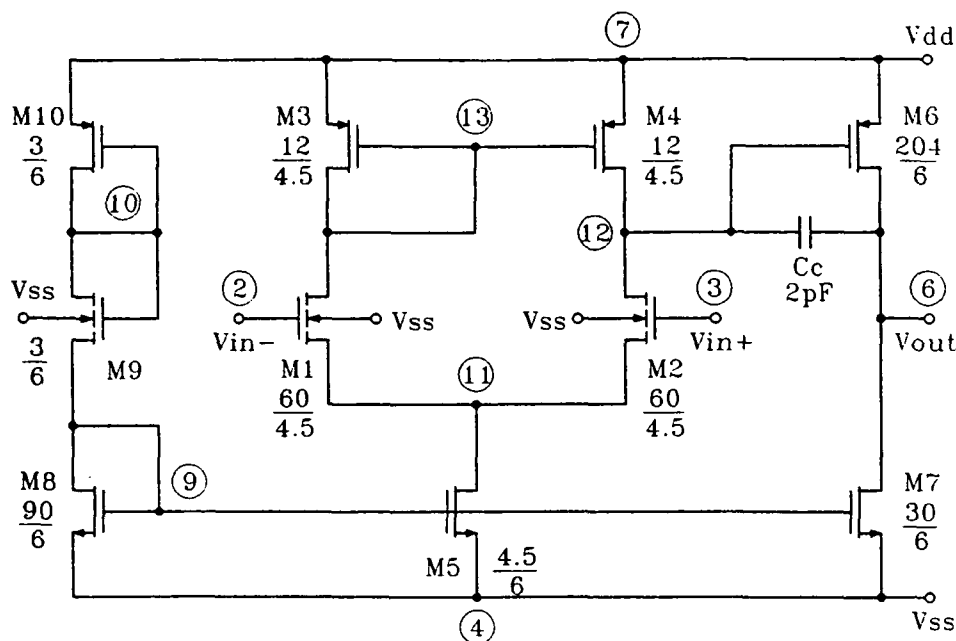


Figure 5: CMOS operational amplifier schematic diagram

6 Summary and Future Work

The possibility of using neural network in IC fault diagnosis problem is demonstrated. Results of the experiment positively showed the capability of the feedforward network in separating the faulty circuit from the normal circuit based on the patterns presented. However, its diagnosis ability depends on the information implicitly in the patterns used in training the network. In the case of sparse sampling, where output signal sampled eleven times, it is unable to train the network to diagnosis the fault from the pattern presented. It is because the essential features of the signals are not presented to the network. Important information is not included in the sparsely sampled pattern. When the output signals were sampled more densely, the network can identify the patterns accurately. Such phenomena is not hard to explain.

Examining the output signals carefully, the distinctive features are in the slew rate and the overshoot of the output signal. Faulty circuit has a slower slew rate and larger overshoot. These distinctive features are not included in the sparsely sampled pattern. Since the output signals of both faulty circuit and normal circuit are very similar in the shape of the waveform. It is hardly to distinguish the signals simply based on their shape. However, when densely sampled, these features present in the pattern used for training and verification. The trained network can positively identify the faulty signals.

Further work will be conducted to investigate the sampling dependent phenomena and to establish techniques to guarantee success in diagnosis using neural network; and also assist how diagnosis measurement should be conducted.

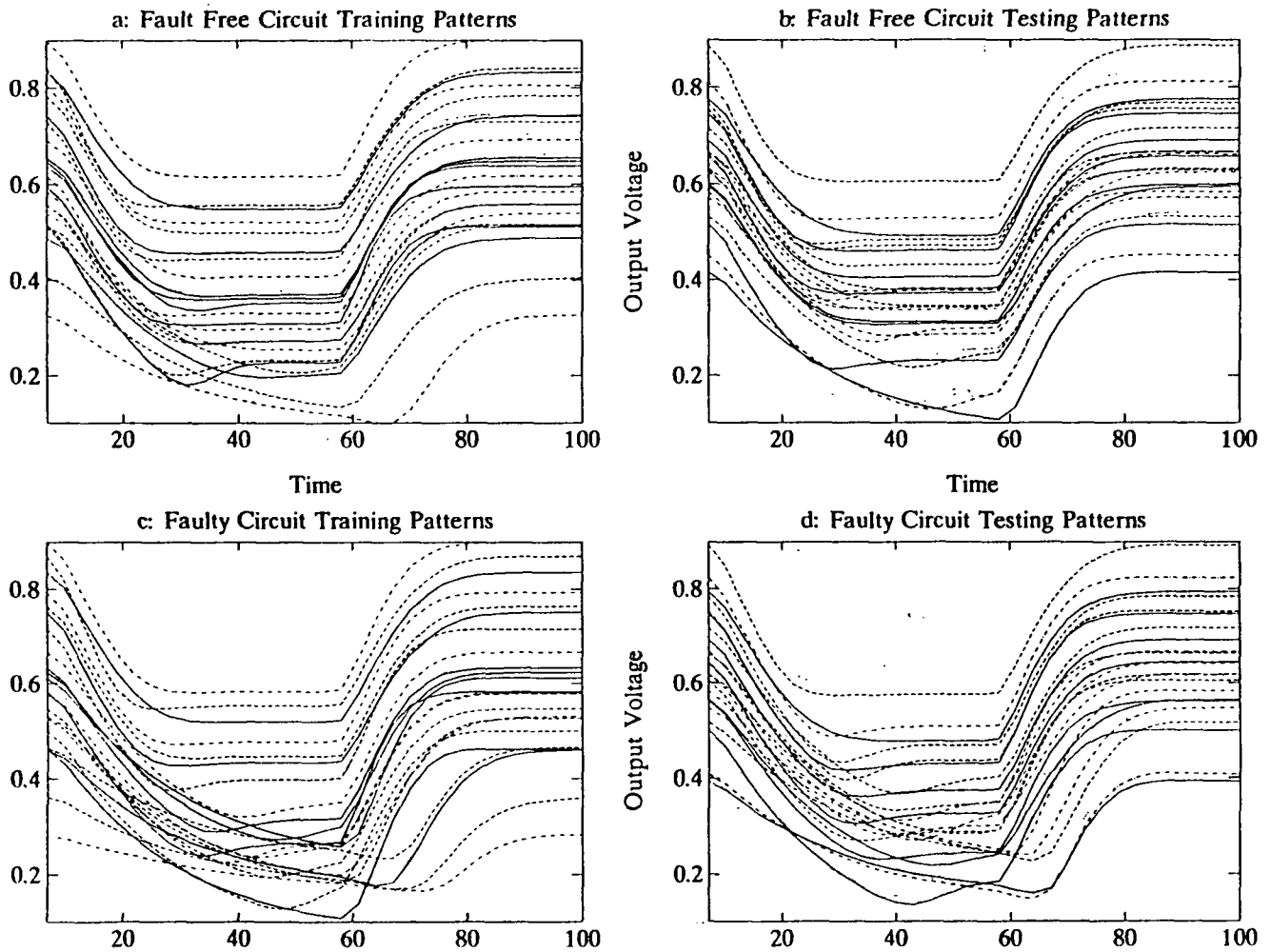


Figure 6: The neural network was trained with patterns in data sets a and c. The properly trained neural network can positively classify any patterns in all four sets. Note that the output voltage and time are normalized.

References

- [1] Brockman, J., and Director, S., "Predictive Subset Testing: Optimizing IC Parametric Performance Testing for Quality, Cost, and Yield," *IEEE Trans. on Semiconductor Manufacturing*, Vol. 2, No. 3, August 1989.
- [2] Funahashi, K. "The Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, Vol. 2, pp. 183-192, 1989.
- [3] Hansen, S., "SUPREM-III User's Manual," Tech. Rep. 8628, Stanford University, 1986.
- [4] Hornik, K., M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, pp. 359-366, 1989.
- [5] Kalos, M. H. and Whitlock, P. A., *Monte Carlo Methods Volume I: Basics*, John Wiley & Sons, 1986
- [6] K.A. Marko, L.A. Feldcamp and G.V. Puskorius, "Automotive Diagnostics Using Trainable Classifiers: Statistical Testing and Paradigm Selection," *Proc. Summer 1990 IEEE IJCNN*, San Diego, CA, pp. I-33, I-38, 1990.
- [7] S.R. Nassif, A.J. Strojwas and S.W. Director, "A Methodology for Worst-Case Analysis of Integrated Circuits," *IEEE Trans. on CAD*, Vol. 5, No. 1, pp. 104-112, 1986.
- [8] Pinto, M., et al., *PISCES-II Technical Report*, Stanford University, 1985.
- [9] Rumelhart, D., G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, Vol. 1, pp. 318-362, MIT Press, 1986.
- [10] J.A. Starzyk, M.A. El-Gamal, "Artificial Neural Network for Testing Analog Circuits," *Proc. 1990 IEEE ISCAS*, New Orleans, LA, pp. 1851-1854, 1990.
- [11] Strojwas, Andrzej J., "Pattern Recognition Based Methods for IC Failure Analysis," Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh PA, 1982.
- [12] White, H., "Learning in Artificial Neural Networks: A Statistical Perspective," *Neural Computation*, Vol. 1, No. 4, 1989.