

Hierarchical Approach to Diagnosis of Electronic Circuits Using ANNs

Miona Andrejević Stošović and Vančo Litovski

Abstract— In this paper, we apply artificial neural networks (ANNs) to the diagnosis of a mixed-mode electronic circuit. In order to tackle the circuit complexity and to reduce the number of test points hierarchical approach to the diagnosis generation was implemented with two levels of decision: the system level and the circuit level. For every level, using the simulation-before-test (SBT) approach, fault dictionary was created first, containing data relating the fault code and the circuit response for a given input signal. Also, hyper-computing was implemented, i.e. we used parallel simulation of large number of replicas of the original circuit with faults inserted to achieve fast creation of the fault dictionary. ANNs were used to model the fault dictionaries. At the topmost level, the fault dictionary was split into parts simplifying the implementation of the concept. During the learning phase, the ANNs were considered as an approximation algorithm to capture the mapping enclosed within the fault dictionary. Later on, in the diagnostic phase, the ANNs were used as an algorithm for searching the fault dictionary. A voting system was created at the topmost level in order to distinguish which ANN output is to be accepted as the final diagnostic statement. The approach was tested on an example of an analog-to-digital converter.

Index Terms— Fault diagnosis, Hierarchical systems, Neural networks, Parallel simulation.

I. INTRODUCTION

WHenever we think about why something does not behave as it should, we are starting the process of diagnosis. Diagnosis is therefore a common activity in our everyday lives [1]. Every system is liable to faults or failures. In most general terms, a fault is every change in a system that prevents it from operating in proper manner. We define diagnosis as the task of identifying the cause and location of a fault manifested by some observed behavior. This is often considered to be a two-stage process: first the fact that fault has occurred must be recognized – this is referred to as *fault detection*. Secondly, the *nature and location* should be determined such that appropriate remedial action may be initiated.

The general structure of a diagnostic system is shown in Fig. 1. Signals $u(t)$ and $y(t)$ are input and output to the system, here denoted as the “Process”, respectively. Faults and disturbances (in our considerations measurement errors)

also influence the system under test but there is no information about the values of these errors. The task of the diagnostic system is to generate a diagnostic statement S , which contains information about fault modes that can explain the behavior of the Process. Note that the diagnostic system is assumed to be passive i.e. it cannot affect the Process itself. The whole diagnostic system can be divided into smaller parts referred here to as tests. These tests are also diagnostic systems, DS_i ($i = 1, \dots, n$). It is assumed that each of them generates diagnostic statement (or hypothesis) S_i ($i = 1, \dots, n$). The purpose of the decision logic (voting system) is then to combine this information in order to form the final diagnostic statement S . Modern automatic test pattern generator may support such concepts [2].

The number of possible faults in an electronic system may be large and a fault can be located everywhere in the system. In order to diagnose in such conditions we adopted a hierarchical approach where successive diagnostic statements are generated as the level of description of the system is lowered going down towards the fault itself [3], [4]. This allows for smaller sets of faults to be considered at a time at a given hierarchical level.

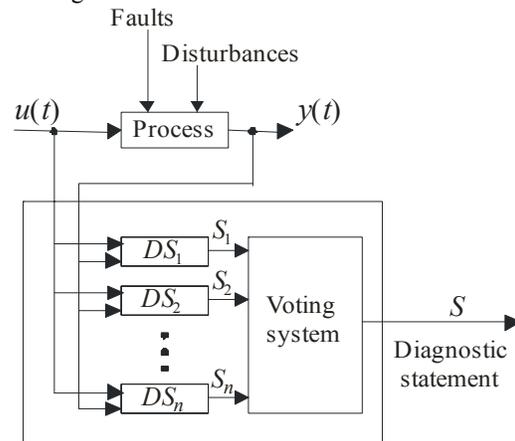


Fig. 1. A general diagnostic system.

After shortly reviewing the existing concepts of diagnosis, in the next, we will consider the specifics of diagnosis of mixed-signal electronic circuits. Hierarchical concept will be applied to mixed-mode electronic circuits diagnosis. Also, we will present the modern aspects of implementation of parallel computing to electronic circuit simulation. Specifically, supercomputing and grid computing will be elaborated. An example will be given expressing both the nature of the subject and the underlying ideas. Short version of this paper was presented in NEUREL 2010 [5].

Manuscript received November 22, 2010.

Miona Andrejević Stošović and Vančo Litovski are with the Faculty of Electronic Engineering, University of Niš, Serbia, Phone: +381 18 529 321, e-mail: (miona.andrejevic, vanco.litovski)@elfak.ni.ac.rs.

II. CONCEPTS OF DIAGNOSIS

Besides the human expert that is performing the diagnosis, one needs tools that will help, and ideally, perform the diagnosis automatically. Such tools are a great challenge to design engineers because, usually, the diagnostic problem is underspecified. In addition, it is a deductive process with one set of data creating, in general, unlimited number of hypotheses among which we try to find a solution. This is why the research community continues to be attracted by this problem [6].

Thanks to the advances in computational intelligence in the last decades new diagnostic paradigms have been applied based on: model-based concepts [1]; production rule based artificial intelligence [7]; ANNs [8]; genetic algorithms [9]; and fuzzy-reasoning [10]; all trying to create an approach that exhibits properties that we might consider to be “intelligent behavior”. A comprehensive overview of the complete subject of diagnosis of analog electronic circuit may be found in [11].

In order to get an idea of why and how ANNs are applied to mixed-mode electronic circuit diagnosis, the application of the diagnostic concept (Fig. 1) will be elaborated in some detail first. It involves collaboration of design, test, and field engineers and the mutual distribution of responsibilities throughout the life cycle of an electronic product. We assume that field engineers are expected to react after a functional failure of the system. In order to diagnose such a system they need to be supplied with: testing equipment, a list of specific measurements to be done (including a set of signals and test points), and diagnostic software to process the measurement data. A similar set of data and tools would be given to a test engineer in a production-plant environment in order to evaluate the production yield and create feedback to process engineers when prototyping the circuit.

We believe, however, design engineers are the most familiar with the product and the most qualified and capable to synthesize test and diagnostic signals, and procedures. The importance of that comes especially in fore when mass produced systems are to be diagnosed before shipping to the customers. This means the simulation-before-test (SBT) approach has to be applied to create fault dictionaries containing exhaustive lists of faults and corresponding responses. The fault dictionary is in fact a table representing the mapping from the fault list into a list of faulty (or possibly, fault-free) responses. In that way the diagnostic process becomes a search through the fault dictionary. Alternatively, modern diagnostic techniques using traditional artificial intelligence and reasoning methods typically fall into the simulation after test (SAT) category. This will increase the time spent on diagnosing the system at production time [12]. SBT systems typically require more initial computational costs, but provide faster diagnosis at production time being additional reason why this concept was accepted here.

We claim here that ANNs, being universal approximators [13], are the best way both to capture the mapping, and to search through the dictionary, thereby to perform diagnosis.

If large number of faults and reduced number of outputs are to be conceived in the same time, thanks to the resemblance of the fault effects, the search process within the fault dictionary requires highly sophisticated decision making algorithm. We will show in the next how ANNs can perform successfully in most difficult conditions.

III. DIAGNOSIS OF MIXED-MODE CIRCUIT

The explosion of integrated circuit technology has brought with it some difficult testing problems. The recent growth of mixed analogue and digital circuits complicates the testing problem even further. It becomes more complicated to determine a set of input test signals and output measurements that will provide high degree of fault coverage. There is also a timing problem when testing such circuits even on the fastest automated equipment.

Analogue electronic circuits are known to be difficult to test and diagnose. Apart from the huge number of possible faults, this difficulty is a consequence of the inherent nonlinearity of this circuit category. Even linear circuits (having linear input-output signal interdependence) exhibit non-linear relations between circuit-parameter values and the output response. There are no linear active networks. Active networks are non-linear with non-linear reactive elements. They may be linearized and thought of as such in situations where signal and parameter changes are small in comparison to nominal values. When large parameter changes or even catastrophic faults occur (affecting the DC quiescent state), however, one must distinguish between linear and analogue circuits. This, unfortunately, is not the case in most research reports bringing confusion into the subject.

A specific aspect of diagnosis is the number and location of the test points. Simply, we can say that internal test points should be avoided and measurements on the primary inputs and outputs are preferred. This is not only related to their automatic accessibility but also to the nature of the diagnostic reasoning. Namely, one looks for functionality in order to start diagnosing, the function being seen at the primary terminals. Of course, in order to compensate for the reduced number of test points additional measurements with different types of applied signals may be needed to extract complete information about the system behavior. For complex analogue systems, however, hierarchical approaches based on decomposition [3], [4], [6], [14], [15] are inevitable provided that no propagation of the fault effect arises between partitions. That is not easy to achieve. Of course, there are circuits that may be partitioned based on functionality known *a priori* from the design process as mentioned in the introduction.

In this paper we describe the results of applying feed-forward ANNs to the diagnosis of non-linear dynamic electronic circuits that are mixed with digital ones with no restriction to the number and type of faults. This method is based on fault dictionary creation and using an ANN for data compression by memorizing the table representing the fault dictionary. The ANN created in this way is, consequently, used for diagnosis by applying to its inputs the signals obtained by measurement of faulty network.

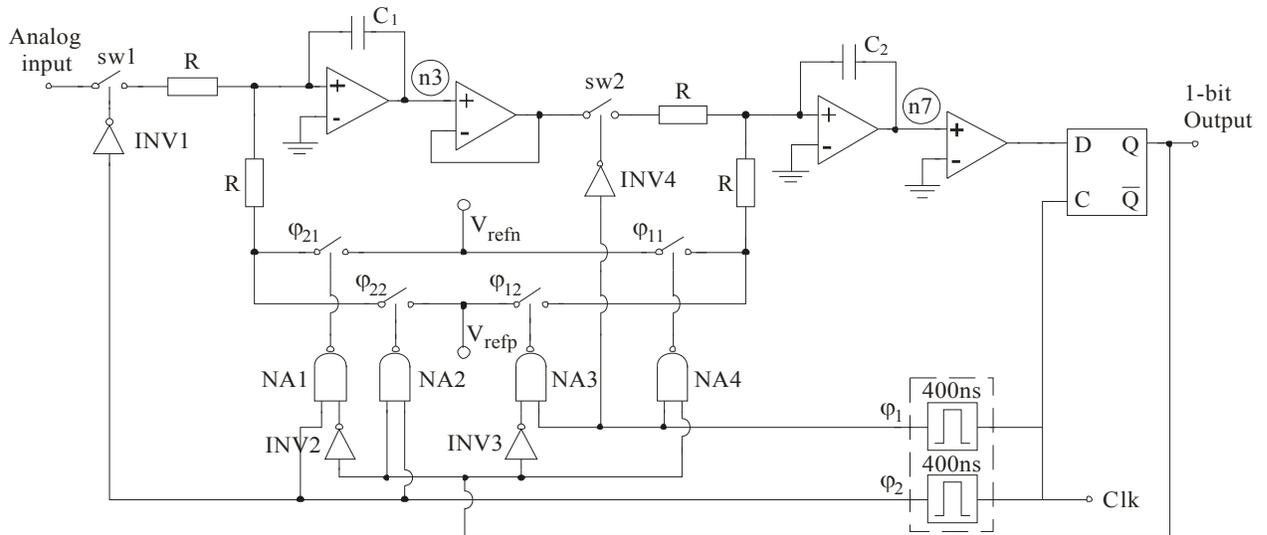


Fig. 2 Sigma-delta modulator structure.

This process may be considered as looking-up for a fault in the fault dictionary. The ANN finds the most probable *fault code* that corresponds to the measured signals. The procedure was applied to analog circuits and illustrated in [11].

Putting this in the general context of diagnosis we first note that the fault dictionary contains all the knowledge we need. In other words by applying the SBT concept all hypotheses are memorized (within the ANN) and no further hypothesis needs to be created after the dictionary is known. This is equivalent to the structural concept of testing. The fault not conceived in advance can't be tested nor diagnosed. Now we look among the hypotheses (by searching the dictionary i.e. by running the ANN) to find the one most similar to the actual (faulty) circuit response. The difficulties here are the complexity of the search and the decision algorithm that finds the "most similar" entry in the dictionary. As it will be shown by an example this can be an extremely difficult task that has been successfully solved using ANNs.

For a mixed signal system such the one depicted in Fig. 2, we are faced with additional difficulties related to the different nature of the responses sought at different nodes. In order to tackle that problem the fault dictionary created at the system level was partitioned in two parts enabling implementation of the concept described in Fig. 1.

Considering the overall efficiency of the process the only "bottle-neck" of the procedure is the long simulation time necessary to create the fault dictionary having in mind the enormous number of possible faults and the necessity for complete time domain simulation of a new replica of the original circuit with a fault inserted. To tackle this problem we implemented parallel simulation in which every faulty version of the circuit is simulated by separate processor in a supercomputer so enabling a considerable speed-up of the fault dictionary creation phase of the diagnostic process [16].

The ANNs used for this diagnostic example are the well

known feed-forward neural networks structured in three layers. They have only one hidden layer, which has been proved sufficient for this kind of applications i.e. approximation [17]. The neurons in the hidden layer are activated by a sigmoid (logistic) function, while the neurons in the output layer use linear activation function. The learning algorithm used for training this network is a version of the steepest-descent minimization algorithm [18].

IV. PARALLEL CIRCUIT SIMULATION

There are mainly two methods for parallel implementation of circuit simulation [19]. The first one implements parallel threads within the implementation of the simulation algorithm. In early days they were created by partitioning the circuit into smaller parts by node or branch tearing [20], [21]. Nowadays, the circuit matrix is created in parallel [22]. The main limitation of this approach is the communication overhead that puts an obstacle of the maximum speed-up of the simulation in this way: "The maximum speed-up obtained with parallel system of N processors that has α % of communication overhead can be no greater than $1/(1-\alpha \cdot 100)$ ".

Accordingly if α is 0.1, the maximum speed-up is 10. In the last decade, the possible speedup degraded due to the nearly constant communication speed and increasing computation performance structure [23]. The second approach uses each core for computing exactly one simulation model, which is also known as Hyper Computing [19]. Large number of cores is used for calculating the models N -times, e.g. by applying different random number seeds in Monte-Carlo simulation [24]. Speeding-up of such computations is nearly equal to the number of the cores and could be guaranteed in practice. For statistical correctness, however, over 20 or more simulation runs must be executed for getting significant results.

In this paper we implement the hyper computing for fault dictionary creation. In fact, there is no fundamental

difference between Monte-Carlo simulation and fault dictionary creation except for the origin of the variations of the parameters. In Monte-Carlo one creates the parameter variation according to a set of given statistical distributions while in fault dictionary creation one takes the faults from a fault list created within the production foundry.

V. SIMULATIONS ON GRID

The development of low-cost personal computers and gigabit LAN network connections offers a possibility for implementation of inexpensive distributed multiprocessor systems such as computer clusters. A cluster has many advantages over classic supercomputer: it is inexpensive, flexible, easy to use, easy for maintenance and highly stackable. One particular implementation of this approach, involving open source system software and dedicated networks, has acquired the name “Beowulf” [25].

The growth of Internet and WAN links of great capacity and speed led to development of the computational Grid, Fig. 3. In the same way as power grid provides electrical power, computational power can be obtained on demand from a network of providers, potentially belonging to the entire Internet. The Grid is a highly heterogeneous and geographically distributed computing system consisting of interconnected shared computer resources (computer clusters) that users can utilize for their demanding tasks. At the beginning, this paradigm has been strictly scientific and academic; but as in the case of the Internet, it became widely accepted and popular. One of the most common definitions says that a computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities enabling on-demand access to computing, data, and services [26]. Grid computing is suitable for intensive calculations that require significant processing power, large operating memory, as well as storage capacity. The simulation of ICs is paradigmatic example of such calculations [27].

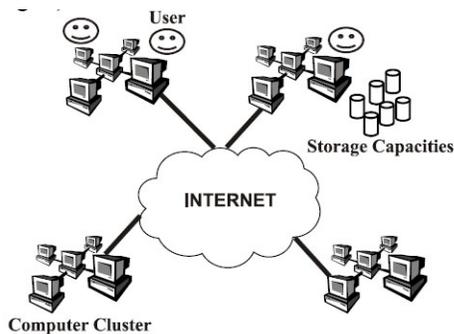


Fig. 3. Grid structure

Fig. 4 shows the structure of a Grid application. In order to enable a designer to run parallel simulations on the Grid resources, it is necessary to develop appropriate Grid interface for the simulator. Such interface should provide submission of simulation jobs together with simulation

models, simulation run on a distant computer cluster and retrieval of simulation results.

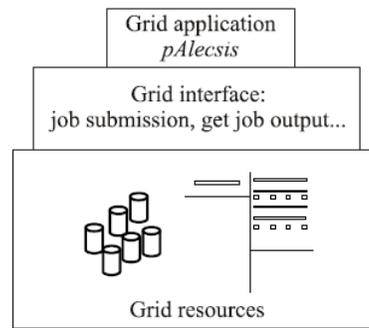


Fig. 4. Basic structure of a Grid application

In the example given here, simulations are executed on the parallel computer structured as a Beowulf cluster, Table I. It is part of the SEE-Grid initiative [28] and is capable to use resources from within the initiative.

TABLE I
BEOWULF CLUSTER STRUCTURE

Component	Specification
8 × 2 quad-core Intel Xeon E5420	2.5GHz, 4GB RAM, 250GB HDD
1.4TB RAID5	NAS (Network attached storage)
LAN	dual 1Gbit Ethernet

VI. FAULT DICTIONARY CREATION AND APPLICATION EXAMPLE

In order to describe the way in which the fault dictionary was created, the sigma-delta modulator circuit depicted in Fig. 2 was used. It is a mixed-signal system with representative functional complexity having analogue, digital and switching elements. The switches in the circuit are modeled as truly ideal, exhibiting zero and infinite resistance for closed and open state, respectively.

We consider in this paper defects in the whole circuit, meaning in analog, digital, and the switching part. We do not intend to diagnose multiple faults.

There are two types of defects in the digital part of the system observed: catastrophic (stuck-at) and delay faults (delays of rising and falling edge of digital signals).

In the system of Fig. 2, the analogue switches are controlled by digital signals, so there are pairs with the same fault effects, i.e. the effect is the same when the switch is stuck at ON (OFF) and when the logic circuit's output is “stuck-at-1” (“stuck-at-0”). So, we will consider hard faults (that are associated to the analogue part of the circuit) as stucked switches [29].

Having in mind that the clock period in the system is 1.2 μs (half period is 600 ns), we examined effects of delays not greater than 400 ns. In fact, effects of rising edge delay are simulated for delay values of: 100 ns, 250 ns, 400 ns, while for the falling edge, we inserted smaller values: 50 ns, 100 ns, 150 ns. The goal was to determine the mapping of the

delay faults onto the output digital signal. All digital gates were examined (4 inverters and 4 nand circuits). Simulations were performed using Alecsis [30] simulator.

The fault dictionaries for both analog and digital part of the system were created using the response of the circuit to an input ramp signal (Fig. 5a). The system output value was registered after every clock period (Fig. 5b), so these output digital values form the output signature (Fig. 5c). These are then represented in more compact hexadecimal form (Fig. 5d). We performed simultaneous simulation on the Grid of these faulty mixed-mode electronic circuits in time domain. The faulty circuits defer only in one parameter/defect, so these gathered results formed the fault dictionary.

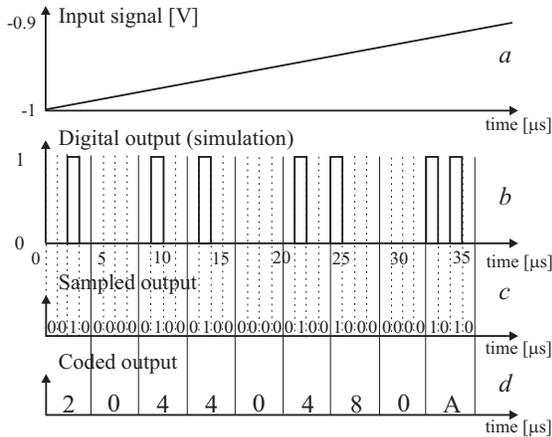


Fig. 5. Fault dictionary creation – response of the fault-free system

TABLE II
PART OF THE FAULT DICTIONARY FOR THE ANALOG PART OF THE CIRCUIT

Defect code	Defect type	Signature
0	FF	20440480A
1	C1 disconnected	E38E38E38
3	0.8*C1	102204210
6	1.2*R1	822211110
10	0.8*R4	804809011
12	OA3 output disconnected	805005012

FF stands for the fault free circuit.

In the analog part of the system, we have considered both parametric and catastrophic defects [31], [32]. As parametric faults we considered variations of resistance and capacitance values. The capacitances of both capacitors are changed. The first stage is more sensitive to parameter variations, while the changes in the second stage have reduced effect on the performance, due to noise shaping. That is best seen when the fault effect of the capacitance in the first stage is observed. However, changes of capacitance in the second stage cause exactly the same effect as the fault free circuit. As an illustration, part of the fault dictionary for the analog part of the system is created as shown in Table II.

Catastrophic (hard) faults in an analogue system change the circuit topology. In order to illustrate this, we have observed the situation when the feed-back capacitor of the operational amplifier is disconnected, and also the situation when there is an open circuit at the operational amplifier's

output (in the example given here, the output of the third operational amplifier-OA3 is disconnected, node *n7* in Fig.2).

The fault coding (column 1 in Table II) is an important issue. In fact, some defects exhibit very similar effects at the circuit output. So, input data (signatures) to the diagnostic system can have very close numerical values. Consequently, if the output values (defect codes) were also similar, the difficulties may arise during the network training. Faults are coded randomly, so that faults with similar effects are unlikely to have similar codes. This approach is proven to be good, because the way of coding influences the training time and error.

The second column of Table II describes the type of the defect. The third column contains the signature seen at the output. Note that, for obtaining the nine-digit hexadecimal number coding the binary output, one has to get 36 samples of the output waveform.

VII. SYNTHESIS OF A HIERARCHICAL DIAGNOSTIC SYSTEM

A two level hierarchical system is depicted in Fig. 6. The idea is to look at: the system and the subsystem (or circuit or component) level. Accordingly, one diagnostic system is to be created at the topmost level the task of which is to locate the faulty subsystem (component) and to deliver enough information (fault code and, possibly, type of hypothesis) to the lower diagnostic level to locate the fault within the subsystem (component). For that, of course, one needs as many diagnostic subsystems at the subsystem level as many circuits are conceived within the system. Generally, such subsystems in the analog part may be operational amplifiers, circuits built of passive components and operational amplifiers (filters, for example) etc. while in the digital part one meets basic logic gates, flip/flops or even registers and memory blocks. It is assumed that fault dictionaries and corresponding ANNs for diagnostic purposes are at disposal and incorporated into the diagnostic software at the time of diagnosis of the whole system.

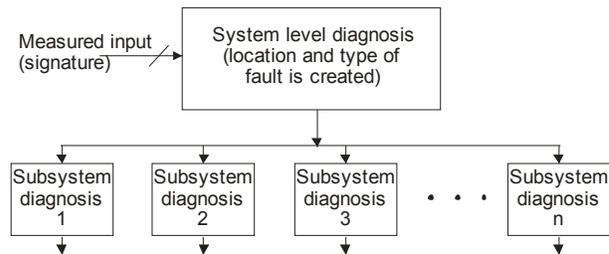


Fig. 6 A two level hierarchical system

At the system level, the modular approach was implemented first making the search for the diagnostic statement easier. The digital and analog part of the system were considered as modules and two artificial neural networks were trained for capturing the look-up tables, one for diagnosis in the digital part, and another for diagnosis in the analog part of the system. Note, the partition is natural from the point of view of creation of fault dictionaries being obtained by simulation successively: firstly for the digital

and later for the analog part. Both networks are feed-forward with one hidden layer. The signatures are inputs to the ANNs, and the fault code is ANN's output to be learned. It means that both neural networks have 9 inputs (one input per hexadecimal digit) and one output terminal. After learning was completed, the number of hidden neurons in the resulting ANN was 10 (Fig. 7), for the network implementing the fault dictionary related to the digital part, and 3, for another, what was found by trial and error after several iterations starting with an estimation based on [33].

The effectiveness of the training process of the obtained ANNs was verified by exciting the ANNs with faulty inputs. Responses of the ANNs show that there were no errors in identifying the faults. Only negligible discrepancies may be observed.

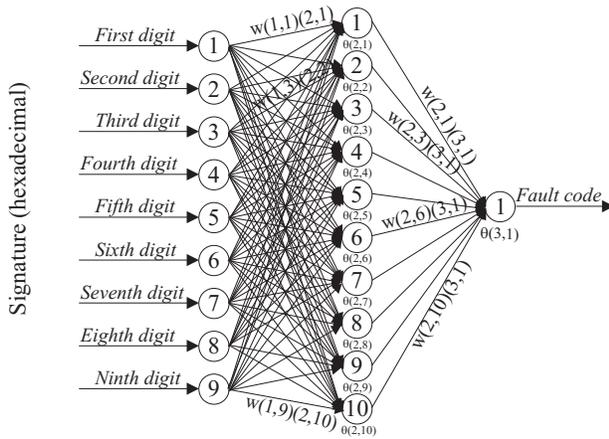


Fig. 7 The structure of one of the diagnostic ANNs

To mention again, two groups of faults were considered in order to reduce the number of faults per ANN so enabling easier learning and reduced complexity of the ANNs. Now, the task is to have a complete diagnosis at system level responding to every signature. The practical implementation of the concept of Fig. 1 is depicted in Fig. 8. *ANN1* diagnoses defects in the digital part of the system and fault codes are in the range from 0 to 45. *ANN2* diagnoses defects in the analog part of the system and fault codes are in the range from 0 to 12. We can notice that we use numbers starting from 0 in both cases in order to denote fault codes. With that notation, when both diagnostic ANNs work in parallel, one can't distinguish whether the fault code refers to analog or digital defect. So, we provided *ANN3* in order to help distinguishing if certain defect is digital or analog.

ANN3 also has 9 inputs and it gets the measured signature as an input as *ANN1* and *ANN2* do. It gets trained so that its output code takes values from the set $\{-1, 0, 1\}$. We refer to these values as to resolution key. Namely, if the defect comes from the digital part, the output code is set to 1, while if it comes from the analog, the output code is set to -1. In the special cases when ambiguity arises, that is when one has the same signature coming from faults belonging to the digital and analog part, we assign 0 to the output of *ANN3*. We will give a few examples now, in order to illustrate the previous explanation.

Suppose that we excite our ANNs with the input signature: $\{0\ 8\ 2\ 2\ 0\ 2\ 2\ 0\ 8\}$. The responses of the three

networks are as follows:

ANN1 response: 30

ANN2 response: -0.0800663

ANN3 response: 0.99934. The resolution key is 1.

The decision logic (Fig. 8) decides that we have digital defect (because the *ANN3* output value is approximately 1), and its code is 30 (because the *ANN1* output value is 30). *ANN2* response is ignored.

Next, we suppose that we excite our diagnostic system with the input signature: $\{8\ 0\ 4\ 4\ 1\ 0\ 4\ 2\ 1\}$. The responses of the three networks are as follows:

ANN1 response: 29.0138

ANN2 response: 4.00001

ANN3 response: -1.00066. The resolution key is -1.

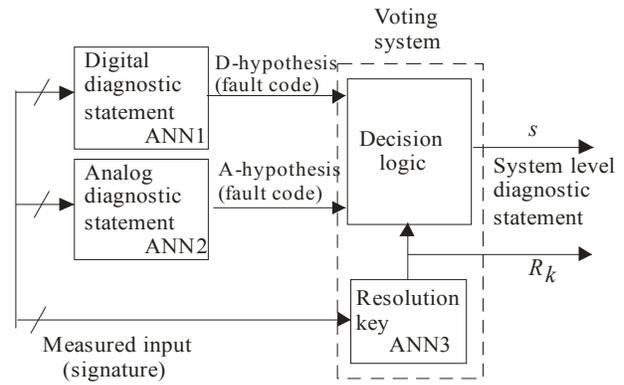


Fig. 8 The ANN based hierarchical diagnostic system

The conclusion is that we have analog defect (because the *ANN3* output value is approximately -1) and its fault code is 4 (because the *ANN2* output value is 4). *ANN1* response is ignored.

Finally, we suppose that we excite our 3 ANNs with the input signature: $\{1\ 0\ 4\ 1\ 0\ 8\ 2\ 1\ 0\}$. The responses of the three networks are as follows:

ANN1 response: 7.99998

ANN2 response: 11

ANN3 response: -0.00172622. The resolution key is 0.

We consider now both *ANN1* and *ANN2* responses because the response of *ANN3* is approximately 0, indicating ambiguity. The conclusion is that we have analog defect with fault code 11, or digital defect coded with 8. We cannot decide which one of them really happened in the system because they have exactly the same response, and this is a problem that may be resolved by increasing the number of sampling intervals or by introducing additional signals for fault dictionary creation. That will be not discussed here anymore.

VIII. CONCLUSION

Artificial neural networks were successfully applied to the diagnosis of the mixed-mode electronic circuit containing analog, digital and a part with internally controlled switches.

The dictionary was separated in two groups. One was related to the faults in the analog part of the circuit while the other was related to the rest of the faults. Simultaneous

simulations of as many electronic circuits as there were defects were performed. Gathered results formed the fault dictionary, and the simulation time was significantly less because the circuits were simulated in parallel, not one by one as it is commonly, so we consider this result very successful. In general, there should not be restrictions on the number of partitions that may be used for diagnosis at any level. In addition, one can introduce as many levels of diagnosis as necessary.

REFERENCES

- [1] R. Benjamins, W. Jansweijer, "Toward a competence theory of diagnosis," *IEEE Expert*, vol. 9(5), pp. 43-52, 1994.
- [2] M. Soma, S. Huynh, J. Zhang, "Hierarchical ATPG for Analog Circuits and Systems," *IEEE Design & Test of Computers*, pp. 72-81, 2001.
- [3] C. K. Ho, F. Eberhardt, W. Tenten, "Hierarchical fault diagnosis of analog integrated circuits," *IEEE Trans. On CAS – II: Analog and Digital Signal Processing*, vol. 48(8), pp. 921-929, 2001.
- [4] H.-T. Sheu, Y.-H. Chang, "Robust fault diagnosis for large-scale analog circuits with measurement noises," *IEEE Trans. CAS-I* 1997; 44, pp. 198-209.
- [5] M. Andrejević Stošović, V. Litovski, "Hierarchical Approach to Diagnosis of Electronic Circuits using ANNs", 10th Symposium on Neural Network Applications in Electrical Engineering, NEUREL 2010, Belgrade, Serbia, 23-25. September 2010, pp. 117-122.
- [6] J. Bandler, A. Salama, "Fault diagnosis of analog circuits", *Proceedings of the IEEE*, vol. 73(8), pp. 1279-1325, 1985.
- [7] F. Pipitone, K. Dejong, W. Spears, "An artificial intelligence approach to analogue system diagnosis", in Liu, R.-W., editor, *Testing and diagnosis of analog circuits and systems, Van Nostrand Reinhold, New York*, 1991, pp. 187-215.
- [8] S. Hayashi, T. Asakura, S. Zhang, "Study of Machine Fault Diagnosis System Using Neural Networks," *Proc. of the Int. Joint Conf. on Neural Networks*, Honolulu, Hawaii, 2002, pp. 233-238.
- [9] T. Golonek, J. Rutkowski, "Use of Genetic Programming to Analog Fault Decoder Design," *ICES'02, Wrocław-Świeradów Zdrój*, 2002.
- [10] C. Pous, J. Colomer, J. Meléndez, J. L. de la Rosa, "Introducing Qualitative Reasoning in fault dictionaries techniques for analog circuit analysis," *Sixteenth International Workshop on Qualitative Reasoning*, 2002, Barcelona, Spain.
- [11] V. Litovski, M. Andrejević, M. Zvolinski, "Analogue Electronic Circuit Diagnosis Based on ANNs," *Microelectronics Reliability*, 2006, pp. 1382-1391.
- [12] R. Spina, S. Upadhyaya, "Linear circuit fault diagnosis using neuromorphic analyzers," *IEEE Trans. on CAS – II: Analog and Digital Signal Processing*, 1997; 44(3), pp. 188-196.
- [13] F. Scarselli, A. C. Tsoi, "Universal approximation using feed-forward neural networks: A survey of some existing methods and some new results," *Neural Networks*, Elsevier, 1998; 11, pp. 15-37.
- [14] D. Liu, A. Starzyk, "A generalized fault diagnosis method in dynamic analogue circuits," *Int. Journal of Circuit Theory and Applications*, 2002; 30, pp. 487-510.
- [15] J. A. Starzyk, D. Liu, "A Decomposition Method for Analog Fault Location," *IEEE Int. Symposium on Circuits and Systems*, Scottsdale, Arizona, 2002, pp. III-157-160.
- [16] M. Andrejević Stošović, M. Dimitrijević, V. Litovski, "Hyper computing implementation in electronic circuits diagnosis," in Proc. of VIII Symposium on Industrial Electronics INDEL 2010, Banja Luka, Bosnia and Herzegovina, 2010.
- [17] T. Masters, "Practical Neural Network Recipes in C++," *Academic Press*, San Diego, 1993.
- [18] Z. Zografski, "A Novel Machine Learning Algorithm and Its Use in Modeling and Simulation of Dynamical Systems," *Proceedings of 5th Annual European Computer Conference, COMPEURO'91*, Bologna, Italy 1991, pp. 860-864.
- [19] J. Heusmann, J. Wiedewitsch, "Future Directions of Modeling and Simulation in the Department of Defense", Proceedings of the SCSC'95, Ottawa, Ontario, Canada, July 1995, pp. 34-26.
- [20] A. Sangiovanni-Vincentelli, C. Li-Kuan, L. Chua, "An efficient heuristic cluster algorithm for tearing large-scale networks," *IEEE Transactions on Circuits and Systems*, vol. 24, Issue 12, Dec. 1977, pp. 709-717.
- [21] T. Kage, F. Kawafuji and J. Niitsuma, "A circuit partitioning approach for parallel circuit simulation," *IEICE Transactions on Fundamentals* E77-A(3), 1994, pp. 461-466.
- [22] B. Andjelkovic, V. Litovski, and V. Zerbe, "Grid-enabled Parallel Simulation Based on Parallel Equation Formulation", *ETRI Journal*, Vol. 32, No. 4, August 2010, pp. 555-565.
- [23] T. Wiedemann, SPEEDUP 512 ? – USING GRAPHIC PROCESSORS FOR SIMULATION, EUROSIM 2010, Prague, September 2010. Proceedings on CD, ISBN 978-80-01-04588-6.
- [24] A. Wakefield, and S. Miller "Improving System Models Using Monte Carlo Techniques on Plant Models" AIAA Modeling and Simulation Technologies Conference and Exhibit, 18 - 21 August 2008, Honolulu, Hawaii.
- [25] T. Sterling, "Beowulf Cluster Computing with Linux", MIT Press, Cambridge, Massachusetts, 2001.
- [26] I. Foster and C. Kesselmann, "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, San Francisco CA, 1999.
- [27] J. A. B. Fortes, R. J. Figueiredo and M. S. Lundstrom, "Virtual Computing Infrastructures for Nanoelectronics Simulation," Proc. of the IEEE, Vol. 93, No. 10, pp. 1839-1847, 2005.
- [28] http://www.rcub.bg.ac.rs/index.php?option=com_content&task=view&id=104&Itemid=181&lang=en
- [29] M. Andrejević, V. Litovski, M. Zvolinski, "Fault Diagnosis in Digital Part of Mixed-Mode Circuit," *Proc. of IEEE 24th Int. Conference on Microelectronics (MIEL2006)*, Niš, Serbia, 2006, pp. 437-440.
- [30] D. Glozić D, "Alecsis 2.1: An object-oriented hybrid simulator," *Ph.D. thesis*, University of Niš, Serbia, 1994, (in Serbian).
- [31] M. Andrejević, V. Litovski, "Fault Diagnosis in Analog Part of Mixed-mode Circuit", *VI simpozijum industrijska elektronika - INDEL 2006*, Banja Luka, Bosnia and Herzegovina, pp. 117-120, 2006.
- [32] M. Andrejević, V. Litovski, "Fault Diagnosis in Digital Part of Sigma-Delta Converter," *Proceedings of Neurel 2006 Conference*, Beograd, Serbia, 2006, pp. 177-180.
- [33] E. B. Baum, D. Haussler, "What size net gives valid generalization," *Neural Computing*, 1989; 1, pp. 151-160. G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15-64.