Analog Integrated Circuits and Signal Processing (2019) 100:167–179
https://doi.org/10.1007/s10470-019-01453-7

Optimum test point selection method for analog fault dictionary techniques

Sara Saeedi[1] · Seyyed Hossein Pishgar[1] · Mahdi Eslami[1]

Received: 29 August 2018 / Revised: 23 February 2019 / Accepted: 5 April 2019 / Published online: 10 April 2019

Abstract

Fault dictionary has two types of online and offline calculation. Online calculation is simple, but offline is excessive and time-consuming. In order to reduce the computation time and dimension of fault dictionary, optimal test points selection is very essential. So, the main purpose, fault isolation, is achieved in a short time. In this paper a new efficient method to select an optimum set of test points for fault diagnosis is proposed. At the first, a fault-isolated table is constructed to pick out the special test points from the candidates. Then, the isolation ability of special test points has been considered. If they can't isolate all of faults, the fault dictionary is rearranged. Therefore, the special test points and isolated faults are eliminated from the table of fault dictionary. In this step, the test point with more single fault is added to special test points. This step is repeated to isolate all of faults. The proposed method is applied on two-stage operational amplifier. By this method, all of faults for this structure are isolated. The computation requirements are very simple than the other methods. In this circuit, the 0.045 s is needed to isolate all of circuit faults. According to the results, it's clear that the method is a good solution to minimize the size of the test points set. Also, it can be a practical method for medium and large scale systems.

Keywords Fault dictionary · Test point · Single fault · Fault isolation

1 Introduction

Fault detection of analog circuits is divided to two types, including simulation before test and after test. Fault detection using fault dictionary is one of the simulation before test techniques [1]. There are three main steps in fault dictionary. The first step to create the dictionary is fault definition that predicts most of possible faults. The circuit is simulated considering these assumptive faults and circuit responses to these faults are obtained. Then, the circuit responses are stored in a dictionary as a signature. The second step is to select the test points. To estimate high degree of isolation with minimum test points and decrease the time of isolation, optimal test points selection can be important [2]. In the final step, the circuit under test is excited by the same input which used to construct the dictionary and signatures. The obtained signatures for this circuit are compared with stored signatures in dictionary. If these signatures were similar, this fault is similar to the stored fault in dictionary at previous step. Therefore, the faulty element can be detected. The proposed method in this paper focuses on the second part, optimum selection of test points.

To construct of fault dictionary in analysis domain can be divided to dc, frequency and time domain. Fault detection in the frequency and time domain has the better performance than the dc domain.

Fault detection in frequency domain has simple and easy online computation. So, to detect of analog circuit fault does not need much time at the test time. This is directly related to the cost and speed of testing. The problem of the fault dictionary techniques is the excessive offline computation and time-consuming. The large number of fault cases is considered in this method and a lot of test points need to isolate all of the faults. Therefore, the dimensions of fault dictionary will be too large; it takes a lot of time to

✉ Seyyed Hossein Pishgar
h_pishgar@wtiau.ac.ir

[1] Department of Electrical Engineering, College of Engineering, Islamic Azad University, West Tehran Branch, Tehran, Iran

process them. In order to reduce the computation time and dimensions of fault dictionary, optimal test points selection is very important. By virtue of this, extra computations are eliminated and the main purpose, fault isolation, is realized in less time.

The nomenclatures of this paper are as follows:

fi	The fault i
nj	The test point j
m	Total number of faults
n	Total number of test points
NAj	Number of ambiguity sets in test point *nj*
Fij	Total number of faults contained in ambiguity set I of test point *nj*
S_c	Candidate test point set
S_{opt}	Optimum test point set

2 Construction of fault dictionary

Detection of fault dictionary has two principal techniques, simulation before and after test. In simulation after test, most of the network simulations are performed at the testing time to identify the network parameters. These techniques are classified as linear and nonlinear according to the nature of the diagnosis equations [3]. In this technique, if the element value is been out of tolerance range, the element is considered as a faulty element. Fault detection using fault dictionary is in the simulation before test category. So, in simulation before the test, the circuit response is obtained by taking different faults and stored in a dictionary [4].

The construction of the dictionary is classified with input signals to the circuit, the analysis domain and the measured responses [3]. In this paper, the classification is based on the analysis domain. The frequency domain approximation is examined to verify this method.

2.1 Frequency domain approximation

Various approximations are proposed for linear and nonlinear networks. The frequency domain approximation has advantages such as: the used theory is much understood and required hardware is so simple [3]. In this method, the Zero and pole values should calculate to create a fault dictionary and choose the test frequencies. In this method, at least one of the test frequencies should be less than the lowest non-zero breakpoints, one greater than the highest break points and one between successive break points [5, 6]. So, four test frequencies will be selected. The basis of these choices is based on zeroes and poles are function in the network parameter. Therefore, the parameter deviation causes a change in the zero and pole location and finally change in the domain of transfer-function H(s). So, the test frequencies will be selected in neighborhood of zero and pole to consider cases that are most likely to be fault.

The process of creating the fault dictionary is summarized below:

1. Computation of network zeroes and poles
2. Selection the test frequency from the break points (zero and pole)
3. Computation of gain in test frequency by using nominal value of elements
4. Computation of gain in test frequency while assumptive faults applied to the circuit
5. Creation a fault dictionary that rows show different faults and columns show different test points

The fault dictionary is created using above steps. As mentioned, the time computation and dimensions of fault dictionary decrease by optimum test points selection. So, the main purpose, fault isolation, is reached in less time. In next section, the new proposed algorithm to select the optimum test point is described.

3 New algorithm for test points selection

The new algorithm contains two parts: the inclusive part and the exclusive part which are discussed below:

Part1, *the inclusive part* [2]:

1. Initialize the desired test-point set S_{opt} as a null set and let S_c consists of all candidate test points. Then, the integer-coded fault dictionary is constructed based on the frequency responses.
2. Construct and extend the fault-isolated table to obtain the extended fault-isolated Table
3. Check the column NI_i of extended fault-isolated table, find all the corresponding test points (column) of which the NI_i value equal to one, and add these test points to S_{opt}.
4. In this step, if the test points of S_{opt} can isolate all the faults, this step is consider as the final step and all faults are isolated by the minimum test points. Otherwise, the algorithm goes to step 5.
5. Rearrange the fault dictionary and eliminate the rows of the fault dictionary that can be isolate by test points in S_{opt} together. Remove the selected test points from S_c and eliminate the corresponding column of the fault dictionary.

Part2, *exclusive part:*

6. In this step, after elimination of special test points and fault isolation, a fault dictionary with smaller

dimensions is constructed. Then, the ambiguity sets of mini dictionary are constructed and the test point with more single fault is selected and added to S_{opt}. Notice, if the single fault in mini dictionary be over, the test point with more ambiguity sets in corresponding column is selected and added to S_{opt}. The algorithm goes to step 4.

Steps 4, 5, 6 repeated until the entire fault being isolated. These steps described below:

3.1 Integer-coded fault dictionary

The integer-coded fault dictionary technique was first proposed by Lin and Elcherif [7]. This technique is an important tool for optimum test point selection [8]. At the first, the fault dictionary is constructed that rows show the difficult faults and columns show the entire test points. Next, ambiguity sets is created. Ambiguity group is defined as any two faulty cases fall into the same ambiguity set if the gap between the gains values of their response is less than 1.5 dB [9–11]. After grouping the faults, the group number is assigned to each fault instead of the fault value. By this way the integer-coded fault dictionary is constructed [2].

For example, the integer-coded fault dictionary of analog circuit under different faulty conditions, include 19 faults and 11 test points, is shown in Table 1.

Table 1 Fault dictionary

Faults	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11
F1	4	3	5	1	3	7	4	3	1	2	1
F2	0	3	0	7	2	0	1	2	0	1	0
F3	7	8	8	1	0	0	1	1	1	1	2
F4	2	0	0	0	1	3	2	3	3	3	3
F5	3	3	4	4	3	5	0	1	1	1	4
F6	1	7	2	1	2	1	0	4	5	0	4
F7	0	0	0	5	0	1	0	5	4	0	6
F8	4	2	0	8	1	1	3	5	3	5	5
F9	4	1	5	0	5	2	5	0	3	6	7
F10	4	4	5	3	3	2	5	0	6	6	7
F11	4	2	0	6	1	8	6	6	6	6	4
F12	4	5	0	4	2	8	7	6	1	0	8
F13	8	2	0	5	8	4	8	3	1	1	4
F14	4	2	2	2	5	4	2	1	1	6	7
F15	4	2	2	3	2	4	3	5	7	0	4
F16	6	6	2	3	2	4	0	7	8	7	7
F17	5	2	6	3	2	4	0	0	8	8	4
F18	4	2	2	3	2	4	0	1	8	6	4
F19	4	2	2	3	2	4	0	6	7	0	7

In step 1 of the proposed algorithm, the candidate test points set, S_c, is initialized as {n1,n2,n3,n4,n5,n6,n7, n8,n9,n10,n11}.

3.2 fault-isolated table

In many cases, some faults can only be detected and isolated by some special test points, whereas the other can be isolated by more than one test points [9]. In order to facilitate the isolation of these particular types of faults, special test points must be selected. As a result, the problem of selecting the test points is easily solved and computational costs are greatly reduced.

In order to find these special test points, a fault isolated table should be constructed and extended. Then, this table will be combined with the integer-coded fault dictionary to find the final solution. In the next section, the design of the fault isolated table will discuss in details.

3.2.1 Construction of fault isolated table

Since, the integer-coded fault dictionary represents the information of different faults at the different test points, the ambiguity group with one fault at each test point should be finding in the dictionary and mark them out. First of all, for each fault in the integer-coded fault dictionary, consider the integers along its row. Each integer in that row is compared with all the integers in its corresponding column, If the integer is not repeated at any row of the corresponding column, the element at the same place of the integer-coded fault dictionary table replaced by “1” and otherwise, it replaced by “0”. By this way, the fault isolated table is constructed. In the fault isolated table, the element “1” means the fault of this row can be isolated by the test point of this column, and the element “0” means opposite [2].

Table 2 gives the fault-isolated table based on the integer-coded fault dictionary that shows in Table 1.

3.3 Extension of the fault-isolated table

In order to access the special test points easily, a new column NI_i is added to the fault isolated table to calculate the total number of test points which can isolate fault f_i. In the fault-isolated table, the element “1” means the fault of this row can be isolated by the test point of this column, and the element “0” means opposite [2]. Result shows in Table 3.

According to the above steps, S_{opt} is obtained using Table 3 and NIi column. S_{opt} = {n2, n4, n6}

Table 2 Fault-isolated table

Faults	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11
F1	0	0	0	0	0	1	1	0	0	1	1
F2	0	0	0	1	0	0	0	1	1	0	1
F3	1	1	1	0	0	0	0	0	0	0	1
F4	1	0	0	0	0	1	0	0	0	1	1
F5	0	0	0	0	0	1	0	0	0	0	0
F6	0	1	0	0	0	0	0	1	1	0	0
F7	0	0	0	0	0	0	0	0	1	0	1
F8	0	0	0	1	0	0	0	0	0	1	1
F9	0	0	0	0	0	0	0	0	0	0	0
F10	0	1	0	0	0	0	0	0	0	0	0
F11	0	0	0	1	0	0	0	0	0	0	0
F12	0	1	0	0	0	0	1	0	0	0	1
F13	1	0	0	0	1	0	1	0	0	0	0
F14	0	0	0	0	0	0	0	0	0	0	0
F15	0	0	0	0	0	0	0	0	0	0	0
F16	1	1	0	0	0	0	0	1	0	1	0
F17	1	1	1	0	0	0	0	0	0	1	0
F18	0	0	0	0	0	0	0	0	0	0	0
F19	0	0	0	0	0	0	0	0	0	0	0

Table 3 Extended fault-isolated table

Faults	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11	NIi
F1	0	0	0	0	0	1	1	0	0	1	1	4
F2	0	0	0	1	0	0	0	1	1	0	1	4
F3	1	1	1	0	0	0	0	0	0	0	1	4
F4	1	0	0	0	0	1	0	0	0	1	1	4
F5	0	0	0	0	0	1	0	0	0	0	0	1
F6	0	1	0	0	0	0	0	1	1	0	0	3
F7	0	0	0	0	0	0	0	0	1	0	1	2
F8	0	0	0	1	0	0	0	0	0	1	1	3
F9	0	0	0	0	0	0	0	0	0	0	0	0
F10	0	1	0	0	0	0	0	0	0	0	0	1
F11	0	0	0	1	0	0	0	0	0	0	0	1
F12	0	1	0	0	0	0	1	0	0	0	1	3
F13	1	0	0	0	1	0	1	0	0	0	0	3
F14	0	0	0	0	0	0	0	0	0	0	0	0
F15	0	0	0	0	0	0	0	0	0	0	0	0
F16	1	1	0	0	0	0	0	1	0	1	0	4
F17	1	1	1	0	0	0	0	0	0	1	0	4
F18	0	0	0	0	0	0	0	0	0	0	0	0
F19	0	0	0	0	0	0	0	0	0	0	0	0

3.4 Checking the fault isolation

In this section, check whether the test points in S_{opt} can isolate the entire fault in integer-coded fault dictionary or not. As shown in Table 3, the test points in S_{opt} can't isolate the entire fault in the integer-coded fault dictionary. So the algorithm goes to part 5.

3.5 Rewrite the integer-coded fault dictionary

Actually, S_{opt} can't isolate the entire fault in the fault dictionary. Therefore, the rows of the fault dictionary that can be isolate by this special test points, eliminated from integer-coded fault dictionary. Also, the special test points removed from S_c and corresponding column of the integer-coded fault dictionary eliminated. Therefore, the mini integer-coded fault dictionary is constructed, as shown in Table 4. Notice, the test points are less than before, so the needed time for isolation of the faults has decreased.

3.6 The principle of adding test points

In this step, the ambiguity groups table is created for integer-coded fault dictionary as shown in Table 5. In this table, the test point with more single fault will be selected and added to S_{opt}. Then, test point $n7$ has these conditions and can isolate $F9$, $F13$, $F14$, and $F15$ faults. So, these faults eliminate from Table 4 and S_{opt} and the test point n7 added to S_{opt}. Then, the algorithm goes to step 4 and checks whether the test points in S_{opt} can isolate entire fault in fault dictionary or not and the answer is no. So, the steps 5 and 6 implemented for the remaining faults. Thus, the integer-coded is created for $F7$, $F18$ and $F19$ faults, as shown in Table 6. Ambiguity groups are created for these faults as shown in Table 7.

In Table 7, one of the test points n8, n9 and n11 can be selected. Because, each of these test points can isolate remaining faults, so the test point n8 added to Sopt. Thus, Sopt is initialized as {n2, n4, n6, n7, n8}.

Table 4 Mini integer coded fault dictionary 1

Faults	n1	n3	n5	n7	n8	n9	n10	n11
F7	0	0	0	0	5	4	0	6
F9	4	5	5	5	0	3	6	7
F13	8	0	8	8	3	1	1	4
F14	4	2	5	2	1	1	6	7
F15	4	2	2	3	5	7	0	4
F18	4	2	2	0	0	7	0	7
F19	4	2	2	0	0	7	0	7

Table 5 Ambiguity groups of Table 4

Ambiguity groups	n1	n3	n5	n7	n8	n9	n10	n11
0	F7	(F7,F13)	F7	(F7, F18, F19)	(F9, F18, F19)	–	(F7, F15, F18,F19)	–
1	–	–	–	–	F14	(F13,F14)	F13	–
2	–	(F14, F15, F18,F19)	(F15, F18, F19)	F14	–	–	–	–
3	–	–	–	F15	F13	F9	–	–
4	(F14,F15, F18,F19)	–	–	–	–	F7	–	(F13,F15)
5		F9	(F9,F14)	F9	(F7,F15)			
6	–	–	–	–	–	–	(F9, F14)	F7
7	–	–	–	–	–	(F15, F18, F19)	–	(F9, F14 F18,F19)
8	F13	–	F13	F13	–	–	–	–

Table 6 Mini integer coded fault dictionary 2

Faults	n1	n3	n5	n8	n9	n10	n11
F7	0	0	0	5	4	0	6
F18	4	2	2	1	8	6	4
F19	4	2	2	6	7	0	7

Now, the algorithm goes to step 4 and checks that whether the test points in Sopt can isolate the entire faults or not and the answer is yes. So eleven test points decrease to five. Also, all of the faults are isolate by these test points. Hence, additional computations are eliminated and the main purpose, faults isolation, realized in less time.

4 Time complexity of test point selection [2]

As discussed above, the fault-isolated table consists of *Nf* rows and *NT* columns. On the other hand, the extended fault-isolated table has one more column. Now, by using the method that Pinjala and Prasad proposed for time computation [12, 13], the time complexity of step 3 is *O(Nf(NT + 1))*.

In step 4, suppose m1 test points are added to Sopt and these points can isolate Nf1 faults, and the time complexity is also O(Nf(NT + 1)). In case of Nf1 = Nf, the stop condition fulfills, the Sopt will be find directly and the time complexity will reduce quite a lot.

In step 6, suppose m2 test points are added into S_{opt} in the next algorithm iterations, the time complexity is calculated Refer to "(1),"

$$O((Nf - Nf1))p' \log(Nf - Nf1)), \text{ where } P' = (NT - m1) + (NT - m1 - 1) + \cdots + (NT - m1 - m2 + 1). \quad (1)$$

In step 5, delete all the corresponding rows and columns in the fault dictionary and the time complexity is O(Nf) + O(m1 + m2) = O(Nf) + O(m). Where m is the total number of test points in $S_{opt}(m = |S_{opt}|)$. Since, the total number of test points is NT > m so p' ≫ NT. When,

Table 7 Ambiguity groups of Table 6

Ambiguity groups	n1	n3	n5	n8	n9	n10	n11
0	F7	F7	F7	–	–	(F7, F19)	–
1	–	–	–	F18	–	–	–
2	–	(F18, F19)	(F18, F19)	–	–	–	–
4	(F18, F19)	–	–	–	F7	–	F18
5	–	–	–	F7	–	–	–
6	–	–	–	F19	–	F18	F7
7	–	–	–	–	F19	–	F19
8	–	–	–	–	F18	–	–

the last stage is running, the total time complexity is O(Nf m log Nf).

5 Experiment

To demonstrate the efficiency of the proposed algorithm, this algorithm is applied to three circuits that described as below.

5.1 Experiment on circuits

5.1.1 Two stage operation amplifier

Two-stage operational amplifier is shown in Fig. 1. The first step is biasing the two stage amplifier in Fig. 1, using 0.18 µm standard CMOS technology; all the transistors are in saturation mode. Then, the fault dictionary in frequency domain will create.

In order to creating the fault dictionary, the test frequencies must be calculated, according to the described method in Sect. 2.1. As shown in Fig. 1, four test frequencies were obtained for operational amplifier. The nominal gain at each test frequency is obtained. To calculate the nominal gain, the entire elements have their nominal values. The results of nominal gain in four test frequencies are shown in Table 8.

Table 8 Nominal Gain in test frequencies

Test frequency	Nominal gain (dB)
39 meg	5.92
50 meg	11.4
290 meg	− 7.25
8 g	− 6.74

In the next step, the value of parameter L for all transistors deviated from the nominal value at each test frequencies. By regarding the variations of these parameters, the gain is obtained again. By this way, the fault dictionary is created. The rows and column are shown the different faults and the different test points, respectively. The results are presented in Table 9. After creating the fault dictionary, the proposed algorithm of optimum test point selection applied (Table 9).

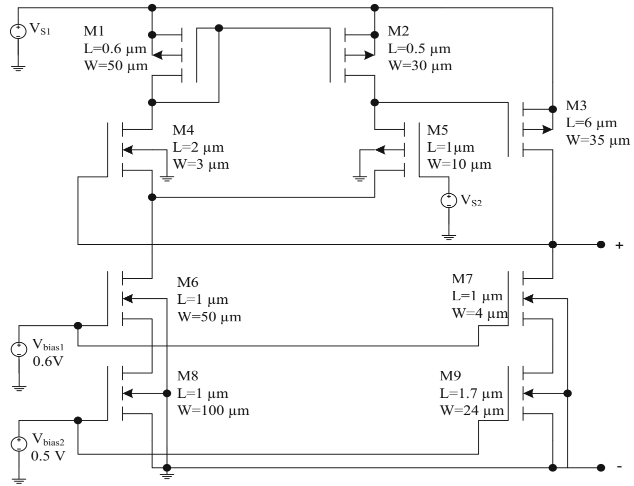

Fig. 1 Two stage operation amplifier [14]

In step 1 of the proposed method, the initialized work is done. So, the candidate test point set S_c is initialized as {39 meg, 50 meg, 290 meg, 8 g}. Then, the ambiguity groups for this fault dictionary are created as shown in Table 10. Next, the integer-coded fault dictionary is created by using ambiguity groups table (Table 11).

In next step, the fault isolated and extended tables are constructed by the introduced procedures in Sect. 3.2. Extended Fault isolated table is shown in Table 12. In this table, the special test points can find (39 meg). So, this test point is selected and added to S_{opt}.

Then, the proposed algorithm is checked whether the test point in Sopt can isolate the entire faults in fault dictionary or not. Because, the element of the test point 39 meg for the entire fault is 1, as shown in Table 12, this test point can isolate the entire fault in fault dictionary and don’t need to consider next section.

Thus, the numbers of test points decreased from four to one and all of the faults are isolated using this test point. Therefore, the offline computation of the fault dictionary and the needed time for fault isolation is decreased. Due to isolating all faults in Sect. 3, the needed time to isolate all faults is O(Nf(NT + 1)). According to value of NT and Nf, the needed time is 0.045 s.

Table 9 Fault dictionary

Faults	39 meg	50 meg	290 meg	8 g
F1 (dB)	6.39	3.2	− 9.59	− 7.17
F2 (dB)	0.31	0.6	− 0.24	− 0.3
F3 (dB)	8.13	9.27	− 0.37	− 0.3
F4 (dB)	− 1.39	− 0.04	− 1.35	− 1.3
F5 (dB)	11	18	− 9.83	− 11.7
F6 (dB)	19.8	10.2	− 6.05	− 4.82
F7 (dB)	15.7	6.07	− 7.98	− 6.74
F8 (dB)	17.3	6.16	− 5.5	− 4.7
F9 (dB)	13	15.6	− 7.75	− 6.74

Table 10 Ambiguity groups of Table 9

Ambiguity groups	39 meg	50 meg	290 meg	8 g
0	(− 1.39 dB) F4	(− 0.04 to 0.6 dB) F2, F4	(− 0.24 to -1.35 dB) F2, F3, F4	(− 0.3 dB) F2, F3
1	(0.31 dB) F2	(3.2 dB) F1	(− 5.5 to − 6.05 dB) F6, F8	(− 1.3 dB) F4
2	(6.39 dB) F1	(6.07 to 6.16 dB) F7, F8	(− 7.75 to − 7.98 dB) F7, F9	(− 4.7 to − 4.82 dB) F6, F8
3	(8.13 dB) F3	(9.27 to 10.2 dB) F3, F6	(− 9.59 to − 9.83 dB) F1, F5	(− 6.74 to − 7.17 dB) F1, F7, F9
4	(11 dB) F5	(15.6 dB) F9	–	(− 11.7 dB) F5
5	(13 dB) F9	(18 dB) F5	–	–
6	(15.9 dB) F7	–	–	–
7	(17.3 dB) F8	–	–	–
8	(19.8 dB) F6	–	–	–

Table 11 Integer coded fault dictionary of Table 10

Faults	39 meg	50 meg	290 meg	8 g
F1	2	1	3	3
F2	1	0	0	0
F3	3	3	0	0
F4	0	0	0	1
F5	4	5	3	4
F6	8	3	1	2
F7	6	2	2	3
F8	7	2	1	2
F9	5	4	2	3

Table 12 Extended Fault isolated table

Faults	39 meg	50 meg	290 meg	8 g	NIi
F1	1	1	0	0	2
F2	1	0	0	0	1
F3	1	0	0	0	1
F4	1	0	0	1	2
F5	1	1	0	1	3
F6	1	0	0	0	1
F7	1	0	0	0	1
F8	1	0	0	0	1
F9	1	1	0	0	2

By comparing the final results of the proposed algorithm and Ref. [14] can released the proposed algorithm is very fast and efficiency.

The [14] is devoted to diagnose the fault of CMOS circuit and estimated the value of a set of potentially faulty process parameters. At the preliminary stage of the diagnosis process, the sensitivities of the output voltage due to variations of several parameters of all the transistors, for different input voltage, are calculated [14]. Because the proposed algorithm in [14] didn't allow to test the transistors whose parameters have slightly influence in the output voltage, the fault of seven transistors was detected. Also, the faults were isolated using nonlinear equation that is very complicated and time-consuming. While, the fault isolation using the proposed algorithm has simple computation which decreased the time consuming.

The required time for isolation all of the faults in dictionary of Fig. 1 using the new algorithm is 0.045 s, Against, the consumption time of isolating one of the dictionary faults of Fig. 1 by mentioned method in Ref. [14] is 227.5 s. Also, as discuss above, the proposed algorithm in [14] can't test all of the transistors, whereas the proposed algorithm examines all transistors and isolates all of the faults.

So, the proposed algorithm is efficient in sense of simple computation and less needed time for fault isolation and the degree of fault isolation is increased by this method.

5.1.2 Experiment on a negative feedback circuit

The negative feedback circuit is shown in Fig. 2. The input signal is a 1-KHz, 7 mV sinusoidal wave. Totally, there are 35 potential faults f1 to f35 (including the nominal case) and ten test points n1 to n10. The responses of all the test points under different faulty conditions are obtained by PSPICE simulation [2]. To demonstrate the efficiency of the proposed algorithm and compare with other algorithms, we use the integer–coded fault dictionary defined in ref. 2 which is shown in Table 13. In the table, "D" represents the integer code of dc voltage responses, and "A" represents the integer code of ac voltage responses. The extended fault-isolated table is shown in Table 14 [2].

Now we apply our algorithm to Table 13. First the special test point should be finding and added to S_{opt} as shown below:

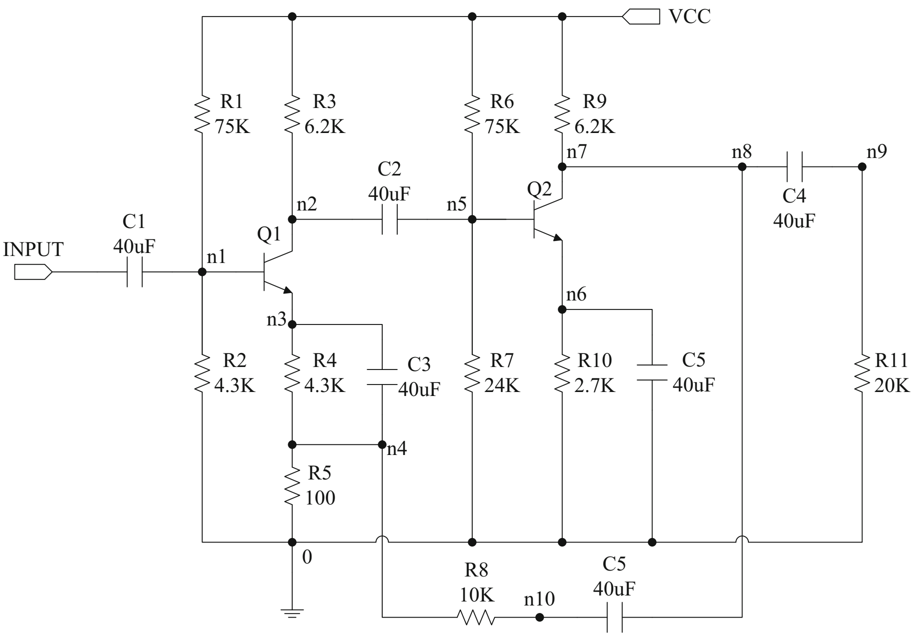

Fig. 2 Negative feedback circuit [2]

$S_{opt} = \{t1, t4, t9, t19, t20\}$

In next step with regard to test point in S_{opt} can't isolate the entire fault in fault dictionary, the algorithm goes to step 5, in this step, the integer-coded fault dictionary is rearranged, and the corresponding column of special test point and corresponding rows of faults which can be isolated by them are eliminated from the integer-coded fault dictionary. Now the mini integer coded fault dictionary is constructed. In next step, any test point with more single fault in corresponding column will be selected and added to S_{opt}. So t18 and t5 are added to S_{opt}. The results are listed in Table 15, and shows that number of optimum test points of our algorithm is less than Yang's algorithm and Dongshengs algorithm, and we estimated this result in less time. So by this way, we can isolate the faults of CUT using 7test points in a short time.

5.1.3 Leapfrog filter

The leapfrog filter circuit is shown in Fig. 3. We can find six operation amplifiers in the circuit and these amplifiers can be treated as modules in practice. We assume the amplifiers have low-input-impedance failure besides the resistance and capacitance's catastrophic faults. The input signal is a 1-KHz, 4 V sinusoidal wave. Totally, there are 23 potential faults f1 to f23 (including the nominal case) and 12 test points n1 to n12. The responses of all the test points under different faulty condition are obtained by PSPICE simulation. According to the response voltage of the test point under different faulty mode, we construct the integer-coded fault dictionary and the extended fault-isolated table which is shown in Tables 16 and 17 respectively [2].

The proposed algorithm in this paper is applied on integer coded fault dictionary which shown in Table 16. First the special test point is selected and added to S_{opt}, as shown below:

$S_{opt} = \{n1, n2, n3, n5, n7, n11\}$

Then checked whether the test point in S_{opt} can isolate the entire fault in fault dictionary or not, that in this step the answer is no. according to Extended fault-isolated table shown in Table 17, the special test point in S_{opt} can't isolate the faults f2, f3, f4, f11, f12, f14, f16, f17, f22. So the algorithm goes to next step and the integer-coded fault dictionary is rearranged and mini dictionary is constructed. Then more test points will added to S_{opt} based on step6 that defined in this paper, as shown in below:

$S_{opt} = \{n1, n2, n3, n5, n7, n11, n10\}$

Now we isolated the entire faults in fault dictionary by 7 test points. Then the time complexity of proposed algorithm is calculated and the results show that we isolate the faults in less time comparison with other algorithms. As shown in Table 18.

5.2 Statistical experiment

To demonstrate efficiency of proposed algorithm, Statistical test is done on 100 hundred of fault dictionaries with 9 faults and 4 test points. These fault dictionaries is generated randomly using Monte Carlo and simulated by Hspice. The proposed algorithm are programmed by MATLAB and

Table 13 Integer-coded fault dictionary for Fig. 2 [2]

Faults	Dt1	At2	Dt3	At4	Dt5	At6	Dt7	At8	Dt9	At10	Dt11	At12	Dt13	At14	Dt15	At16	Dt17	At18	Dt19	At20
F1(NOM)	2	0	2	0	1	0	0	0	2	0	1	0	4	7	4	7	0	7	0	7
F2(R1 open)	0	0	3	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F3(R2 open)	3	0	1	0	3	0	0	0	2	0	1	0	4	6	4	6	0	6	0	6
F4(R3 open)	1	0	0	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F5(R4 open)	2	0	3	0	2	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F6(R5 open)	2	0	3	0	2	0	1	0	2	0	1	0	4	0	4	0	0	0	1	0
F7(R6 open)	2	0	2	6	1	0	0	0	0	6	0	0	5	0	5	0	0	0	0	0
F8(R7 open)	2	0	2	0	1	0	0	0	3	0	3	0	1	1	1	1	0	1	0	1
F9(R8 open)	2	0	2	1	1	0	0	0	2	1	1	0	4	10	4	10	0	10	0	10
F10(R9 open)	2	0	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
F11(R10 open)	2	0	2	4	1	0	0	0	2	4	2	0	5	2	5	2	0	2	0	2
F12(R1 short)	4	0	3	0	4	0	0	0	2	0	1	0	4	3	4	3	0	3	0	3
F13(R3 short)	2	0	3	0	1	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F14(R4 short)	1	0	0	0	0	0	0	0	2	0	1	0	4	5	4	5	0	5	0	5
F15(R5 short)	2	0	2	8	1	0	0	0	2	7	1	0	3	9	3	9	0	9	0	9
F16(R6 short)	2	0	2	0	1	0	0	0	5	0	4	0	5	0	5	0	0	0	0	0
F17(R7 short)	2	0	2	0	1	0	0	0	0	0	0	0	5	0	5	0	0	0	0	0
F18(R8 short)	2	0	2	0	1	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F19(R9 short)	2	0	2	2	1	0	0	0	2	2	1	0	5	0	5	0	0	0	0	0
F20(R11 short)	2	0	2	2	1	0	0	0	2	2	1	0	4	0	4	0	0	0	0	0
F21(C2 open)	2	0	2	7	1	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F22(C3 open)	2	0	2	0	1	0	0	0	2	0	1	0	4	3	4	3	0	3	0	3
F23(C4 open)	2	0	2	0	1	0	0	0	2	0	1	0	4	7	4	7	0	0	0	7
F24(C5 open)	2	0	2	3	1	0	0	0	2	3	1	1	4	4	4	4	0	4	0	4
F25(C6 open)	2	0	2	1	1	0	0	0	2	1	1	0	4	10	4	10	0	10	0	0
F26(C2 short)	2	0	1	0	1	0	0	0	4	0	3	0	1	0	1	0	0	0	0	0
F27(C4 short)	2	0	2	0	1	0	0	0	2	0	1	0	2	7	2	7	1	7	0	7
F28(C6 short)	2	0	2	0	1	0	0	0	2	0	1	0	1	7	1	7	0	7	2	7
F29(Q1B open)	2	0	3	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F30(Q1C open)	1	0	3	0	0	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F31(Q1BC short)	3	0	1	0	3	0	0	0	2	0	1	0	4	8	4	8	0	8	0	8
F32(Q1CE short)	2	0	1	0	3	0	0	0	2	0	1	0	4	0	4	0	0	0	0	0
F33(Q2B open)	2	0	2	5	1	0	0	0	2	5	0	0	5	0	5	0	0	0	0	0
F34(Q2BC short)	2	0	2	0	1	0	0	0	3	0	3	0	1	0	1	0	0	0	0	0
F35(Q2CE short)	2	0	2	5	1	0	0	0	2	5	3	0	1	0	1	0	0	0	0	0

Table 14 Extended fault-isolated table for Fig. 2 [2]

Faults	Dt1	At2	Dt3	At4	Dt5	At6	Dt7	At8	Dt9	At10	Dt11	At12	Dt13	At14	Dt15	At16	Dt17	At18	Dt19	At20	NIi
F1(NOM)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F2(R1 open)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
F3(R2 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
F4(R3 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F5(R4 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F6(R5 open)	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2
F7(R6 open)	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2
F8(R7 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
F9(R8 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
F10(R9 open)	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	3
F11(R10 open)	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	1	7
F12(R1 short)	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
F13(R3 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F14(R4 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
F15(R5 short)	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	0	1	0	1	8
F16(R6 short)	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2
F17(R7 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F18(R8 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F19(R9 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F20(R11 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F21(C2 open)	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
F22(C3 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F23(C4 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F24(C5 open)	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	7
F25(C6 open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F26(C2 short)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
F27(C4 short)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	3
F28(C6 short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
F29(Q1B open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F30(Q1C open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F31(Q1BC short)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	4
F32(Q1CE short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F33(Q2B open)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F34(Q2BC short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F35(Q2CE short)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15 Results of different algorithms for Fig. 2

Algorithms	S_{opt}	Number of optimum test points	Elapsed time(s)
Starzyk's [15]	t1, t3, t4, t5, t7, t9, t14, t15, t18, t20	10	0.138
Yang's [9]	t1, t3, t4, t5, t7, t9, t13, t18, t20	9	0.133
Pinjala's [13]	t1, t3, t4, t5, t7, t9, t11, t13, t18, t20	10	0.106
Dongshengs [2]	t1, t3, t4, t5, t7, t9, t13, t18, t20	9	0.085
Proposed algorithm	t1, t4, t5, t9, t18, t19, t20	7	0.013

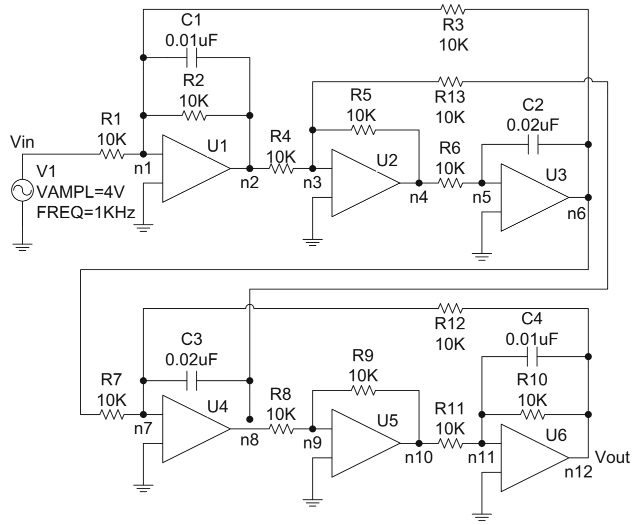

Fig. 3 Leapfrog filter [2]

tested on Intel core f_4790 K, 4G-Hz with 8G RAM. The results show in Table 19.

6 Conclusion

For large-scale circuits with hundreds of elements, the problem is isolation all the faults of circuit by the minimum test point. The proposed methods are very costly and time-consuming.

In this paper, with respect to isolation all of the faults by the test points in S_{opt} in step 3, the time consumption of isolation the entire faults will be O(Nf(NT + 1)) and using the value of NT and Nf in the fault dictionary, the consumption time will be 0.045 s. Also, this algorithm has the high degree of isolation to isolate the fault of the entire transistor in two-stage operation amplifier circuit.

Table 16 Integer-coded fault dictionary for Fig. 3 [2]

Faults	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11	n12
F1(NOM)	4	2	2	3	1	5	3	3	2	3	3	3
F2(C2 open)	8	7	6	7	4	9	7	9	4	6	4	7
F3(C2 short)	4	3	2	3	1	3	3	2	2	2	3	2
F4(C3 open)	1	0	0	0	0	1	1	1	1	1	1	1
F5(R1 open)	9	8	6	7	4	10	7	9	4	6	4	7
F6(R1 short)	5	5	4	6	4	8	7	8	4	6	4	7
F7(R2 open)	6	7	6	6	4	9	7	8	4	6	4	7
F8(R2 short)	4	1	2	1	1	3	3	2	2	2	3	2
F9(R3 open)	2	0	0	0	0	0	0	0	0	0	0	0
F10(R3 short)	0	0	0	0	0	1	1	0	0	0	0	0
F11(R4 open)	4	3	2	1	1	3	3	2	2	2	3	2
F12(R7 short)	1	0	0	0	0	1	1	0	0	0	0	0
F13(R8 open)	8	7	5	6	4	9	6	6	4	6	4	6
f14(R10 open)	4	4	2	2	1	4	3	3	2	3	3	5
F15(R11 open)	7	6	3	5	3	7	4	7	4	6	4	7
F16(R12 short)	1	0	0	0	0	1	0	0	0	0	0	0
F17(R13 open)	4	4	2	4	1	6	3	4	2	4	3	4
F18(U1 fail)	3	0	0	0	0	1	1	0	0	0	0	0
F19(U2 fail)	1	0	1	0	0	1	1	1	1	1	1	1
F20(U3 fail)	8	7	6	6	2	9	7	8	4	6	4	7
F21(U4 fail)	1	0	0	0	0	1	2	1	1	1	1	1
F22(U5 fail)	8	7	6	7	4	9	7	7	3	5	4	7
F23(U6 fail)	1	0	0	0	0	1	1	1	1	1	2	1

Table 17 Extended fault-isolated table For Fig. 3 [2]

Faults	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11	n12	NIi
F1(NOM)	0	1	0	0	0	1	0	0	0	0	0	1	3
F2(C2 open)	0	0	0	0	0	0	0	0	0	0	0	0	0
F3(C2 short)	0	0	0	0	0	0	0	0	0	0	0	0	0
F4(C3 open)	0	0	0	0	0	0	0	0	0	0	0	0	0
F5(R1 open)	1	1	0	0	0	1	0	0	0	0	0	0	3
F6(R1 short)	1	1	1	0	0	1	0	0	0	0	0	0	4
F7(R2 open)	1	0	0	0	0	0	0	0	0	0	0	0	1
F8(R2 short)	0	1	0	0	0	0	0	0	0	0	0	0	1
F9(R3 open)	1	0	0	0	0	1	0	0	0	0	0	0	2
F10(R3 short)	1	0	0	0	0	0	0	0	0	0	0	0	1
F11(R4 open)	0	0	0	0	0	0	0	0	0	0	0	0	0
F12(R7 short)	0	0	0	0	0	0	0	0	0	0	0	0	0
F13(R8 open)	0	0	1	0	0	0	1	1	0	0	0	1	4
f14(R10 open)	0	0	0	1	0	1	0	0	0	0	0	1	3
F15(R11 open)	1	1	1	1	1	1	1	0	0	0	0	0	7
F16(R12 short)	0	0	0	0	0	0	0	0	0	0	0	0	0
F17(R13 open)	0	0	0	1	0	1	0	1	0	1	0	1	5
F18(U1 fail)	1	0	0	0	0	0	0	0	0	0	0	0	1
F19(U2 fail)	0	0	1	0	0	0	0	0	0	0	0	0	1
F20(U3 fail)	0	0	0	0	1	0	0	0	0	0	0	0	1
F21(U4 fail)	0	0	0	0	0	0	1	0	0	0	0	0	1
F22(U5 fail)	0	0	0	0	0	0	0	0	1	1	0	0	2
F23(U6 fail)	0	0	0	0	0	0	0	0	0	0	1	0	1

Table 18 Results of different algorithms for Fig. 3

Algorithms	S_{opt}	Number of optimum test points	Elapsed time(s)
Starzyk's [15]	n1, n2, n3, n4, n7, n8, n11	7	0.093
Yang's [9]	n1, n2, n3, n4, n7, n8, n11	7	0.083
Pinjala's [13]	n1, n2, n3, n4, n7, n8, n11	7	0.084
Dongshengs [2]	n1, n2, n3, n4, n7, n8, n11	7	0.070
Proposed algorithm	n1, n2, n3, n5, n7, n10, n11	7	0.009

Table 19 Statistical results of proposed algorithm and computation time per dictionary

Size of min. set	Proposed algorithm
2	70
3	30
The average running time (in sec)	0.002

Finally, as shown the results, the proposed method has a better performance in sense of simple computation and less needed time for fault isolation and the degree of fault isolation increased by this method. Thus, this algorithm is a good solution to minimize the size of the test point-set and practical for large scale systems.

References

1. Rao, V. M., & Sundari, S. P. S. (2014). Optimized multi frequency approach to analog fault diagnosis using monte carlo analysis. *Electrical and Electronic Engineering, 4,* 25–30.
2. Zh, D., & Yuzhu, H. (2015). A new test points selection method for analog fault dictionary techniques. *Analog Integrated Circuits and Signal Processing, 82,* 435–448.
3. Bandler, J. (1985). Fault diagnosis of analog circuits. *Proceedings of the IEEE, 73*(8), 1279–1325.
4. Rao, S. V. M., & Sundari, K. S. (2014). Optimized multi frequency approach to analog fault diagnosis using Monte Carlo analysis. *Electrical and Electronic Engineering, 4*(2), 25–30.
5. Seshu, S., & Waxman, R. (1966). Fault isolation in conventional linear systems-a feasibility study. *IEEE Transactions on Reliability, 15*(1), 11–16.
6. Meanpaa, J. H., Stehman, C. J., & Stahl, W. J. (1969). Fault isolation in conventional linear systems: A progress report. *IEEE Transactions on Reliability, 18*(1), 12–14.
7. Lin, P. M., & Elcherif, Y. S. (1985). Analogue circuits fault dictionary—New approaches and implementation. *The*

International Journal of Circuit Theory and Applications., 13(2), 149–172.

8. Yang, C. L., Tian, S. L., & Long, B. (2009). Test points selection for analog fault dictionary techniques. *The Journal of Electronic Testing, 25*(2–3), 157–168.
9. Yang, C. L., Tian, S. L., & Long, B. (2009). Application of heuristic graph search to test point selection for analog fault dictionary techniques. *The IEEE Transactions on Instrumentation and Measurement, 58*(7), 2145–2158.
10. Hochwald, W., & Bastian, J. D. (1979). A dc approach for analog fault dictionary determination. *IEEE Transactions on Circuits and Systems, 26,* 523–529.
11. Yang, C. L., Tian, S. L., & Long, B. (2010). A test points selection method for analog fault dictionary techniques. *Analog Integrated Circuits and Signal Processing, 63*(2), 349–357.
12. Prasad, V. C., & Babu, N. S. C. (2000). Selection of test nodes for analog fault diagnosis in dictionary approach. *The IEEE Transactions on Instrumentation and Measurement., 49*(6), 1289–1297.
13. Pinjala, K. K., & Bruce, C. K. (2003). An approach for selection of test points for analog fault diagnosis. In *Proceedings of the 18th IEEE international symposium on defect and fault tolerance in VLSI systems* (pp. 287–294).
14. Tadeusiewicz, M., & Hałgas, S. (2016). Multiple soft fault diagnosis of DC analog CMOS circuits designed in nanometer technology. *Analog Integrated Circuits and Signal Processing, 88*(1), 65–77.
15. Starzyk, J. A., Liu, D., Liu, Zh, Nelson, D. E., & Rutkowski, J. O. (2004). Entropy-based optimum test nodes selection for analog fault dictionary techniques. *The IEEETransactions on Instrumentation and Measurement, 53,* 754–761.

Sara Saeedi was born in Tehran, Iran in 1990. He received B.Sc. degree in Electrical Engineering in 2014. He is a graduate student in electrical engineering since 2016 in West Tehran Branch, Islamic Azad University, Tehran, Iran. He was defended his thesis on the field of 'The Optimal Method to Reduce the Required Computation in Analog Fault Dictionary Technique' and this paper completes his dissertation.

Seyyed Hossein Pishgar was born in Langroud, Guilan, Iran in 1982. He received B.Sc. in Electrical Engineering from Guilan University, Guilan, Iran in 2005. Also, he received M.Sc. and Ph.D. in Electrical Engineering in Shahid Beheshti Univesity in 2007 and 2013, respectively. He is a member of Electrical Engineering Department of West Tehran Branch, Islamic Azad Univesity, Tehran, Iran as an assistant professor since 2011. His research interests include Analog-to-Digital and Digital-to-Analog Converters, Integrated Circuit Design, Low-Voltage and Low-Power Circuit Design, and Integrated Filter Design.

Mahdi Eslami received the B.Sc. degree in control system engineering from Tehran University, Tehran, Iran, in 1998, the M.S. in telecommunication engineering from Khaje Nasir University of Technology, Iran, in 2000, and Ph.D. degrees in telecommunication engineering from Amirkabir University of Technology, in 2007. Since June 2015, he has been with the Electrical Engineering Department, West Tehran Branch of Azad Islamic University, Tehran, Iran, where he is currently assistant professor. He holds One Iran patent. His current research interests include 5G wireless communications, NGN, Internet of Things (IOT).