COMPUTATIONAL PROBLEMS OF ELECTRICAL ENGINEERING

Vol. 7, No. 2, 2017

TESTING OF DIGITAL CIRCUITS BY CYCLIC CODES

Vasyl Semerenko, Oleksandr Roik
Vinnytsia National Technical University, Vinnytsia, Ukraine
vpsemerenko@ukr.net, roik-alex@mail.ru

Abstract: The application of error correction coding theory to the tasks of technical diagnostics is considered. Known methods of testing based on signature analysis allow detecting only the faults in the digital circuit under test (CUT). The purpose of the research is to provide the possibility of an exact localization of the faults in logic subcircuits within the CUT. In the proposed method, a full test T for testing the CUT is subdivided into an input test T_1 (supplied to the inputs of the CUT) and an output test T_2 of the expected signatures (recorded into a memory block). Tests T_1 and T_2 are interpreted as a set of information words and a set of check words of the cyclic Hamming code respectively and are generated by the encoder. The decoder decodes words from both tests simultaneously and searches for errors only in the test T_1. As a result, full burst errors in the information words of error correcting code are corrected, which is equivalent to the exact localization of the faults within the CUT.

Key words: faults diagnosis, error correcting coding, cyclic codes, burst errors, test.

1. Introduction

The main criterion for the effectiveness of computer technology, automation and communication is the adequacy of information which they process. The adequacy of information is largely dependent on the reliability of the hardware and the degree of data protection from distortion due to errors of various nature.

There are several ways to achieve this goal, for example, by means of technical diagnostics and error-correcting coding. These technical branches have much in common and joint use of them will help to solve the tasks.

2. Problem Statement

Let us introduce the digital circuit under test (CUT) consisting of a set of logic subcircuits realizing either switching function, or sequential function (finite automaton).

Let the CUT have m_a input checkpoints, m_b internal check points and m_y output checkpoints. While testing the CUT, sets are supplied to the CUT inputs (input checkpoints a_i) and the test results are read from the outputs of the logical subcircuits (internal checkpoints b_x) and CUT outputs (output checkpoints y_j).

As it follows from the fundamental results of information theory and the theory of CUT synthesis, the universal method of data protection consists in introduction of different types of redundancy [1]. In the error-correcting coding the information redundancy is used by introducing the checkwords, and in the CUT the structural (hardware or software) redundancy is applied with the help of additional check circuits [2], [3].

There are many differences between the error-correcting coding and the technical diagnostics, but still there are also many similarities. In these technical branches, two major check tasks are solved:

– determining the presence or absence of errors;

– finding error locations for their subsequent correction.

In the error-correcting coding, the first task is solved by using some error detection code, and the second task is solved with the help of the error correcting code.

In the technical diagnosis, the code check methods are used only for solving the first check task. These methods are known as signature analysis (SA) [4].

The essence of this method consists in the preliminary determination of the correct signatures (checksums) in internal and output test points of the CUT and comparing them with the actual signatures when checking the CUT.

In SA, the generator polynomials of Hamming codes are commonly used, which guarantees detection and correction of the single random errors and detection of short burst errors. The Reed-Solomon codes and the BCH codes (Bose–Chaudhuri–Hocquenghem codes) can slightly increase the correcting capability of the code, but their disadvantages are large latency and high redundancy [5].

Therefore, error-correcting codes have been successfully used only to check the computer memory [6] and there are essential difficulties to use them for the CUT of arbitrary logical structure.

All known methods of code checking of the CUT consider the errors only within the test sequence without

fault location in the CUT. The long multi-stage check procedure is required for the exact fault location.

Thus, it is necessary to have some algorithmic check procedure of the arbitrary CUT indicating the fault location inside the CUT up to the separate logic subcircuits. To solve this problem, a new mathematical model of fault diagnosis in the CUT with the help of the error correcting code is required.

3. The model of fault diagnosis in CUT with help of error correcting code

Let us first consider the theoretical basis of fault diagnosis in the CUT with the help of the error correcting code. We will take into consideration only the errors causing logic inversion of variables.

In the technical diagnosis the stuck-at faults on lines in a logic circuit (stuck-at-0 or stuck-at-1) are the nearest to this error model [4], and in coding theory those are inverse errors (random errors or burst errors). The stuck-at fault model also covers most of other fault types.

We will consider the generators of deterministic or pseudo-random test patterns as a source of input data for the CUT [5]. Supplying the test vectors to the primary inputs of the CUT and receiving the verification result can be regarded as a data channel in which data is transmitted not in space but in time.

Full test T for the CUT is always subdivided into two parts: the input test T_{in} and the test of correct results T_{out}^{c}. Accordingly, the data channel can also be separated into two parts: the subchannel for transmitting T_{in} and subchannel for transmitting T_{out}^{c}.

These subchannels have three significant differences:

– structural (subchannel D_{in} is the CUT, subchannel D_{out} is RAM for storing correct test results);

– temporary (at first the correct test results are recorded in the RAM, and then the test begins),

– degree of reliability (subchannel D_{out} has rather higher reliability than the subchannel D_{in}).

The distortions in the data due to faults in the CUT are possible only in subchannel D_{in}, that is equivalent to the effect of noise in the usual communication channel.

Due to the faults in the CUT on the output of subchannel D_{in} the test T_{out}^{err} will be obtained.

For analysing the results of the CUT test, a special unit similar in function to the decoder in a communication channel is required.

Similar units are also used in the known SA method. The main difference of the proposed code check method is that these tests are interpreted differently.

In the proposed method, the test is a set of vectors of length k_w consisting of the correct logical values in the input, output and in internal checkpoints of the CUT. Each of such vectors is regarded as the information word J_w of cyclic (n_w, k_w) code, and the task of the encoder consist in the calculation of the checkwords Y_w of length ($n_w - k_w$) for them.

The checkwords are recorded to the RAM before the start of testing. During the testing process, only a part of the word J_w, namely, test vectors for input check points of the CUT (test T_{in}) come to the CUT input from the encoder.

The decoder receives the information word (consisting of correct values in the input checkpoints and the actual values in the internal and output checkpoints) and the corresponding checkword from the RAM.

The decoder calculates the error syndrome by a standard procedure. If the error syndrome is not equal to zero, the procedure of finding the fault subcircuit in the CUT is performed.

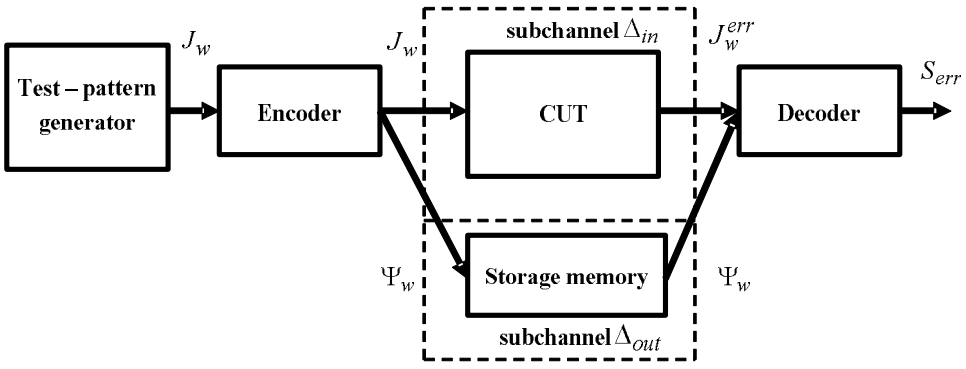


Fig.1. Model of a data transmission system with CUT checking coded by cyclic codes.

As a result, the model of the system for the data transfer with the CUT testing with the use of error-correcting coding is designed in the following way (Fig. 1).

The majority of the methods of test generation are based on the method of paths sensitization [5]. This method consists in propagating the effect of a possible fault along paths from the primary input (input checkpoints) to the primary output (output checkpoints). At the beginning a direct path $h_i = a_i \to y_j$ is sensitized from the input checkpoint a_i to a point y_j, and then reverse paths h_{rev} are sensitized from the point y_j and the internal checkpoints being on the path η_i to other primary inputs (input checkpoints) of the CUT (Fig. 2).

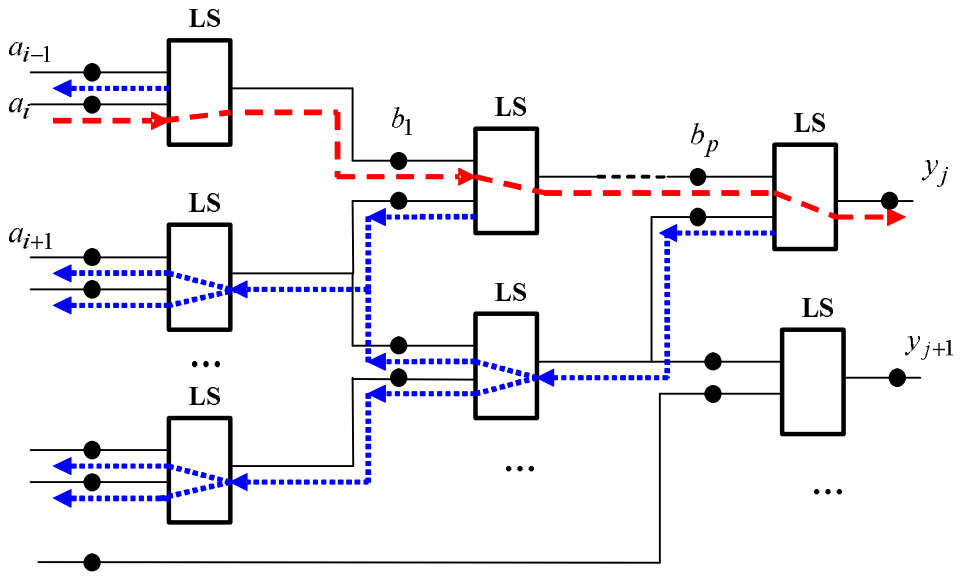


Fig. 2. Fault diagnosis in CUT based on the method of path sensitization (LS – logic subcircuit – direct path; – reverse paths).

If the CUT consists only of the logical switching subcircuits, then only two test vectors are enough for checking stuck-at faults in the path $h_i = a_i \to y_j$ and they differ only in logical values at the point a_i. If there are sequential logic subcircuits, then the number of test vectors can be more than two.

As a result, a test T_{in} to check the CUT will contain the test vectors of length k_w to check the paths from all input checkpoints. Each test vector t_i from test T_{in} contains the binary values for h input checkpoint ($h \le m_a$) and p internal checkpoints ($p \le m_b$) belonging to verified path $h_i = a_i \to y_j$ and one output checkpoint y_j i.e. the test vector t_i will be set in this format

$$a_1 \mathbf{K} a_i \mathbf{K} a_h b_1 \mathbf{K} b_x .... b_p y_j .$$

As it was already noted, the test vector t_i is regarded as the information word J_w of cyclic (n_w, k_w)-code W. When the previously calculated checkword Y_w is added on the left of it, the codeword Z_w of length n_w of cyclic code W will be obtained.

Now, let us consider the influence of faults in the CUT according to the type of errors in the code W.

If the fault is detected only at the output checkpoint y_j, it will be equivalent to the error in the rightmost digit of the information word J_w.

If the fault is detected on the input checkpoint a_i, then the whole of the path $h_i = a_i \to y_j$ will be functioning incorrectly. Therefore, all the digits of the word J_w will be erroneous.

Finally, if the faults are in the internal checkpoints of path $h_i = a_i \to y_j$ beginning from point b_x, then $(p - x + 2)$ right digits of the word J_w beginning from output y_j will be erroneous.

As the theory of cyclic codes states, the abovementioned types of errors are covered by a single

type of error: the full burst errors of length from 1 to k_w [7]. The information word J_w contains the full burst errors of length k_i if there is a continuous sequence of k_i erroneous digits.

In general, cyclic full burst errors can differ in their length and location in the word J_w.

In our case, only full burst errors of length no more than k_w with the beginning in the first (rightmost) digit of the word J_w are possible. Let us call them the fixed full burst errors.

4. Finding the faults in CUT with the cyclic codes

Let us consider fault diagnosis in the CUT as the decoding task of cyclic codes. There are different subclasses of cyclic codes and their choice depends on the required diagnosis accuracy. In the simplest case, it is sufficient to establish the presence or absence of the errors. The cyclic Hamming codes perfectly cope with this task. The generator polynomials of these codes are also used in the SA method.

For the exact location of erroneous digits in the codeword Z_w, hence, for finding failures in the CUT, more powerful codes are required. However, with the growth of the correcting code capability, the redundancy of (k,n)-code proportionally increases [8]. For these codes, $(n-k)\approx k$, i.e. when it is used in technical diagnostics, the size of stored correct test information T_{out}^{c} will be almost equal to its input test T_{in}.

THEOREM. The cyclic Hamming (k,n)-code allows to correct fixed full burst errors of length from 1 to $n/2$.

Proof. The proof of this theorem is based on the analysis of the graphical model of cyclic code [9]. As it was shown in [10], to localize an arbitrary full burst error of length less than $n/2$ the graph model (k,n) code should contain $n/2$ zero cycles (ZC) of length n.

Thus the *i*-th ZC permits to diagnose the full burst error which starts from *i*-th digit of the codeword Z_w. The fixed full burst error always starts in the first digit of the codeword Z_w, so for this type of error it is enough to have one ZC of length n. The cyclic Hamming (k,n) code has exactly such a ZC.

If all test paths in the CUT contain different logical subcircuits, then the presence of full burst errors in *i*-th code word Z_w will uniquely indicate a fault with the *i*-th subcircuit. If logical subcircuit belongs to several test paths, then additional analysis is required to clarify the location of the fault. For complicated subcircuits of automaton type, the additional test vectors may be required.

5. Conclusions

Known error correcting codes can be used to detect and correct errors in the CUT of the arbitrary logical structure. To increase the number of detected and corrected errors there is no need to use the powerful codes, such as Reed-Solomon codes. It is possible to choose very simple codes such as the Hamming codes having low redundancy and, correspondingly, small size of stored test data (similar to the SA method). However, at the same time, two conditions must be satisfied:

– to carry out the reconfiguration of the checkpoints in the CUT, so that they might be placed in the order of passing the test signals along them (to form the test path);

– to use a fixed full burst error as an error model.

As opposed to known methods of checking a digital device [5] correcting only single and double errors, the method proposed here makes it possible to correct the full burst errors of length from 1 to $n/2$ in the CUT.

The considered error models precisely describe the faults in modern semiconductor memory devices [11].

References

[1] E. Dubrova, *Fault Tolerant Design: an Introduction*. Boston, USA: Kluwer Academic Publishers, 2008.

[2] C. Shannon, *A mathematical theory of communication.* Bell Syst. Tech. J., vol. 27, pp. 379–423, 623–656, 1948.

[3] M. Gavrilov, "Structural redundancy and functional reliability of switching units", in *Proc. First World Congress IFAC*, vol. 3, Moscow, USSR: Academy of Science, 1960. (Russian).

[4] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures Design for Testability*. New York, London: Morgan Kaufmann Publishers, 2006.

[5] A. Babitha and S. Divya. "Modified Hamming Codes with Double Adjacent Error Correction along with Enhanced Adjacent Error Detection", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, issue 8, pp.7706-7713, Aug. 2015.

[6] E. Fujiwara, *Code Design for Dependable Systems.Theory and Practical Applications.* USA: John Willy & Sons, Inc., 2006.

[7] V. Semerenko, "Burst-Error Correction for Cyclic Codes", in *Proc. Int. IEEE Conference EUROCON-2009*, S. Petersburg, Russia: May 2009, pp. 1646–1651. *DOI: 10.1109/eurcon.2009.5167864*

[8] X. Tang and S. Wang, “A Low Hardware Overhead Self-Diagnosis Technique Using Reed-Solomon Codes for Self-Repairing Chips”, *IEEE Transactions on Computers*, vol. 59, no. 10, pp. 1309–1319, October 2010.

[9] V. Semerenko, “Parallel Decoding of Bose-Chaudhuri-Hocquenghem Codes”, *Engineering Simulation*, vol. 16, no. 1, pp. 87–100, Jan.1998.

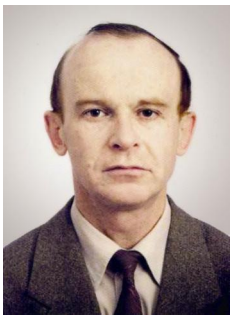
[10] V. Semerenko, “Estimation of the correcting capability of cyclic codes based on their automation models”, *Eastern-European Journal of Enterprise Technologies*, no. 2/9 (74), pp. 16–24, 2015. (Russian).

[11] A. Neale and M. Sachdev, “A New SEC-DED Error Correction Code Subclass for Adjacent MBU Tolerance in Embedded Memory”, *IEEE Transactions on Device and Materials Reliability*, vol. 13, no. 1, pp. 223, 230, March 2013.

ТЕСТУВАННЯ ЦИФРОВИХ СХЕМ ЗА ДОПОМОГОЮ ЦИКЛІЧНИХ КОДІВ

Василь Семеренко, Олександр Роїк

Розглянуто застосування теорії завадостійкого кодування до завдань технічної діагностики. Відомі методи тестового контролю на основі сигнатурного аналізу дають змогу встановлювати тільки факт наявності або відсутності помилок у ЦУ. Мета дослідження полягає в забезпеченні можливості точної локалізації помилки в логічних підсхемах всередині ЦУ, який перевіряється. У пропонованому методі повний тест Т для перевірки ЦУ, який формується кодером циклічного коду Хеммінга, підрозділяється на вхідний тест T_1 (подається на входи ЦУ) і тест T_2 еталонних вихідних значень (записується в блок пам’яті). Тест T_1 інтерпретується як безліч інформаційних слів циклічного коду, а тест T_2 – як безліч перевіряючих слів циклічного коду. Декодер спільно декодує тест T_1 і T_2 і шукає помилки тільки в тесті T_1. У результаті виправляються щільні пакети помилок в інформаційних словах перешкодостійкого коду, що еквівалентно точної локалізації помилок усередині ЦУ.



Vasyl Semerenko – Engineer, Ph. D. He holds the positionof Associate Prof. at Department of Computer Technologies in Vinnytsia National Technical University, Ukraine. His present research interests include error-correcting codes, parallel data processing, technical diagnostics.

Fellow of IEEE Computer Society.

Oleksandr Roik – Engineer, Ph.D., D. Sc. He hoplds the position of Professor and Head of the Department of Management and Security of Information Systems in Vinnytsia National Technical University, Ukraine. His present research interests include technical diagnostics and security of information systems.