VLSI DESIGN
2000, Vol. 00, No. 00, pp. 1–12
Reprints available directly from the publisher
Photocopying permitted by license only

Self-checking Synchronous FSM Network Design with Low Overhead

A. Yu. MATROSOVA[a,*], I. LEVIN[b,†] and S. A. OSTANIN[a,‡]

[a]*Tomsk State University, Russia;* [b]*Tel Aviv University, Israel*

(Received 1 April 1999; In final form 5 October 1999)

A method of a self-checking synchronous Finite State Machine (FSM) network design with low overhead is developed. Checkers are used only for FSMs, which output lines are at the same time output lines of the network. The checkers observe output lines of these FSMs. The method is based on reducing the problem to a self-checking synchronous FSM design. The latter is provided by applying a special description of FSM namely, so-called unate Programmable Logic Array (PLA^u) description. Single stuck-at fault on the FSM poles and gate poles are considered. PLA^u realization of FSM allows a factorized or multilevel logic synthesis. They both provide a unidirectional manifestation of the above mentioned faults on the output lines of the corresponding FSMs. This realization also gives rise to a transparency of each component FSM of the network for the faults. PLA^u realization is derived from the State Transition Graph (STG) description of FSMs with using the *m*-out-of-*n* encoding of its states and insignificant expanding the products of STG. The problem of replacing an arbitrary synchronous FSM network for the self-checking one with low overhead is discussed.

Keywords: Finite State Machine, self-checking, FSM network

1. INTRODUCTION

A self-checking circuit usually consists of a functional block that generates encoded outputs, and a checker that checks the validity of the outputs [1–3]. In the case when the functional block is a Finite State Machine (FSM), concurrent checking is usually based not only on outputs checking but also on checking of the FSM transitions. For example, the approach proposed in paper [4] is based on a specific decomposition architecture where state transitions checking only guarantees both totally self-checking (TSC) property of the FSM, and error detection latency of one clock cycle. In paper [5] not only state and output variables, but also input variables are checked.

*e-mail: mau@fmpk.tsu.ru
†Address for correspondence: Tel Aviv University, School of Education, Ramat-Aviv 69978, Israel. Tel.: 972-3-6407799, Fax: 03-5026733, e-mail: ilia1@post.tau.ac.il
‡e-mail: ostanin@fpmk.tsu.ru

The paper [5] suggests the use of pre-designed code-disjoined flip–flops to receive high fault coverage, particularly including state register and clock errors detection.

Paper [6] shows a possibility of observing only output lines of a self-checking synchronous sequential circuit, without its state lines. Detection of single stuck-at faults of the PLA realization of synchronous sequential circuits (SSC), except for single stuck-at faults on the input lines of the SSC, has been assumed there.

Our intention here is to design a self-checking synchronous FSM network so that checkers are used only for FSMs which output lines are at the same time the network output lines. Moreover, each checker observes the values of the FSM output lines but not both state and output lines as it was done in paper [7].

In this paper, the self-checking FSM network design reduces to its self-checking component design that is the specific self-checking synchronous FSM design. Realizing FSM as a sequential circuit, we use m-out-of-n codes for the FSM states assignment and insignificantly increase the number of FSM input variables. We assume that any FSM is preliminarily described by State Transition Graph (STG) and derive a special Programmable Logic Array (PLA) description from STG, namely a so-called "unate PLA" description (PLA^u). The PLA^u is a standard PLA description, where all "0" values are replaced with "don't care" symbols. The PLA^u represents the system of unate Boolean functions. Having applied either factorizing or multilevel logic synthesis to this system, we obtain the synchronous sequential circuit (SSC) that is a structural description of the self-checking FSM.

It is shown in paper [7] that for a specific fault model, which will be discussed below, the faults manifest themselves as unidirectional errors on the SSC combinational part output lines. They can be undetectable on these output lines in the working area of a separate FSM. The authors of [7] proposed the m-out-of-n encoding of state and output variables of FSM separately, and the observing of all these variables by a checker. To provide the self-checking synchronous FSM network design, they insignificantly increase the number of input variables of each FSM of the network. Each FSM has its own checker.

The present paper is an attempt to combine the extended class of single stuck-at faults considered in [7] with observing only output lines of SSC considered in [6]. It turned out in [8] that either factorizing or multilevel logic synthesis can be applied to the unate PLA (PLA^u) description without loss of manifesting the above faults as unidirectional errors. As a result we obtain SSC with the following properties:

- Any above fault is either unidirectional or undetectable on the SSC output lines.
- The accumulating of undetectable faults in SSC is not dangerous for next faults from the above-mentioned class.
- SSC is transparent for unidirectional errors on its inputs. It means that SSC manifests these errors on the own output lines either as unidirectional or undetectable faults in the FSM working area.
- SSC preserves the transparency property in the presence of own undetectable faults.

Methods of decomposition of a large FSM into the FSM network were discussed in [9, 10]. We recommend during decomposition to reduce the number of the additional input variables of each FSM and decrease the number of output codewords of FSM which output lines are at the same time the network output lines. The latter allows reducing the overhead through using Sum-of-Minterms based (SOM-based) checkers [6]. We also propose a method of replacing the given arbitrary synchronous FSM network by the self-checking synchronous FSM network with low overhead. The price must be paid in changing the STG description of a separate FSM for the unate PLA description (PLA^u).

First (Section 2), we consider the problem of deriving PLA^u from STG. Then (Section 3), we investigate the properties of SSC faults on the assumption that SSC realizes PLA^u. In Section 4 the problem of replacing an arbitrary synchronous

FSM network by the self-checking synchronous FSM network with low overhead is considered.

2. FAULT MODEL

Any gate pole single stuck-at fault or input line single stuck-at fault of the combinational part of a synchronous sequential circuit (SSC) leads to a unidirectional error on output lines of the combinational part. It is also possible that the fault will be undetectable on these lines.

A single stuck-at fault on a delay flip–flop pole of a synchronous sequential circuit manifests itself as a single stuck-at fault of the corresponding input line of the combinational part. This line correlates with the state variable of the synchronous sequential circuit.

A single stuck-at fault of an input pole of the certain FSM also manifests itself as a single stuck-at fault of the corresponding input line of the combinational part of the SSC. A single stuck-at fault of an output pole of the certain FSM manifests itself as a single stuck-at fault of the corresponding input lines of the FSMs (SSCs) connected with the FSM considered.

A fault considered can manifest itself as unidirectional error on the FSM network output lines in the network working area. A fault is undetectable if it does not manifest itself on the network output lines in the network working area. The above mentioned faults can be either unidirectional or undetectable. We assume that the next fault of this kind may appear after the FSM network's working area with a foregoing fault is exhausted. The foregoing fault, being unidirectional, has to manifest itself on the network output lines within the working area.

3. DERIVING UNATE PLA DESCRIPTION FROM STG

Divide FSMs of an arbitrary network into two groups. The first group comprises FSMs so that some of their output lines are at the same time the output lines of a network. Call them external FSMs. The second group comprises the rest FSMs of a network. Call them internal FSMs.

Let the states of any FSM of the network be encoded with codewords of the same weight. Then the STG description of FSM converts into the PLA description.

We propose to implement the self-checking design of an external FSM using the basic scheme shown in Figure 1.

According to this scheme, SSC to be checked consists of three portions: the output functions portion (its outputs are $y_1, \ldots, y_m$), the redundancy portion (its outputs are $y_{m+1}, \ldots, y_s$) that provides encoding SSC outputs and the transition functions portion (its outputs are $z_1, \ldots, z_p$). The transition functions portion represents the next states of FSM that are encoded by the constant weight codes. The output functions portion and the redundancy portion together form codewords that are Berger Codes.

SSC (FSM) of the second group consists of two portions: the output functions portion and the transition functions portion. We believe that output codewords of FSM of the second group is free of the codeword consisting of only 0 components.

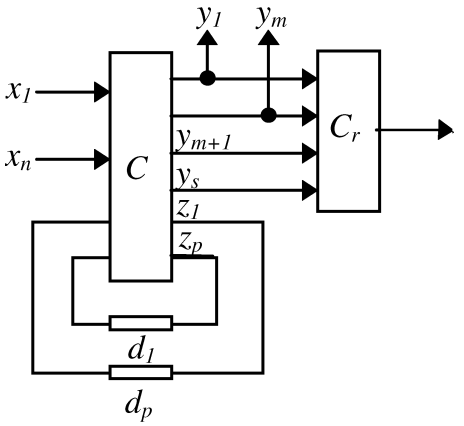

FIGURE 1 The basic scheme of a self-checking external FSM.

Products u_i, u_j from PLA are bidirectional if there is the component l taking the 1 value for u_i and the 0 value for u_j, and the component r taking the 0 value for u_i and the 1 value for u_j. For example, $-10-0$, $-00-1$ are bidirectional.

Let U be the set of products of PLA. Any product depends on the input and the state variables. The state variables represent the encoded state of FSM. Divide U into subsets $U_1, \ldots, U_{|Q|}$ in accordance with the different states of FSM. Here, $|Q|$ is the number of states.

Let STG for the internal FSM be described by Table I.

After encoding states we obtain the PLA description (Tab. II).

The PLA products are divided into the subsets in accordance with different states of FSM.

THEOREM 1 *Products u_i, u_j are bidirectional for $u_i \in U_i$, $u_j \in U_j$, $i \neq j$.*

TABLE I STG description of FSM

x_1	x_2	x_3	q	q	y_1	y_2	y_3	y_4	y_5
0	–	–	1	1	0	0	0	1	0
–	0	–	1	1	0	0	0	1	0
1	1	–	1	2	1	0	0	1	0
–	–	0	2	2	0	0	1	1	0
–	–	1	2	3	1	0	1	1	0
1	0	–	3	3	0	1	0	0	0
0	–	–	3	4	1	1	0	0	0
–	1	–	3	4	1	1	0	0	0
–	–	0	4	4	0	1	0	0	1
–	–	1	4	1	1	1	0	0	1

TABLE II PLA description of FSM

$x_1x_2x_3$	$z_1z_2z_3z_4$	$z_1z_2z_3z_4$	$y_1y_2y_3y_4y_5$
0 – –	1 0 0 0	1 0 0 0	0 0 0 1 0
– 0 –	1 0 0 0	1 0 0 0	0 0 0 1 0
1 1 –	1 0 0 0	0 1 0 0	1 0 0 1 0
– – 0	0 1 0 0	0 1 0 0	0 0 1 1 0
– – 1	0 1 0 0	0 0 1 0	1 0 1 1 0
1 0 –	0 0 1 0	0 0 1 0	0 1 0 0 0
0 – –	0 0 1 0	0 0 0 1	1 1 0 0 0
– 1 –	0 0 1 0	0 0 0 1	1 1 0 0 0
– – 0	0 0 0 1	0 0 0 1	0 1 0 0 1
– – 1	0 0 0 1	1 0 0 0	1 1 0 0 1

Any product of PLA has the corresponding full codeword representing the values of the next state and output variables.

For example, product $\begin{smallmatrix} x_1x_2x_3 \\ -0- \end{smallmatrix}$ $\begin{smallmatrix} z_1z_2z_3z_4 \\ 1000 \end{smallmatrix}$ from the second line of the Table II has the full codeword $\begin{smallmatrix} z_1z_2z_3z_4 \\ 1000 \end{smallmatrix}$ $\begin{smallmatrix} y_1y_2y_3y_4y_5 \\ 00010 \end{smallmatrix}$.

Execute the following steps.

1. Divide the set U_i into subset $U_{i1}, \ldots, U_{i\tau_i}$ in accordance with their different full codewords, $i \in \{1, \ldots, |Q|\}$.
2. Correlate the different input codewords of the same weight to the different $U_{i\gamma}$, $\gamma \in \{1, \ldots, \tau_i\}$.
3. Represent any input codeword with the proper Boolean vector α^* of the length k in additional input variables.
4. Add the same α^* to each product u_i, $u_i \in U_{i\gamma}$.
5. Execute Steps 2–4 for every U_i, $i \in \{1, \ldots, |Q|\}$, minimizing the number k as much as possible.

For the example considered we have Table III.

THEOREM 2 *Products u_k, u_s, $u_k \in U_{ik}$, $u_s \in U_{is}$ are bidirectional.*

Change each 0-value component from the products of PLA for symbol "–" (don't care). We obtain the unate products of PLA that is the unate PLA description. Notice it as PLA^u.

THEOREM 3 *PLA^u preserves the FSM behavior in its working area.*

Proof The PLA description derived from STG represents the FSM behavior. Any minterm of the

TABLE III Introduction of additional input variables into PLA description

$x_1x_2x_3x_4x_5$	$z_1z_2z_3z_4$	$z_1z_2z_3z_4$	$y_1y_2y_3y_4y_5$
0 – – 0 1	1 0 0 0	1 0 0 0	0 0 0 1 0
– 0 – 0 1	1 0 0 0	1 0 0 0	0 0 0 1 0
1 1 – 1 0	1 0 0 0	0 1 0 0	1 0 0 1 0
– – 0 0 1	0 1 0 0	0 1 0 0	0 0 1 1 0
– – 1 1 0	0 1 0 0	0 0 1 0	1 0 1 1 0
1 0 – 0 1	0 0 1 0	0 0 1 0	0 1 0 0 0
0 – – 1 0	0 0 1 0	0 0 0 1	1 1 0 0 0
– 1 – 1 0	0 0 1 0	0 0 0 1	1 1 0 0 0
– – 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0 1
– – 1 1 0	0 0 0 1	1 0 0 0	1 1 0 0 1

input and state variables from the FSM working area activates the products of PLA with the same full codeword. This minterm lengthened with the proper additional input variables activates the corresponding products of PLA^u with this full codeword. Consequently, the minterm gives rise to the same full codeword both for PLA and PLA^u. Q.E.D.

The unate PLA description allows applying both the factorizing and the multilevel logic synthesis [8]. We obtain minimized description of the PLA^u products with the same full codeword. Namely for any such product there exists a minterm from PLA that is covered only by this product. Hereafter we will use the minimized PLA^u description of FSM.

For the above-mentioned example, we have minimized PLA^u description represented by a Table IV.

Having obtained the PLA^u description (Tab. IV) we need to correct the input sequences built from the given STG. The correction reduces to adding the proper values of the additional input variables, using Table III.

Illustrate it by the following example. Let 1 be an initial state of the STG description (Tab. I) and we have the input sequence 000, 001, 111, 111, 101, 000. It gives rise to the sequence of the state: 1 1 1 2 3 3 4. The input sequence has to be corrected (Tab. III) as follows: 00001, 00101, 11110, 11110, 10101, 00010. This sequence arrives at the SSC that realizes the PLA^u description given in Table IV.

TABLE IV PLA^u description

$x_1 x_2 x_3 x_4 x_5$	$z_1 z_2 z_3 z_4$	$z_1 z_2 z_3 z_4$	$y_1 y_2 y_3 y_4 y_5$
– – – – 1	1 – – –	1 0 0 0	0 0 0 1 0
1 1 – 1 –	1 – – –	0 1 0 0	1 0 0 1 0
– – – – 1	– 1 – –	0 1 0 0	0 0 1 1 0
– – 1 1 –	– 1 – –	0 0 1 0	1 0 1 1 0
1 – – – 1	– – 1 –	0 0 1 0	0 1 0 0 0
– – – 1 –	– – 1 –	0 0 0 1	1 1 0 0 0
– – – – 1	– – – 1	0 0 0 1	0 1 0 0 1
– – 1 1 –	– – – 1	1 0 0 0	1 1 0 0 1

It is possible to minimize the number k of additional input variables, applying different sophisticated algorithms.

4. PROPERTIES OF THE SSC FAULTS

We believe that SSC is derived from PLA^u either factorized or multilevel logical synthesis. According to the basic scheme (Fig. 1), we cannot observe the values of state variables. We consider gate pole single stuck-at faults and input line single stuck-at faults of the combinational part of SSC. Call them as T.

Any fault from T either appears [8] as unidirectional error on the combinational part output lines (state lines and output lines of SSC), or remains undetectable on these output lines.

A fault is detectable (for SSC), if there exist the input sequence (in the working area of FSM) for which the fault manifests itself as error on the SSC output lines. We restrict the sequence to the first manifestation. Otherwise the fault is undetectable.

If a fault manifests itself as a unidirectional error on the SSC output lines, it is unidirectional. We will distinguish unidirectional manifestations as follows. If a unidirectional manifestation of a fault (for the certain minterm of the input and state variables) reduces to changing the certain 1-values for the 0-values on the SSC output lines, it is a 0-unidirectional manifestation. If it is reduces to changing the certain 0-values for the 1-values, it is a 1-unidirectional manifestation.

A certain fault can manifest itself on the state lines of SSC as a unidirectional error and remain undetectable on the SSC output lines in the working area of FSM. It is also possible that a certain fault appears as unidirectional error on the state lines of SSC several steps before than on its output lines.

What will happen, when a next fault from T appears in SSC but a foregoing fault from T is undetectable? In this case we simultaneously deal with two faults from T. It is also possible that

there exist several undetectable faults from T and a next fault from T appears.

We have to investigate the problem of accumulating undetectable faults. First, we have to study the manifestations of single stuck-at faults from T, taking into consideration that the state lines are not observable. A fault from T gives rise to the following:

(a) disappearance of the certain products from PLA^u, or
(b) disappearance of the certain literals from the certain products of PLA^u, or
(c) conversion of the certain functions of the unate system of Boolean functions of PLA^u into a constant 1(0).

The following Theorems 4–6 will prove that a fault is either undetectable on SSC output lines or unidirectional. The proof will be based on determining different manifestations of a single constant fault on a specific gate (i, ii, iii) for any method of synthesis preserving the PLA^u.

Keeping in mind the same manifestations we further prove that accumulation of undetected faults is harmless (Theorems 7–10). Specific features of the PLA^u (both properly and improperly working) can explain such properties and these features have been used when proving the theorems. In other words any fault or multiple fault will not cause appearance of negation in the PLA^u. Every next single fault occurs only after it is determined that the previous fault is undetectable.

THEOREM 4 *A fault that gives rise to disappearance of the certain products from PLA^u is* 0-*unidirectional.*

Proof In fact, there exists the minterm from the FSM working area that activates the only product of PLA^u. (We use the minimized PLA^u description.) It takes place for each product from PLA^u. If the activated product disappears, we obtain the none-code state vector that immediately results in the output vector of SSC, which consists of only 0-values. Q.E.D.

If we don't minimize a portion of PLA^u with the same full codeword, the fault can be undetectable.

THEOREM 5 *A fault that gives rise to disappearance of the certain literals from the certain products of PLA^u is either* 1-*unidirectional or undetectable.*

Proof This fault is undetectable when any minterm α of the state and input variables from the FSM working area is among products of PLA^u with the same full codeword. Assume that, the certain minterm α activates simultaneously the products with the different full codewords. The fault is 1-unidirectional when the parts of these codewords corresponding to the output variables of SSC are different. If these parts are the same, we obtain the wrong next state vector β that contains more the 1-values than the corresponding truth state vector. It is possible that β does not result in any errors on the SSC output lines in the FSM working area. Then the fault is undetectable. The fault can be 1-unidirectional, when β results in the error on the SSC output lines in the FSM working area. Since the PLA^u products consist of only uncomplemented variables and β has the additional 1-values (in comparison with the truth state vector), then the number of the PLA^u products activated by a minterm resulted from β in the FSM working area can only increase. It means that the error is a 1-unidirectional one. Q.E.D.

THEOREM 6 *If a fault converts the certain transition functions into constant* 1(0) *it is either* 1(0)-*unidirectional or undetectable.*

Proof When the fault converts the certain transition functions into the constant 0 it can give rise to the none-code state vectors. Let the none-code state vector β be derived from the corresponding truth state vector α by changing the certain 1-values for the 0-values. The vector α activates the only PLA^u product k_i of the state variables (by the construction of PLA^u from PLA). The vector α is orthogonal to all other products of the state variables from PLA^u. The product k_i is derived

from α by changing all 0-values for symbol – when we obtain PLA^u from PLA. Consequently, β is orthogonal to k_i and all the more to other products of the state variables. It means that β gives rise to the none-code vector of the state variables with only 0-value components. The none-code state vector immediately forms the 0-unidirectional error on the SSC output lines.

In the case of conversion of the certain transition functions into the constant 1 we also can obtain the none-code state vectors. Let none-code state vector β be derived from the corresponding truth state vector α by changing the certain 0 values for the 1 values: β and α activate k_i. But β can also activate the other products of state variables. Consequently, the minterm resulted from β in the FSM working area can simultaneously activate the products from PLA^u with the different full codewords. It means the fault is either 1-unidirectional or undetectable. Q.E.D.

If the fault converts the certain output functions into the constant 1(0), it is 1(0)-unidirectional.

Manifestation of single stuck-at faults at input lines of SSC is of primary importance for the self-checking FSM network design. Even a repeated duplication of the sequential circuit does not ensure manifestation of single stuck-at faults on the common input lines of the sequential duplicates as a unidirectional error.

Taking into consideration the Theorems 4 – 6 we conclude that an undetectable fault is possible when any none-code state vector resulted from the fault does not manifest itself directly on the SSC output lines in the FSM working area. Any none-code state vector is obtained from the corresponding truth state vector by changing the certain 0 values for the 1 values. A fault is also undetectable when it does not manifest itself on the SSC both state and output lines in the FSM working area. In this case none-code state vectors are absent.

An individual none-code state vector (or a truth state vector) with the corresponding input minterm from the FSM working area call an undetectable portion of a fault, and notice as t. Consider t as the first fault of any pair of faults and check a second fault impact on the next step of the behavior in the working area. A second fault appears just after t arrives on the SSC combinational part input lines.

Let t^a indicate disappearance of the certain products from PLA^u.

THEOREM 7 *The pair* (t, t^a) *is either* 0-*unidirectional or undetectable during a next state in the FSM working area.*

Proof Disappearance of the certain products from PLA^u can (for t) results in appearance of the none-code state vector containing only 0-components. Then the pair is 0-unidirectional on the SSC output lines during a next state in the FSM working area. It is possible that the pair remains undetectable on the minterm t. In fact if t activates simultaneously several products from PLA^u and some of them disappeared then the rest products have the same full codeword. A first undetectable fault considering as a whole can only increase the number of 1-value for the next states followed by t. In this case the pair (t, t^a) also results in the undetectable fault during a next state in the FSM working area. Q.E.D.

Let t^b indicate disappearance of the certain literals from the certain products of PLA^u.

THEOREM 8 *The pair* (t, t^b) *is either* 1-*unidirectional or undetectable during a next state in the FSM working area.*

Proof The fault t^b can effect the additional 1 values among the corresponding next state vectors resulted from t in the FSM working area. A first undetectable fault considering as a whole can only increase the number of 1-value for the next states followed by t. Consequently, (t, t^b) is either unidirectional or undetectable during a next state in the FSM working area. Q.E.D.

Let t^{c_1} indicate conversion of the certain state variables into the constant 1.

THEOREM 9 *The pair* (t, t^{c_1}) *is either* 1-*unidirec-*

tional or undetectable during a next state in the FSM working area.

Proof The fault t^{c_1} can effect the additional 1 values in the corresponding next state vectors resulted from t in the FSM working area. A first undetectable fault considering as a whole can only increase the number of 1-values for the next states followed by t. Consequently, (t, t^{c_1}) is either unidirectional or undetectable during a next state in the FSM working area. Q.E.D.

Let t^{c_0} be a conversion of the certain state variables into the constant 0.

THEOREM 10 *The pair* (t, t^{c_0}) *is either* 0*-unidirectional or undetectable during a next state in the FSM working area.*

Proof The fault t^{c_0} can cut the number of 1 values in the corresponding next state vector resulted from t in the FSM working area. This vector can become the none-code state vector that is orthogonal to any state product of PLA^u and then (t, t^{c_0}) is 0-unidirectional. This vector can activate the products with the same full codeword. A first undetectable fault considering as a whole can only increase the number of 1-values for the next states followed by t. In this case the fault is undetectable during a next state in the FSM working area. Q.E.D.

We have shown that if a first fault from T is undetectable, then appearance of a next fault from T is either unidirectional or undetectable. We believe that any next fault from T appears in the SSC after exhausting the FSM working area in the presence of a foregoing fault from T. The foregoing fault being detectable has to manifest itself as unidirectional error on the SSC output lines. It means that accumulating undetectable faults from T in SSC is not dangerous.

It is a very important property of SSC.

Moreover, any unidirectional fault preserves the unidirectional property. It is impossible that a fault from T for the minterm t_1 manifests itself as a 1-unidirectional error but for the minterm t_2 – as a 0-unidirectional error in the FSM working area.

Consider a fault t_0^{in}, which results in 0-unidirectional errors on the certain SSC input vectors (minterms) from the FSM working area. This fault is out of T, and SSC is functioning properly.

THEOREM 11 *A fault* t_0^{in} *is either* 0*-unidirectional, or undetectable.*

Proof The fault changes some 1-value input components of the truth minterm α in the FSM working area for the 0-values. The truth minterm α activates several products from PLA^u with the same full codeword. The additional 0-values can result in the situation when only the certain of these products are activated. Then the fault is undetectable. If there are no the activated products the fault is 0-unidirectional during the next state in the FSM working area. Q.E.D.

Consider a fault t_1^{in}, which results in 1-unidirectional errors on the certain SSC input vectors (minterms) in the FSM working area. This fault is out of T; an SSC is functioning properly.

THEOREM 12 *A fault* t_1^{in} *is either* 1*-unidirectional, or undetectable.*

Proof The fault changes some 0-value input components of the truth minterm α in the FSM working area for the 1-values. Then the number of activated products from PLA^u can increase and consequently, the fault is either 1-unidirectional or undetectable during the next state in the FSM working area. Q.E.D.

We will call the SSC property described by the Theorems 11, 12 a transparency property.

Show that SSC preserves the transparency property in the presence of an undetectable fault from T.

THEOREM 13 *The pair* (t, t_0^{in}) *is either* 0*-unidirectional or undetectable during a next state in the FSM working area.*

Proof The minterm t activates one or several

products from PLA^u with the same output codeword. The fault t_0^{in} can change some 1-value input components of t for the 0-values. As a result, only the certain of these products can be activated. Then the fault (t, t_0^{in}) is undetectable. If there is no activated products, the fault is 0-unidirectional during the next state in the FSM working area. Q.E.D.

THEOREM 14 *The pair* (t, t_1^{in}) *is either* 1-*unidirectional or undetectable*.

Proof The fault t_1^{in} can change some 0-values of input components of t for the 1-values. It can increase the number of activated products from PLA^u and consequently, the fault (t, t_1^{in}) is either 1-unidirectional or undetectable during the next state in the FSM working area. Q.E.D.

Stand out the basic properties of SSC on assuming that only SSC output lines are observable.

- Any fault from T is either unidirectional or undetectable on SSC output lines and preserves the unidirectional property.
- The accumulating undetectable faults from T in SSC is not dangerous for a next fault from T.
- SSC is transparent to unidirectional errors on its input lines. Namely, 1(0)-unidirectional errors on the input lines give rise to 1(0)-unidirectional errors on the output lines of SSC.
- SSC preserves the transparency property in the presence of undetectable faults from T.

It is important to note that generally, error detection latency of the proposed PLA^u based self-checking SSC is equal of one clock cycle. Only in a very rare case of a faulty transition of the FSM into two different states having the same output codeword (which case is caused by a fault in the next state portion) will lead to increase of the latency. In most cases the latency of the proposed architecture is equal to the latency of the classical architecture where both output and state variable are checked.

Let's try to use above-mentioned properties of SSC for designing a self-checking FSM network with low overhead.

5. SELF-CHECKING FSM NETWORK DESIGN

We have an arbitrary FSM network N on the assumption that any FSM is described with STG. The object is obtaining the following self-checking synchronous FSM network N^S:

- Checkers are used only for external FSMs.
- Any checker observes only the output lines of the external FSM that are at the same time output lines of N^S.
- The only unidirectional fault from T is available for the only component of N^S at the same moment of time. If all components are fault free the only FSM pole can be stuck-at fault.
- The STG description of any FSM from N is changed for the PLA^u description.

Let us show that N^S preserves the behavior of N.

THEOREM 15 *The network* N^S *preserves the behavior of a network* N.

Proof Consider a portion of the working area of a network N. The portion is represented by the network input minterm consisting of the corresponding FSMs input minterms and the network state minterm consisting of the corresponding FSMs state minterms and the state minterm of the global loops. Call it α. The minterms α give rise to the full code words of FSMs that form full codeword γ of a network N. Changing PLA for PLA^u we only lengthen α through the proper additional input variables of the separate FSMs. It does not change corresponding full codewords of FSMs and consequently the full codeword γ of a network. It means N^S preserves the behavior of N. Q.E.D.

Figure 2 illustrates the network N^S. The dotted lines point to the additional inputs and checkers.

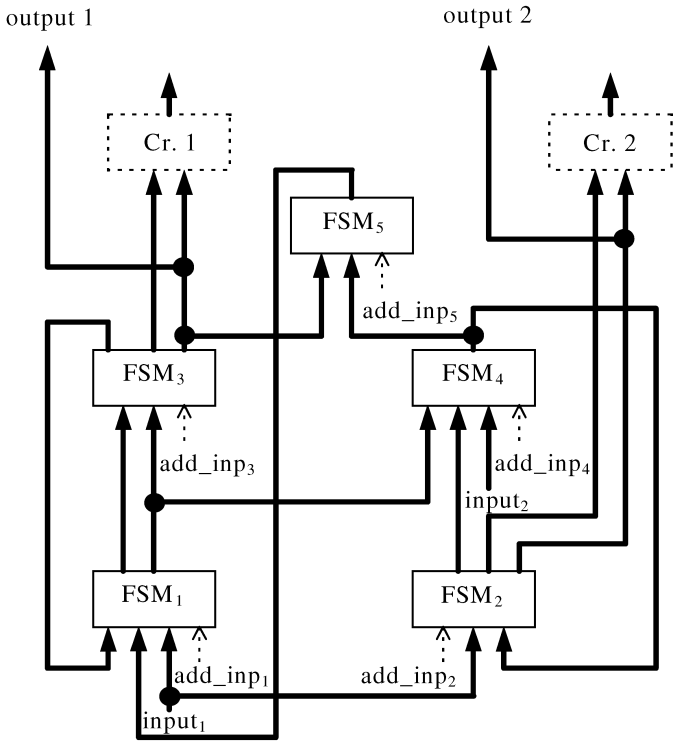

FIGURE 2 Self-checking synchronous FSM network.

TABLE V Experimental results

	I				II			III				IV	
	n	m	s	l	p	Σ_1	Σ_2	n_d	p	Σ_1	Σ_2	m_d	Σ_2
BBARA	4	2	10	60	4	410	89	4	5	270	128	3	240
BBSSE	7	7	16	56	4	350	174	4	6	317	219	4	330
BBTAS	2	2	6	24	3	120	35	2	4	90	52	3	78
BEECOUNT	3	4	7	28	3	152	79	3	5	106	112	3	168
CSE	7	7	16	91	4	631	132	5	6	576	309	4	463
DK14	3	5	7	56	3	336	154	3	5	245	213	3	342
DK15	3	5	4	32	2	160	80	3	4	108	88	3	162
DK16	2	3	27	108	5	756	305	2	7	513	412	3	660
DK27	1	2	7	14	3	56	24	1	5	42	34	3	56
DK512	1	3	15	30	4	150	65	1	6	90	73	3	150
DONFILE	2	1	24	96	5	672	304	2	7	456	384	3	480
KEYB	7	2	19	170	5	1344	130	7	6	1136	524	3	712
LION	2	1	4	11	2	40	17	2	4	28	18	3	37
LION9	2	1	9	25	4	150	55	2	5	89	67	3	109
MODULO12	1	1	12	24	4	120	40	1	6	72	48	3	96
S8	4	1	5	20	3	140	38	0	4	60	60	2	80
SAND	11	9	32	184	5	1623	675	7	7	1031	891	4	1334
SHIFTREG	1	1	8	16	3	64	32	1	5	48	40	3	64
SSE	7	7	16	56	4	350	129	4	6	317	219	4	330
STYR	9	10	30	166	5	1390	489	6	7	1090	696	4	1017
TAV.KIS	4	4	4	49	2	258	85	6	4	243	85	3	204
TBK.KIS	6	3	32	1569	5	16107	2235	8	7	13707	5006	3	9418
TRAIN11.KIS	2	1	11	25	4	150	52	2	6	83	68	3	94
TRAIN4.KIS	2	1	4	14	2	56	26	2	4	34	24	3	51

A next fault appears in N^S after exhausting the working area of this network with a foregoing fault, and the foregoing fault being unidirectional has to manifest itself in the network working area on the N^S output lines.

In conclusion, we illustrate the overhead required to change STG for PLA^u (Tab. IV).

The Table V is divided into 4 portions. The first portion describes STG. Here n is the number of input variables, m – output variables, l – the number of products and S – the number of states of FSM. The second portion describes PLA on assumption that we use encoding states with the minimal number of state variables. Here, p is the number of the state variables, Σ_1 – the number of literals in PLA, Σ_2 – the number of 1-values among the full codewords. The third portion describes PLA^u. Here, n_d – the additional number of the input variables. Finally, the fourth portion represents the additional output lines for the external FSM. Here, m_d – the number of the additional output variables. We see that $\Sigma_1 + \Sigma_2$ for PLA^u is, as a rule, less than for PLA but namely, this sum represents the complexity of SSC.

6. CONCLUSION

A method of designing a synchronous totally self-checking FSM network has been suggested. The network comprises a plurality of internal FSMs and a plurality of external FSMs. The proposed method is based (1) on the checking of the output codewords of the external FSMs only, and (2) on using a so called unate PLA description of the FSM, which is a standard PLA description where all of "0" values is changed for "don't care".

It has been proven that, in the frame of the discussed fault model, the whole network can be considered checked upon checking the output lines of the external FSMs only.

It has been proposed to use a SOM-based checker [6] as a low-overhead checker of the network.

The obtained benchmark results show that the proposed method allows receiving the promising results from the point of the required overhead.

The obtained results clearly demonstrate that any arbitrary FSM, if decomposed into a FSM network, can be easily checked with a low overhead using the above method.

Based on the proposed approach, the following recommendations for decomposition of an arbitrary FSM into the FSM network can be formulated:

- The number of the external FSMs of the network should be minimized.
- The external FSMs having a small number of different output codewords are preferable. It allows using the SOM-based checker [6] with low overhead.
- The number of additional input variables of any FSM of the network should be minimized.

References

[1] Smith, J. and Metze, G., "The Design of Totally Self-Checking Combinational Circuit", *Proc. Int. Symp. Fault-Tolerant Computing*, Los Angeles, CA, pp. 130 – 134, June, 1977.

[2] Diaz, M., "Design of Totally Self-Checking and Fail Safe Sequential Machines", *Proc. Int. Symp. Fault-Tolerant Computing*, Urbana, IL, pp. 9 – 24, June, 1974.

[3] Smith, J. and Lam, P., "A Theory of Totally Self-Checking Design", *IEEE Trans. on Computers*, **C32**, 831 – 844, September, 1983.

[4] Hellebrand, S., Wunderlich, H. and Hertwig, A. (1998). "Synthesizing Fast, Online-Testable Control Units", *IEEE Design and Test of Computers*, (4), 36 – 41.

[5] Kia, S. M. and Parameswaren, S., "Self-Checking Synchronous Controller Design", *IEE Proc.-Comput. Digit. Tech.*, **146**(1), January, 1999.

[6] Levin, I. and Karpovsky, M., "On-Line Self-Checking of Microprogram Control Unit", *4th IEEE Int. On-Line Testing Workshop*, Capri, Italy, pp. 152 – 156, July, 1998.

[7] Matrosova, A. Yu. and Ostanin, S. A., "Self-Checking Synchronous FSM Network Design", *4th IEEE Int. On-Line Testing Workshop*, Capri, Italy, pp. 162 – 166, July, 1998.

[8] Busaba, F. Y. and Lala, P. K. (1994). "Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Error", *JETTA*, **5**, 19 – 28.

[9] Levin, I. (1987). "Hierarchical Model of the Interaction of Microprogrammed Automata", *Automatic Control and Computer Sciences*, **21**(3), 67 – 73.

[10] Levin, I. (1986). "Decompositional Design of Automata Based on PLA with Memory", *Automatic Control and Computer Sciences*, **20**(2), 61 – 68.

Authors' Biographies

Professor A. Yu. Matrosova has served at the Tomsk State University (Russia) since 1964 till now. Currently she is a Professor of the Department of Computer Science, Applied Mathematics and Cybernetics of the Tomsk State University and a Chairman of the Software Engineering Program. Her research interests include Determinate and Random Testing, Design for Testability, Self-checking Design. She is an author of more than 70 papers in the field of Fault Tolerant Computing and Logical Design.

Dr. Ilya Levin received the Ph.D. degree in Computer Science from Latvian Academy of Science. During 1993 – 1996 years he was the Chairman of the Computer Systems Department of the Holon Center of Technological Education, Israel. Being presently a Senior Lecturer of the Tel Aviv University, he is a supervisor of the Technology Education program. His research interests include Design Automation, Fault-Tolerant Computing, Formal Methods in VLSI design and Technological Education. He is a member of the IEEE and an author of more than 50 papers both in the Design Automation and in the Technology Education fields.

Sergey Ostanin is a Ph.D. student of the Applied Mathematics and Cybernetics Department of the Tomsk State University. His research interests include Design for Testability and Self-checking Design.