A Fault-tolerant Sequential Circuit Design for SAFs and PDFs Soft Errors*

A. Matrosova, S. Ostanin, I. Kirienko, E. Nikolaeva
Tomsk State University (TSU)
Tomsk, Russia
{mau11, sergeiostanin}@yandex.ru, irina.kirienko@sibmail.com, nikolaeve-ea@yandex.ru

Abstract—This paper presents a fault-tolerant synchronous sequential circuit design based on self-checking system with low overhead. The scheme has a self-checking sequential circuit, a not self-testing checker and a normal (unprotected) sequential circuit. It is proved the reliability properties of the suggested scheme both for single stuck-at faults at gate poles and path delay faults transient and intermittent.

Keywords-fault-tolerant scheme; sequential circuit; self-checking circuit; stuck-at fault; path delay fault; soft error

I. INTRODUCTION

High performance integrated circuits have to be protected not only for single stuck-at faults (SAFs) (transient or intermittent) at gate poles but also for delays that arise in a circuit operation. One of the most widespread and useful in practice delay models is a model of a path delay fault (PDF). In this model, it is considered that for small delays in path elements and connections between its elements a delay in propagating a change in a signal value may exceed an admissible level for a circuit as a whole. This leads to incorrect operation of an entire circuit.

One of the approaches to increase reliability of the system is fault tolerance. The most common technique providing the fault-tolerant property is triple modular redundancy (TMR).

Different fault-tolerant systems that are based on self-checking circuits suggested in [1-5].

In this paper we propose the fault-tolerant sequential circuit design based on the architecture suggested in [5]. The scheme consists of a self-checking sequential circuit, a normal (unprotected) sequential circuit, a not self-testing checker and a multiplexor. Such scheme implements the correct behavior of a sequential circuit when any permissible (among SAFs and PDFs) transient or intermittent fault occurs.

II. FAULT-TOLERANT ARCHITECTURE

We assume that each fault is transient or intermittent, and a next fault appears after a previous one has disappeared, and only one module of the fault-tolerant architecture can be faulty. We suggest applying the architecture of fault-tolerant scheme [5] for sequential circuits. The implementation of a fault-tolerant sequential circuit is shown in Fig. 1.

Here ***SCSC*₁** is a self-checking sequential circuit. Assume that all outputs (primary output lines and next state lines) of combinational part (***K*₁**) of sequential circuit are under observation. We may use any technique providing the unidirectional error manifestation for combinational part of sequential circuit, for example [6]. This technique provides the unidirectional manifestation of single stuck-at faults at gate poles of a combinational part of a sequential circuit when using the proper circuit design. Considering a sequential circuit as a whole (Fig. 1) we admit single stuck-at faults on flip-flop poles. These faults also manifest themselves as unidirectional ones. Thus the considered faults manifest themselves as unidirectional one. It means that ***SCSC*₁** is self-checking sequential circuits for single stuck-at faults at gate poles of its combinational part and the same faults at flip-flop poles.

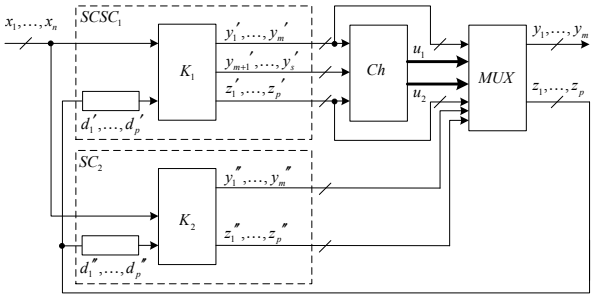

Fig. 1. Fault-tolerant scheme.

***SC*₂** is a sequential circuit realizing Finite State Machine behavior. It has the same encoding states (like ***SCSC*₁**), but has no additional outputs that are used for providing unidirectional error detection.

***K*₂** is a combinational part of sequential circuit ***SC*₂**, implementing the system of partially monotonous Boolean functions (without functions corresponding to additional outputs). The circuit is derived by using any method that provides low cost realization.

Variables $y'_1,\ldots, y'_m$, $(y''_1,\ldots, y''_m)$ correspond to outputs of sequential circuits; variables $y'_{m+1},\ldots y'_{m+s}$ correspond to additional outputs for ***SCSC*₁** providing (k, l)-code words; variables $z_1',\ldots, z_p'$ $(z''_1,\ldots, z''_p)$ are state ones, and $d_1',\ldots, d'_p$ $(d''_1,\ldots, d''_p)$ are flip-flops corresponding to these variables.

Ch is an arbitrary (k, l)-code checker. It may be not self-testing. It is supposed that each next fault appears when a previous one has disappeared. If a checker fault manifests itself, it is masked by correct outputs of ***SC*₂**. This fault does not effect on any next transient or intermittent fault. The checker detects erroneous code words on outputs of ***K*₁**: $y'_1,\ldots, y'_m, y'_{m+1},\ldots y'_{m+s}, z_1',\ldots, z_p'$.

MUX is a multiplexor with control inputs u_1, u_2 and data inputs $y'_1,\ldots, y'_m$, $z_1',\ldots, z_p'$, $y''_1,\ldots y''_m$, $z''_1,\ldots,z''_p$. The ***MUX*** connects lines $y'_1,\ldots,y'_m, z'_1,\ldots, z'_p$ with lines $y_1,\ldots y_m, z_1,\ldots, z_p$ when checker outputs have "10" values otherwise the ***MUX*** connects lines $y''_1,\ldots y''_m, z''_1,\ldots,z''_p$ with lines $y_1,\ldots, y_m, z_1,\ldots,z_p$.

*The reported study was partially supported by Russian Science Foundation, research project № 14-19-00218.

III. FAULT-TOLERANCE ANALYSIS

We consider single stuck-at faults at gate poles of the combinational parts of sequential circuit ***SCSC*₁** and its flip-flops, and single path delay faults of ***SCSC*₁**. As for ***SC*₂** and ***Ch*** their faults may be arbitrary but any fault keeps circuit as combinational one. All above mentioned faults must be transient or intermittent, and a fault occurs one at a time and a next fault from permissible set can appear only after a forgoing fault has disappeared. Only one circuit among ***SCSC*₁**, ***SC*₂**, ***Ch***, ***MUX*** may be faulty.

A. Stuck-at faults

Notice as V_{scsc1} a set of permissible faults of ***SCSC*₁**. V_{scsc1} consists of single stuck-at faults at gate poles and single stuck-at faults at inputs and outputs of flip-flops. All these faults manifest themselves as unidirectional ones on outputs of sub-circuit ***K*₁**. In the presence of any fault from V_{scsc1} circuit ***SCSC*₁** may produce non-code word at the outputs of combinational part ***K*₁** that will be detected by the checker and the multiplexer will use erroneous free outputs from ***K*₂**.

Let V_{Ch} be a set of arbitrary faults of the checker. In presence of any fault from V_{Ch} the checker can produce arbitrary signals at the outputs that drives the multiplexer switching between error free outputs from ***K*₁** or ***K*₂**.

Let V_{SC2} be a set of arbitrary faults of circuit ***SC*₂**. It is supposed only one module of the system may be faulty. This means that other modules are fault-free and the multiplexor uses error free outputs from ***K*₁**.

Let V_{MUX} be a set of permissible faults of the multiplexer. These faults can change connection of some lines $y'_1,\ldots,y'_m$, $z'_1,\ldots, z'_p$ for corresponding lines $y''_1,\ldots y''_m$, $z''_1,\ldots,z''_p$. In this case ***K*₁** and ***K*₂** are fault free and both have error free outputs.

Faults on primary inputs ($x_1, x_2,\ldots,x_n$), primary outputs $(y_1, y_2, ..., y_m)$ and lines $z_1,z_2,\ldots,z_p$ are not considered. Note $V = V_{\text{SCSC1}} \cup V_{\text{Ch}} \cup V_{\text{SCSC2}} \cup V_{\text{MUX}}$.

Proposal 1. The scheme of Fig. 1 keeps correct functioning in the presence of any fault from V.

B. Path delay faults

Consider a combinational circuit in which at time moment t vector v_1 of values of input variables of a circuit is replaced by another vector v_2. Let τ be a maximal admissible path delay in the circuit. If in time period τ after the time moment t, the expected value of vector v_2 on the circuit output does not appear, we say that the circuit has path delay faults for some paths. We say that a pair (v_1, v_2) detects such fault, and the fault manifests itself on this pair.

We call a pair that detects a delay of a signal changing from 0 to 1 on a circuit output as a test for a rising transition; a delay of a signal changing from 0 to 1on the circuit output as a test for a falling transition.

They distinguish single and multiple path delay faults, meaning faults of one or several paths, but we consider only single PDFs.

If a test pair detects the path delay fault regardless of delay faults of other paths of a circuit, we call this pair a robust test for this fault and the fault itself is called robust testable. If a test pair detects the path delay fault of the considered path only on the assumption that other paths of a circuit are fault-free, this pair is called a non-robust test for the fault, and the fault itself is called non robust testable.

Consider test pattern v_2 from a test pair of any PDF. Note that a PDF manifests itself on this test pattern only if previous vector v_1 differs from v_2 by a value of the variable marking the beginning of this path and possibly values of other variables. Otherwise this PDF does not manifest itself on the circuit output.

Masking PDFs we don't need to know what path is fault. That is why we do not distinguish robust testable and non-robust testable PDFs.

As we consider transient or intermittent PDFs we may observe these faults only during time T, $T \geq \tau$. During time T path delay may manifest itself several times both for rising and falling transitions, but only on the same circuit output.

Let $V_{PDF} = V_{SCSC1}^{PDF} \cup V_{Others}^{PDF}$. Here V_{SCSC1}^{PDF} is a set of single path delay faults in ***SCSC*₁**. These faults manifest themselves changing only one output of the combinational part ***K*₁**, and a non code word appears on inputs of the checker. V_{Others}^{PDF} is a set of path delay faults in checker (***Ch***) and ***K*₂**. If ***Ch*** has a PDF then combinational parts ***K*₁** and ***K*₂** are fault free. If ***K*₂** has a PDF then ***K*₁** is fault free. Both cases provide correct outputs.

Proposal 2. The scheme of Fig. 1 keeps correct functioning in the presence of any fault from V_{PDF}.

IV. CONCLUSION

In this paper we have proposed a fault-tolerant sequential circuit design based on a self-checking system with low overhead. The suggested scheme masks both (transient or intermittent) single stuck-at faults at gate poles and path delay faults. The experimental results show that the proposed technique provides an average saving of 33,79% overhead in comparison with TMR implementation and 16,76% overhead in comparison with previous technique based on two self-checking modules.

REFERENCES

[1] I. Levin, A. Matrosova, and S. Ostanin, "Survivable Self-checking Sequential Circuits," Proc. of the IEEE Int. Symp. DFT'01, San Francisco, USA, October 2001, pp. 395-402.

[2] M. Lubaszewski, B. Courtois, "A reliable fail-safe system," IEEE Tran. On Comp., vol. 47, №2, 1998, pp. 236-241.

[3] A. Matrosova, V. Andreeva, Yu. Sedov, "Survivable discrete circuits design," Proc. of the 8 IEEE Int. On-Line Testing Workshop, Bendor, France, 2002, pp. 13-17.

[4] A. Matrosova, S. Ostanin, I. Kirienko, E. Nikolaeva, "Fault-tolerant High Performance Scheme Design," Proc. of IEEE East-West Design&Test Symp., Batumi, Georgia, 2015, pp. 286-289.

[5] S. Ostanin, I. Kirienko, V. Lavrov, "Fault-Tolerant Combinational Circuit Design," Proc. of IEEE East-West Design&Test Symp., Batumi, Georgia, 2015, pp. 302-305.

[6] A. Matrosova, S.A. Ostanin, "Self-Checking Synchronous FSM network Design," Proc. of the IEEE Int. On-Line Testing Workshop, Capri, Italy, 1998, pp. 162-166.