# Dependable Design for FPGA based on Duplex System and Reconfiguration

Pavel Kubalík, Radek Dobiáš, Hana Kubátová
*Department of Computer Science and Engineering*
*Czech Technical University in Prague*
*Karlovo nam. 13, 121 35 Prague 2*
*e-mail: (xkubalik, dobiasr, kubatova)@fel.cvut.cz*

## Abstract

*A technique for highly reliable digital design in FPGAs is presented. Two FPGAs are used for duplex system design, but better dependability parameters are obtained by combination of totally self checking blocks based on parity predictor. Each FPGA can be reconfigured when a SEU fault is detected. Combinational circuit benchmarks have been considered in all our experiments and computations. All our experimental results are obtained by XILINX FPGA implementation by EDA tools. The dependability model and dependability calculations are presented.*

## 1. Introduction

Systems realized by FPGAs are more and more popular due to several properties and advantages:

- High flexibility in achieving multiple requirements such as cost, performance, turnaround time.
- Possible reconfiguration and later changes of the implemented circuit e.g. only via radio net connections.
- Mission critical applications such as aviation, medicine, space missions or also in railway applications [1].

The FPGAs are based on SRAM memories sensitive to Single Even Upsets (SEUs), therefore simple usage of FPGA circuits in mission critical applications without any method of SEUs detection is impossible.

One change of a bit in the configuration memory by SEUs leads to a change of a circuit function, even drastically. The CED techniques allow a faster detection of soft errors (errors which can be corrected by the reconfiguration) caused by Single Event Upsets (SEU) [2, 3, 4]. SEUs can change the content of the embedded memory or Look-up Tables (LUTs) used in the design. These changes are not detectable by off-line tests, therefore some CED techniques have to be used. The probability of a SEU occurrence in the random access memory (RAM) is described in [5].

The possibilities how to keep proper system functions are based always on some redundancy. Redundancy obviously means great area and/or time overhead. Our proposed structure increases dependability parameters together with ensuring a relatively low area overhead compared with classical methods such as duplication or triplication [6]. The term dependability is used to encapsulate the concepts of reliability, availability, safety, maintainability, performability, and testability. Availability is a function of time, A(t), defined as the probability that a system is operating correctly and is available to perform its function its functions at the instant of time ,t [7]. We use availability computation to compare our modified duplex system with standard duplex system.

Our solution combines on-line testing design methods with the classical duplex design. It assumes the dynamic reconfiguration of the faulty part of the system after on-line fault detection. The most important criterion is the speed of the fault detection and the safety of the whole circuit with respect to the application requirement.

Our previous research shows the relation between the area overhead and the SEUs fault coverage [8]. Due to a need for a small area overhead, the SEUs fault coverage for most circuits is less than 100%. The SEUs fault coverage varies typically from 75% to 95%. Therefore an additional method of fault detection has to be used to ensure complete SEUs fault coverage and to increase dependability parameters. The experiments about the scalability of the proposed method, the results closed to the specific design method and the dependability computations are presented. Combinational circuit benchmarks have been considered in all our experiments and computations. All of our experimental results are obtained by XILINX FPGA implementation by EDA tools. The dependability model and dependability calculations based on Markov chains are presented.

The paper is organized as follows: first, basic terms concerning the classification of the faults are presented in Section 2. The proposed structure to be implemented in FPGAs is described in Section 3. The dependability models and computations are presented in Section 4. Section 5 summarizes and expresses the results obtained from these models by several graphs and Section 6 concludes the paper.

## 2. Basic On-Line Testing Criteria

There are three basic quantitative criteria in a field of CED: fault security (FS), self-testing (ST) and totally self-checking (TSC) properties [7]. These three aspects have to be used in an on-line testing field to evaluate the level of safety of the designed or modeled system.

To determine whether the circuit satisfies the TSC property, detectable faults belonging to one of four classes A, B, C and D [9] have to be calculated.

- Class A - hidden faults. These are faults that do not affect the circuit output for any allowed input vector. Faults belonging to this class have no impact to the FS property, but if this fault can occur, a circuit cannot be ST.
- Class B - faults detectable by at least one input vector and they do not produce an incorrect codeword (valid code word, but incorrect one) for other input vectors. These faults have no negative impact to the FS and ST property.
- Class C - faults that cause an incorrect codeword for at least one input vector and they are not detectable by any other input vector. Faults from this class cause undetectable errors. If any fault in the circuit belongs to this class, the circuit is neither FS, nor ST.
- Class D - faults that cause an undetectable error for at least one vector and a detectable error for at least one another vector. Although these faults are detectable, they do not satisfy the FS property and so they are also undesirable.

This fault classification can be used to calculate how much the circuit satisfies the FS or ST property and then calculate TSC properties.

Parity predictor is used to generate the proper output code of the circuit in our research, Figure 1. These techniques ensure a small area overhead and a higher SEUs fault coverage but the SEUs fault coverage reached is not 100% [10, 11, 12].

The circuit area overhead significantly depends on parity codes used. If we use a strong error detecting code, like a Hamming code or Berger code, the FS parameter is almost 100% but the area overhead is high [8, 13].

The following structures are vulnerable to SEUs: mux select lines, programmable interconnect point states, buffer enables, LUT values, and control bit values.

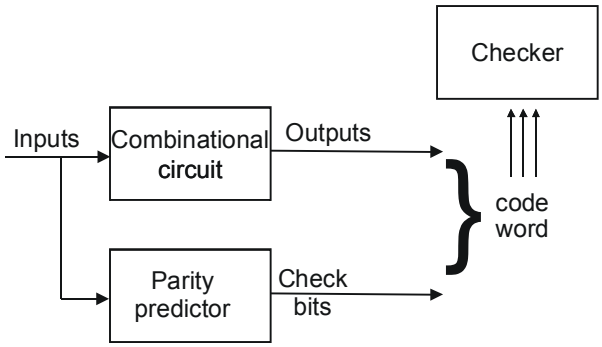


**Figure 1. Basic structure of TSC circuit**

Any changes of a mux select lines, programmable interconnect point states or buffers lead to a significant circuit function change but the function change is hardly detected for SEUs impacted in LUTs [14]. The probability of SEUs impacting routing resources (mux select lines, programmable interconnect point states and buffers) is about 78% and only about 15-21% for LUTs. It means many SEUs leads to significant circuit function change. But any change in LUTs is hardly detected because of their small impact on the realized function. In some cases these faults may be undetected.

We have used the LUT upset failure in our calculations. The only LUT upset assumption giving to us the worst case for availability values obtained for our benchmarks. It means the final results are worst in comparison with method assuming all fault in FPGA. The most faults belonging to routing resources group. In a case we using fault occurring in routing resources, the dependability parameters are higher then for case where we calculate only LUTs.

We want obtain worst case of dependability parameters and due to this fact our fault model accepts only changes in LUTs memory. The FS property depends on the class B. Low number of fault belonging to class B leads to low FS property. The FS values for MCNC and ISCAS [15] benchmarks used to validate our modified duplex system are shown in Table 1. Here "C" is benchmark circuit, "IN" is number of inputs, "OUT" is number of outputs, "AO" is the area overhead, "FS" is a probability, that a fault is detected by code word and "Ass" is the steady-state availability.

We have used our simulator described in [16] to obtain FS property. This simulator has these features:

- The simulation is performed for circuits described by a netlist format (EDIF).
- The stuck-at-1 and stuck-at-0 faults on inputs and outputs of components are considered.
- Combinational and sequential circuits are supported.
- This simulator supports circuits where inputs, outputs and internal states (in the case of a sequential circuit) are coded by even parity,

multiple parity and 1 out of N code. Multiple code groups can be used to ensure TSC. The simulator also supports Hamming like codes and M out of N code.

**Table 1. Single even parity – PLA**

| C | IN | OUT | ORIG [LUT] | AO [%] | FS [%] |
|---|---|---|---|---|---|
| alu1 | 12 | 8 | 8 | 688 | 100 |
| apla | 10 | 12 | 45 | 53 | 83 |
| b11 | 8 | 31 | 38 | 8 | 75 |
| br1 | 12 | 8 | 50 | 20 | 63 |
| al2 | 16 | 47 | 52 | 12 | 94 |
| alu2 | 10 | 8 | 30 | 140 | 92 |
| alu3 | 10 | 8 | 28 | 121 | 90 |
| s1488 | 14 | 25 | 312 | 13 | 86 |
| s1494 | 14 | 25 | 317 | 13 | 86 |
| s2081 | 18 | 9 | 24 | 125 | 96 |
| s27 | 7 | 4 | 4 | 75 | 72 |
| s298 | 17 | 20 | 39 | 49 | 91 |
| s386 | 13 | 13 | 51 | 39 | 71 |

The FS property expresses the probability that an existing fault is detected on a primary output of the circuit. If the FS is fully satisfied (to 100%) a fault occurring in a circuit is always detected.

## 3. Proposed Structure

Our previous results show that to fully satisfy TSC property (100%) is difficult, so we have proposed a new structure based on two FPGAs, see Figure 2.

Each FPGA has one primary input, one primary output and two pairs of checking signals OK/FAIL. The probability of the information correctness depends on the FS property. When the FS property is satisfied only to 75%, the correctness of the checking information is also 75%. It means that the signal "OK" give a correct information for 75% of occurred errors (the same probabilities for both signals "OK" and "FAIL").

To increase the dependability parameters we must add two comparators, one for each FPGA. The comparator compares outputs of both FPGAs. The fail signal is generated when the output values are different. This information is not sufficient to determine, which TSC circuit is wrong. Additional information to mark out the wrong circuit is generated by the original TSC circuit. The probability of the information correctness depends on the FS property and in many cases it is higher than 75%. In a case when outputs are different and one of the TSC circuits signalizes fail function, the wrong FPGA is correctly recognized. Correct outputs are processed by the next circuit.

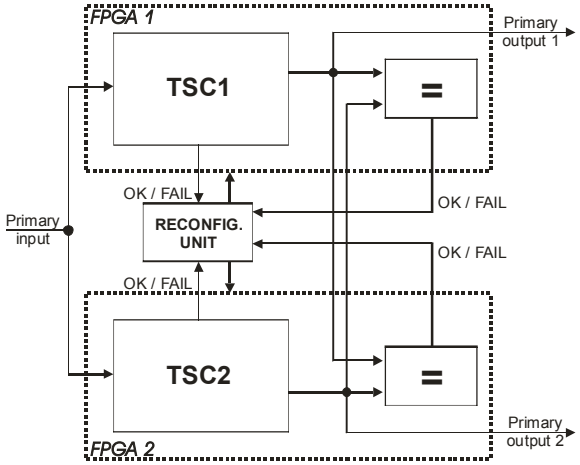


**Figure 2. Reconfigurable duplex system**

The reconfiguration process is initiated after a fault is detected. The reconfiguration solves two problems: localization and correction of the faulty part. The time needed to localize the faulty part is not negligible and must be included in the calculation of dependability parameters. We only select the faulty FPGA and we reconfigure it in our solution. It means that we do not localize the faulty block inside the compound design. The time to localize a fault and to reconfigure the faulty part can be similar to the time to reconfigure the whole FPGA. The whole FPGA reconfiguration also repairs the faults which occurred in an unused logic. The reconfiguration process can be initiated also when one of the two FPGAs signalize the "FAIL" signal. This situation occurs when a fault is detected by one of the small TSC blocks inside the compound design. The fault propagation to the primary outputs may take a long time.

When the outputs are different, and both circuits signalize a correct function, we must stop the circuit function and the reconfiguration process is initiated for both FPGA circuits. After the reconfiguration process is performed, states of both FPGAs are synchronized. It means that our modified duplex system can be used in an application where the system reset synchronization is possible.

Each FPGA contains a TSC circuit and a comparator. The TSC circuit is composed of small blocks where each block satisfies the TSC property. The structure of the compound design satisfying the TSC property is shown in Figure 3.

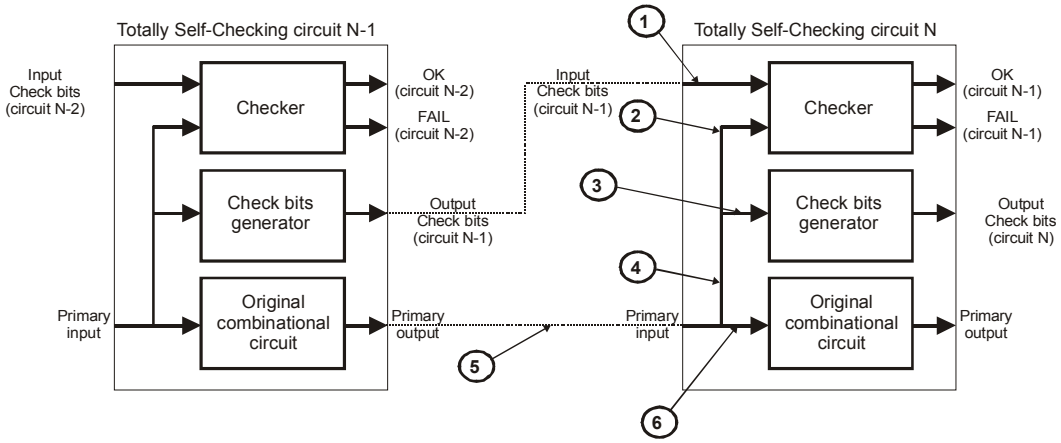


**Figure 3. Proposed structure of TSC circuits implemented in FPGA**

We can assume six places where an error can be observable for this compound design. We assume, for simplicity, that an error occurring in the check bit generator will be observable at the parity nets (number 1) and an error occurring in the original circuit will be observable at the primary outputs (number 5).

The checker in the block N will detect the error if it occurs in net number 1, 2, 4 or 5. If an error occurs in the net number 3 or 6, it will be detected in the next checker (N+1). The method used to satisfy the TSC property for the compound design is described in [17] in more detail.

Every small block (in compound design) does not satisfy TSC property to 100%. The TSC property depends on FS and ST properties which are also not satisfied to 100%. For availability computations, we find the block with the lowest FS property value in the compound design.

## 4. Dependability Analysis

To evaluate the influence of a sequence of SEU faults, a more precise definition of "single fault" is needed. We use availability computation for dependability analysis. In the following text we will assume that a "single data damaging" is defined as follows:

- It will occur at a single time that is arbitrarily located at the time axis.
- The fault can change a data item located within the FPGA configuration memory. Both FPGAs can be affected with the same probability. We assume the single fault changes only one bit of the FPGA configuration memory. Each bit in the FPGA configuration memory can be attacked with the same probability.
- The time between any two single faults is so long, so that a single fault will be successfully detected and corrected. In otherwise it is a multiple fault.

Some basic rules are defined to calculate the availability parameters. We assume that:

- There is at least one input vector coming between two SEUs, which make an output differ from the normal operation.
- SEUs impacting an unused logic do not change the function of the used part. These faults are hidden faults.
- The comparator and the checker are fully TSC.
- The area overhead of the comparator and the checker is negligible.
- The reconfiguration unit loads correct configuration data after a fault is detected.
- The time needed to reconfigure the faulty part depends on the configuration data size.
- The fault that occurred in unused logic does not cause the damaging of the whole FPGA.

The Markov model shown in Figure 4 describes our architecture.

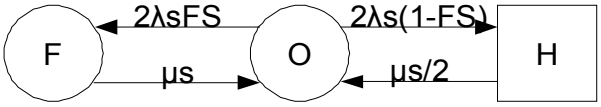


**Figure 4. Model of our modified duplex system**

There are three states (**O**, **F**, **H**).

The **O** state (operational) represents the regular fault-free state of the system, where both FPGAs operate correctly. It means that the fail function is signalized neither by TSC circuit, nor by comparator.

There is transition from **O** to **F** state (one FPGA is faulty) corresponding to the situation when a fault occurs in one FPGA and this fault is detected by one of TSC circuits. The system enters this state with a probability FS. $\lambda$ is the failure rate for one bit of a configuration memory and $s$ is the size of a configuration memory. Number 2 (in the 2λsFS expression) means that one of two FPGAs can be affected with SEUs. The reconfiguration process is initiated only for the faulty FPGA. The repair rate is represented by $\mu$. The second FPGA is running correctly and performs the function of the system.

Some faults are not detected, when the output vector is an incorrect codeword. The probability that an occurred fault causes incorrect codeword is equal to **1-*FS***. In this case the system comes to the state **H**.

The **H** state (hazard) means that the system is in the hazard state. The hazard state is detected (e.g., by comparators), because the output vectors are not identical. Both FPGAs have to be reconfigured in this case. The repair rate is equal to $\mu/2$, because we are reconfiguring each FPGA separately. If we are able to reconfigure both FPGAs at the same time, the availability parameters will increase.

$$\begin{aligned} 2s\lambda p_O - \mu s p_F - \frac{\mu s p_H}{2} &= 0 \\ \mu s p_F - 2s\lambda FS\, p_O &= 0 \\ \frac{\mu s p_H}{2} - 2s\lambda(1-FS)p_O &= 0 \\ p_O + p_F + p_H &= 1 \end{aligned} \quad (1)$$

The described model introduces four parameters: the failure rate ($\lambda$), the repair rate ($\mu$), the fault security (***FS***) and the configuration memory size ($s$). These parameters are discussed in the next section. Now let us transform the Markov model into a system of equations describing the steady state probabilities of each of the states (Equations 1). The system of equations is completed with a normalisation condition.

$$A_{SS} = p_O + p_F \quad (2)$$

The value of the steady-state availability $A_{SS}$ is a sum of probabilities for all working states (Equation 2).

## 5. Results

Firstly we discuss the model parameters. The failure rate ($\lambda$) depends on the probability that the impacting SEUs will change a bit in the FPGA configuration memory. Due to this fact we took into account the result presented in [5] and set the failure rate to:

$$\lambda = 1.8e^{-5}\ [h^{-1}]$$

The repair rate ($\mu$) depends on the time needed for the reconfiguration of an FPGA. The clock frequency was set to 25 MHz. The configuration memory size $s$ (needed for each benchmark) was calculated as a product of the configuration memory size for AT94K40 ATMEL FPSLIC and the circuit area overhead (AO[%]).

$$s = 233k \cdot AO[bits] \quad (3)$$

The graphs (Figure 5, 6, 7, 8) were constructed by solution of the equations (1, 2). We have used the equations (1) and (2) for the following calculations. Firstly the circuit area overhead was fixed to 50 percent. The FS parameter varies from 0 to 100% FS. The availability parameter increases with higher FS. See figure 5.

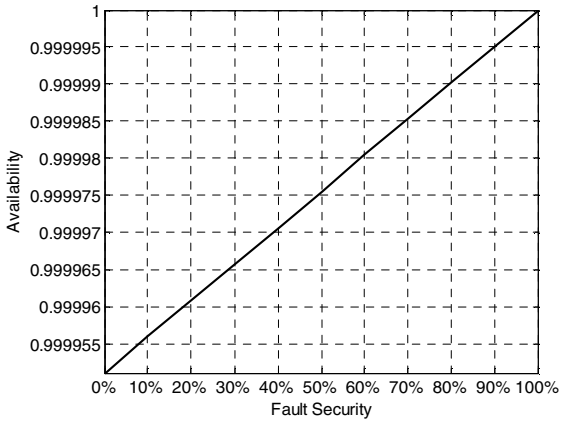


**Figure 5. Availability for 50% overhead**

The curve in Figure 5 is generally described by the following equation 4.

$$A_{SS} = \frac{2FS\lambda + \mu}{4\lambda - 2FS\lambda + \mu} \quad (4)$$

In the second case the FS is 80% and area overhead varies from 0 to 100%.

Figure 6 shows that higher area overhead means a low availability parameter but the availability parameter is decreasing slower than in our first case, when the value of FS parameter is changing.

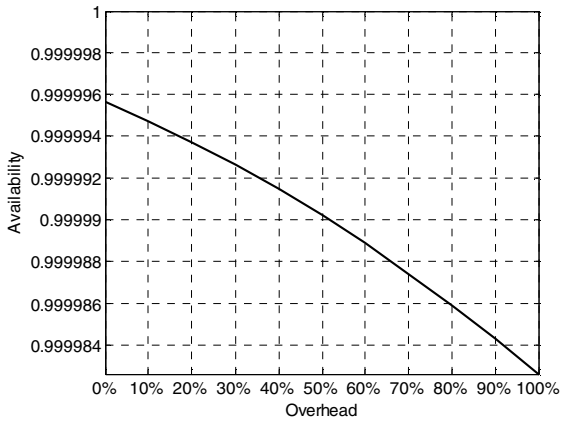


**Figure 6. Availability for 80% FS**

In the third case we show the relation between the area overhead, the FS property and the availability. The results are shown in Figure 7. One point (number 1 in Figure 7) corresponds to the standard duplex system. The availability of standard duplex system is 0,999978248.

When both the area overhead and FS are 0% (the front corner in Figure 7), the availability of our system would be the same as for the standard duplex system without any detection of a faulty FPGA.

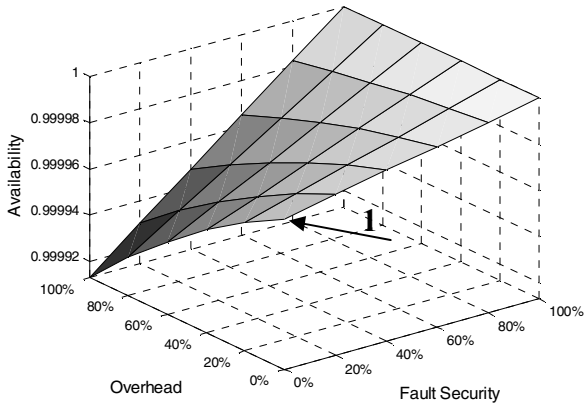


**Figure 7. Availability 3D graph**

The graph in Figure 8 describes the dependency of AO on FS parameterized by the availability. One curve (number 1 in Figure 8) corresponds to the standard duplex system. Due to this, when FS is 50%, the area overhead must be less than 40%. In other cases the system is worse than standard duplex system with respect to availability.

The arrow 2 in Figure 8 shows the area where the system is worse than a standard duplex system with respect to availability. And the arrow 3 shows where the system is better standard duplex system with respect to availability. Each curve in Figure 8 represents one value of the availability parameter.

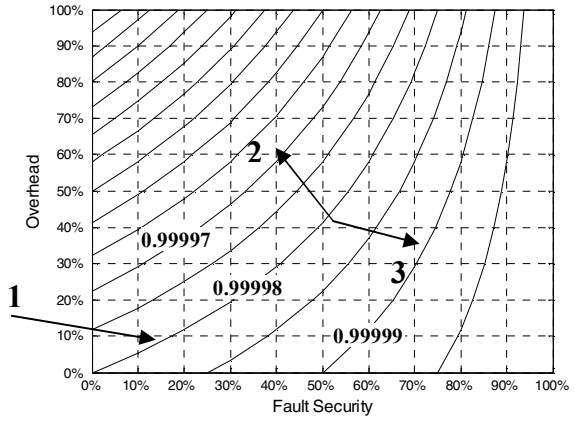


**Figure 8. Curves of availability values**

The results obtained by our case study were validated on MCNC and ISCAS benchmarks. Our results are shown in Table 2. The fault security (***FS***) and the area overhead (***AO***) are summarized in Table 2, where the results obtained by the computation of the models is also included.

Here "CIRCUIT" is benchmark circuit, "AO" is the area overhead, "FS" is a probability, that a fault is detected by code word and "Ass" is the steady-state availability.

**Table 2. Availability parameters**

| CIRCUIT | AO [%] | FS [%] | ASS [%] |
|---|---|---|---|
| alu11 | 687,5 | 100 | 1 |
| apla | 53,3 | 82,8 | 0.9999912 |
| b11 | 7,9 | 75,5 | 0.9999938 |
| br1 | 20,0 | 62,9 | 0.9999847 |
| al2 | 11,5 | 94,3 | 0.9999985 |
| alu2 | 140,0 | 92,5 | 0.9999906 |
| alu3 | 121,4 | 90,3 | 0.9999897 |
| s1488 | 13,1 | 86,3 | 0.9999962 |
| s1494 | 12,9 | 86,3 | 0.9999962 |
| s2081 | 125,0 | 96,2 | 0.9999958 |
| s27 | 75,0 | 72,2 | 0.9999815 |
| s298 | 48,7 | 91,0 | 0.9999957 |
| s386 | 39,2 | 71,1 | 0.9999878 |

The availability of original duplex system is 0,999978248. If we compare original duplex system with our modified duplex system we increase availability parameter for all tested benchmarks. The availability parameter is same as for triplex system in a case when FS property is 100%.

## 6. Conclusion and future work

Our modified duplex system based on two FPGAs has been presented. Our system increase dependability parameters for standard duplex system. Dependability parameters have been increased due to reconfiguration process and two methods of SEUs detection. The first method compares primary output of each FPGA and the second one signalizes faulty FPGA. We described the system by dependability Markov model. This model was used for computation of availability parameters with respect to SEU fault model. The results on MCNC and ISCAS benchmarks have been compared with those of the standard duplex system. We found out that availability depends more on the FS property than on the area overhead. When the FS is not 100%, the area overhead is strictly limited by the availability value of the standard duplex system. When this value is surpassed, the availability is inferior to the standard duplex system. We can summarize that for the tested benchmarks, the availability parameters have increased. E.g., "apla" with 82.8 % of FS and 53 % of area overhead, the time when the system is unavailable is about 2.5 times shorter than for the standard duplex system.

Our future work will be dedicated to several practical case studies (e.g., railway applications). The dependability parameters will be calculated more precisely using assumptions about routing resources impacted by SEUs. We will use a hardware fault simulator based on the ATMEL FPSLIC circuit.

## 7. Acknowledgement

This research has been supported in part by the GA102/04/2137 grant and MSM6840770014 research program.

## 8. References

[1] Dobiáš, R., Kubátova, H.:"FPGA Based Design of Raiway's Interlocking Equipment", In Proceedings of EUROMICRO Symposium on Digital System Design. Piscataway: IEEE, 2004, pp 467-473.

[2] Sterpone, L., Violante, M.: "A design flow for protecting FPGA-based systems against single event upsets ", DFT2005, 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 436 – 444

[3] QuickLogic Corporation.: Single Event Upsets in FPGAs, 2003, www.quicklogic.com

[4] Bellato, M., Bernardi, P., Bortalato, D., Candelaro, A., Ceschia, M., Paccagnella, A., Rebaudego, M., Sonza Reorda, M., Violante, M., Zambolin, P.: "Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA." Design Automation Event for Electronic System in Europe 2004, pp. 584-589.

[5] Normand, E.: "Single Event Upset at Ground Level," IEEE Transactions on Nuclear Science, vol. 43, 1996, pp. 2742-2750.

[6] Dobiáš, R., Kubalík, P., Kubátová, H.: "Dependability Computations for Fault-Tolerant System Based on FPGA", In Proceedings of the 12th International Conferrence on Electronics, Circuits and Systems, IEEE Circuits and Systems Society, 2005, vol. 1, s. 377-380.

[7] Pradhan, D. K., Fault-Tolerant Computer System Design, Prentice-Hall, Inc., New Jersey, 1996.

[8] Kubalík, P., Fiser, P., Kubátová, H.: "Minimization of the Hamming Code Generator in Self Checking Circuits", Proceedings of the International Workshop on Discrete-Event System Design - DESDes'04. Zielona Gora: University of Zielona Gora, 2004, s. 161-166.

[9] Kafka L., Kubalík P., Kubátová H., Novák O.: "Fault Classification for Self-checking Circuits Implemented in FPGA", Proceedings of IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop. Sopron University of Western Hungary, 2005, s. 228-231.

[10] Drineas, P., Makris, Y.: "Concurrent Fault Detection in Random Combinational Logic", Proceedings of the IEEE International Symposium on Quality Electronic Design (ISQED), 2003, pp. 425-430.

[11] Mitra, S., McCluskey E. J.: "Which Concurrent Error Detection Scheme To Choose?" Proc. International Test Conf. 2000, pp. 985-994.

[12] Mohanram, K., Sogomonyan, E. S., Gössel, M., Touba, N. A.: "Synthesis of Low-Cost Parity-Based Partially Self-Cheking Circuits", Proceeding of the 9th IEEE International On-Line Testing Symposium 2003, pp. 35.

[13] Bolchini, C., Salice, F., and Sciuto, D.:"Designing Self-Checking FPGAs through Error Detection Codes", 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02), pp. 60, Canada.

[14] Graham, P., Caffrey, M., Zimmerman, J., Sundararajan, P., Johnson, E., Patterson, C.:"Consequences and Categories of SRAM FPGA Configuration SEUs", Military and Aerospace Programmable Logic Devices International Conference, Washington DC, MAPLD 2003 Paper C6.

[15] Brglez, F., Bryan, D., Kozminski, D.: "Combinational Profiles of Sequential Benchmark Circuits", Proc. of International Symposium of Circuits and Systems, pp. 1929-1934, 1989.

[16] Kafka, L.: Design of TSC circuits implemented in FPGA, CTU FEE, 2004.

[17] Kubalik, P., Kubatova, H.: "High Reliable FPGA Based System Design Methodology." Work in Progress Session of 30th EUROMICRO and DSD 2004, Universitat Linz 2004 pp. 30-31.