Fault Tolerant System Design Method Based on Self-Checking Circuits

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

Abstract

This paper describes a highly reliable digital circuit design method based on totally self checking blocks implemented in FPGAs. The bases of the self checking blocks are parity predictors. The parity predictor design method based on multiple parity groups is proposed. Proper parity groups are chosen in order to obtain minimal area overhead and to decrease the number of undetectable faults.

1. Introduction

This paper presents a parity predictor design method based on parity nets grouping. FPGAs are based on SRAM memories sensitive to Single Event Upsets (SEUs), therefore using FPGA circuits in mission critical applications without any method of SEUs detection is impossible. Our structure increases dependability parameters together with ensuring a relatively low area overhead compared with classical methods such as duplication or triplication [1]. Our solution assumes a possible dynamic reconfiguration of a faulty part of the system.

2. Proposed Method

There are three basic qualitative criteria in a field of CED: fault security (FS), self-testing (ST) and totally self-checking (TSC) properties. Our previous results [2] show that to fully satisfy the TSC property (to 100%) is difficult, so we have proposed a new structure based on two FPGAs. Each FPGA has one primary input, one primary output and two pairs of checking signals OK/FAIL.

The parity predictor is used to generate proper output code of the circuit. These techniques ensure small area overhead and higher fault coverage while the fault coverage reached is not 100% [3, 4, 5].

2.1. Parity Bits Grouping

An algorithm used for grouping the circuit's outputs is described here. Two outputs are XORed in each step, until a desired number of parity bits is obtained. The selection of outputs to be joined is of a key importance for the final design area overhead. We propose a method based on a "similarity" of functions. The algorithm is based on these assumptions:

(1) When two equal functions are XORed, the result will be '0' for all minterms. If values of two functions will differ in a few minterms, there will be only few '1' values in the resulting function. Experiments show that a low number of '1's at the output is very advantageous for the subsequent minimization process (Figure 1).

(2) Two inverse functions, when XORed, yield a '1' value for each minterm. If the output values of two functions are inverse but a few minterms, there will be only few '0' values in the result, which is advantageous too (Figure 1).

(3) If two functions are "similar", there is a big probability that they will share a lot of logics.

A typical dependency of an area on the number of '1' values in the output is shown in Figure 1. The number of '1's in the output varied from 10% to 90% while the number of gate equivalents of the circuit obtained after a minimization by BOOM [6, 7] was measured.

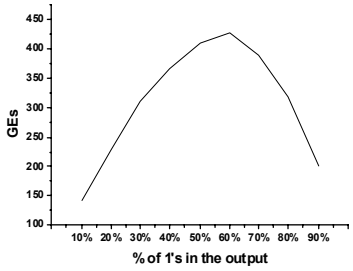

Figure 1: Dependency of the area overhead on the ratio of output '1's

2.2. Evaluation of Similarity of Functions

The first two criteria could be a sufficient criterion to choose the two outputs to be joined. Experiments show that this is not an efficient way to do so (see 3.1). A *scoring function* is introduced to obtain better results. Its value describes the measure of a "similarity" of the two functions. The method is based on a comparison of values of two functions, when a value of one input variable is changed. To compute the values of scoring functions, all the minterms are processed. For each minterm each input variable value is changed and values of the outputs of the two functions are observed. If both values remain unchanged, the scoring function is increased by one, since this represents the same behavior of these two functions. If both values change, the scoring function is increased by one as well.

3. Experimental Results

3.1. Comparison of the Methods

The results obtained by our case study were validated on MCNC and ISCAS [8] benchmarks.

Four methods are compared here, to evaluate the area overhead reached by each of them. The newly proposed method, described in Subsection 2.2 ("Prop") is compared with two simple methods where only the minterm values are compared ("Equal", "Inverse"), (1) and (2) in 2.1. Then, values obtained by the proposed method are compared to a random choice of outputs to be grouped together ("Rand"). An average of 500 random grouping is considered in this experiment.

All the values are in terms of gate equivalents, obtained after the synthesis. An area reduction obtained by the proposed method, with respect to the random method is shown in the "Red." column. It can be seen that there is often a significant improvement with respect to the random method. Simple "Equal" and "Inverse" methods do not yield satisfactory results.

Table 1. Comparison results

Circuit	Prop [GEs]	Equal [GEs]	Inverse [GEs]	Rand [GEs]	Red.
alu1	156	1670	1442	967	83.9 %
apla	76	81	136	128	40.6 %
b11	21	20	17	36	41.7 %
alu2	40	1122	547	418	90.4 %
alu3	320	534	573	433	26.1 %
s1488	241	299	289	364	33.8 %

3.2. Evaluation of the Availability Parameters

The results obtained by the computation of the models are summarized in Table 2. Here "Circuit" is the benchmark circuit, "AO" is the area overhead, "C" and "D" is the number of undetected faults, that are not detected by code word , "SP" is number of parity nets and "Ass" is the steady-state availability

Table 2. Availability parameters

Circuit	AO [LUT]	SP	AO	C	D	Ass [%]
alu1	55	1	687%	0	0	1
alu1	16	2	200%	0	0	1
apla	24	1	53%	1	109	0.9999912
apla	22	2	49%	1	92	0.9999928
b11	3	1	8%	42	59	0.9999938
b11	7	2	18%	38	52	0.9999937
alu3	34	1	121%	0	63	0.9999897
alu3	33	2	118%	0	63	0.9999888
s1488	41	1	13%	94	321	0.9999962
s1488	94	2	30%	80	267	0.9999961

4. Conclusions

A fault tolerant system design method based on parity bits grouping is proposed. We design a parity predictor, composed as a duplicate of the original circuit with its outputs joined by "xor" gates. The method is based on an algorithm properly choosing the circuit outputs to be joined. This yields a big area overhead reduction, with respect to other methods. The method has been verified by experiments including dependability models for dependability computations.

Acknowledgement

This research has been supported by GA102/04/0737 and MSM6840770014.

References

[1] Dobiáš, R., Kubalík, P., Kubátová, H.: "Dependability Computations for Fault-Tolerant System Based on FPGA", Proc. of 12th ICECS Conf., 2005, vol. 1, pp. 377-380.

[2] Kafka L., Kubalík P., Kubátová H., Novák, O.: "Fault Classification for Self-checking Circuits Implemented in FPGA", Proc. of IEEE DDECS Workshop. Sopron, 2005, pp. 228-231.

[3] Drineas, P., Makris, Y.: "Concurrent Fault Detection in Random Combinational Logic", Proc. of the IEEE ISQED Symposium, 2003, pp. 425-430.

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

[5] Mohanram, K., Sogomonyan, E. S., Gössel, M., Touba, N. A.: "Synthesis of Low-Cost Parity-Based Partially Self-Checking Circuits", Proc. of the 9th IOLTS 2003, pp. 35.

[6] Hlavička, J, Fišer, P.: „BOOM - a Heuristic Boolean Minimizer". Proc. of ICCAD 2001, San Jose, California (USA), 4.-8.11.2001, pp. 439-442.

[7] Hlavička, J, Fišer, P.: „BOOM - A Heuristic Boolean Minimizer", Computers and Informatics, Vol. 22, 2003, No. 1, pp. 19-51.

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