Miscellaneous Types of Partial Duplication Modifications for Availability Improvements

Jaroslav Borecký, Martin Kohlík and Hana Kubátová
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
Email: {borecjar; kohlimar; kubatova}@fit.cvut.cz

***Abstract*—This paper compares four different redundancy methods, which includes parity code, partial duplication and their combinations, with two standard methods (Duplex and Triple Module Redundancy). Two main attributes are observed: the *Total size* of system including overhead caused by redundancy addition and steady-state availability – dependability parameter defining the readiness for correct service of a system.**

I. Introduction

We have designed railway equipment systems based on Field Programmable Gate Arrays (FPGAs) composed of co-operating modules in our department ([1], [2]) that must meet dependability requirements given by standard ČSN 50129 [3] in accordance with the standard EN 50129:2003 [4].

FPGA-based systems are sensitive to many effects that can change their programmed function. [5] These changes are most unwelcome in systems, where the material loss or mortality can be caused because of their failure. The improvement of dependability parameters of a final design of a system is required to minimize the impact of such effects.

The dependability of a system is the ability to avoid service failures (situation when the delivered service deviates from correct service) that are more frequent and more severe than is acceptable. [6]

Dependability is an integrating concept that includes the following attributes: [6]

- *Availability* – readiness for correct service.
- *Safety* – absence of catastrophic consequences on the user(s) and the environment.
- *Reliability* – continuity of correct service.
- *Integrity* – absence of improper system alterations.
- *Maintainability* – ability to undergo modifications and repairs.

One of the most important design techniques allowing improvement of dependability is redundancy. This means that if one part of the system fails, there is an alternate functional part. However, redundancy can have a negative impact on a system performance, size, weight, power consumption, and others. [7]

There are many redundancy techniques like hardware, information, time, software redundancy etc. [7] We focus on

- *hardware redundancy* – replication of hardware (duplication),
- *information redundancy* – addition of redundant information (parity code)

and/or their combinations in this paper.

We want to compare efficiency of these methods measured by *Total size* (a size of an original circuit and overhead caused by redundancy addition) and *availability* (we use *availability* as steady-state availability in this paper) with standard methods (Duplex and Triple Module Redundancy (TMR)).

We will consider a method to be better than TMR, if *availability* is equal to 1 and the *Total size* is lower than 300%.

We will consider a method to be significantly better than Duplex, if *availability* is at least 10x better than *availability* of Duplex and the *Total size* is lower than 300%.

The paper is organized as follows: Section II introduces Modified Duplex System architecture, fault classification, *availability* calculations and partial duplication method. Section III contains detailed description of presented redundancy methods. The results are provided in Section IV. Section V concludes the paper.

II. Theoretical Background

A. Modified Duplex System

We encase design using tested redundancy method to Modified Duplex System architecture to calculate *availability*.

Modified Duplex System (MDS) architecture uses two instances of design that may be not fault tolerant. The purpose of MDS architecture is to achieve the whole system including all checkers and comparators to be fault tolerant. The block diagram of MDS is shown in Figure 1.

The probability of detecting error inside a design is measured by Fault Security (FS) parameter. FS is dependability parameter expressing the probability that the erroneous outputs produced under a fault belong to the output code[1].

[1]Output code is formed from data word (original information) and parity bit (redundant information).

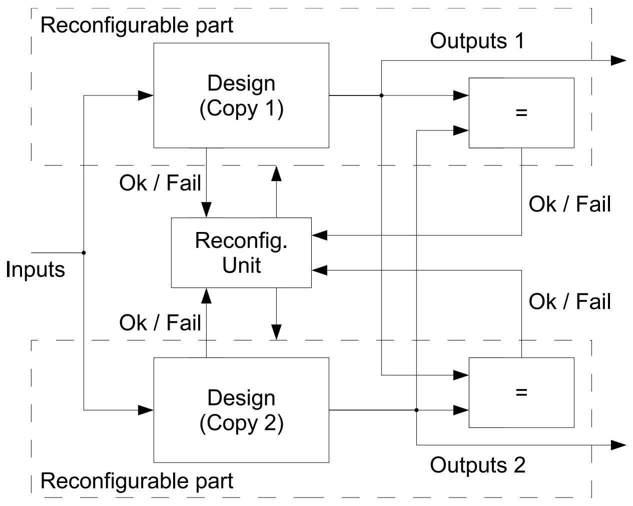

Fig. 1. Block scheme of Modified Duplex System.

Each improvement of FS of a design increases the probability of detecting error inside the design. Detecting an error inside the design initiates reconfiguration of the damaged block, but the second design is operational. If the error is not detected inside the design, it is detected by comparators. The error detected by comparators initiates the reconfiguration of both designs, because outputs from designs are different, but the source of the error cannot be determined.

More details about MDS architecture can be found in [2].

B. Fault Classification

We need to classify faults to calculate Fault Security (FS) parameter that is used during *availability* calculations.

A (logical) fault can result from particular physical incorrectness. The fault can lead to an error, which means a mistake in data (output code). [6]

There are two main criteria to classify a fault. Both of them are concerned about existence of a valid input data word causing that the outputs produced under the fault are erroneous.

- *Test criterion* – Does it exist a valid input data word causing that the output code produced under the fault is erroneous, and it does not meet output code specification (parity bit is invalid)?
- *Error criterion* – Does it exist a valid input data word causing that the output code produced under the fault is erroneous, but it meets output code specification (parity bit is valid)?

The combination of answers to these criterion-questions results to fault classification (see Table I).

More detailed description of fault classification can be found in [8].

FS is given as the ratio of numbers of faults according to their classification

$$FS = \frac{n_A + n_B}{n_A + n_B + n_C + n_D}$$

TABLE I
FAULT CLASSIFICATION

Fault class	Class name	Test crit.	Error crit.
A	Hidden	false	false
B	Detectable	true	false
C	Undetectable	false	true
D	Partially detectable	true	true

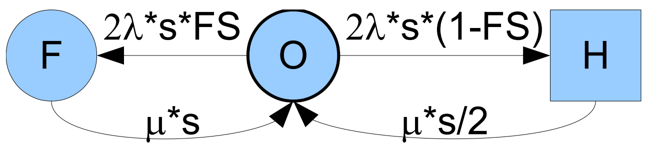

Fig. 2. Markov chain of Modified Duplex System.

C. Availability calculations

Availability is calculated using MDS architecture. *Availability* is the sum of steady-state probabilities of non-hazard states of Markov chain shown in Fig. 2.

The description of the states and the arcs in Fig. 2:

- O – the operational/fault-free state of the system
- F – the system contains a fault that has been detected by output code of one of design copy
- H – the system contains a fault that has been detected by output comparators – the system is not operational until a repair is finished
- λ – the fault rate ($2\,\lambda$ – the fault can affect two copies of design)
- s – the size of the design (one copy including overhead)
- FS – the probability that the fault is detected by output code
- μ – the repair/reconfiguration rate ($\mu/2$ – two copies are reconfigured sequentially)

FS depends on the design and/or selected redundancy method, λ and μ depend on the environment and/or technology, s depends on all of these factors.

More details about MDS architecture *availability* model can be found in [2].

D. Partial Duplication

Partial duplication is the basis of tested redundancy methods. Partial duplication uses two-level redundancy (see the block scheme in Fig. 3):

1) *predictor1* – a parity code generator
2) *predictor2* – a partial copy of the circuit and *predictor1* created on the basis of results of fault simulation

We use Bit-flip fault model in fault simulation. This means, that the fault manifests as a flip of one bit of a memory located in a lookup table (LUT). A LUT is purposed to model Boolean functions as truth tables – it is one of the basic parts of FPGA.

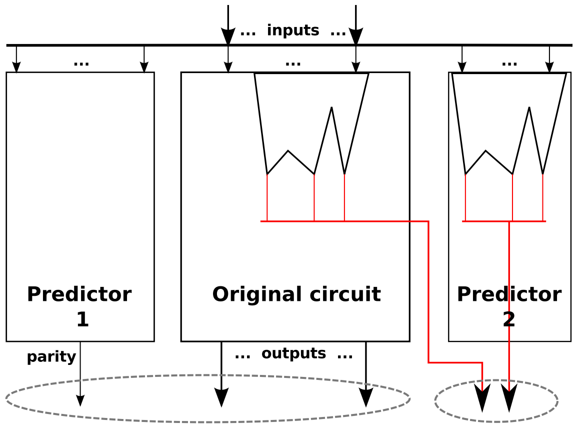

Fig. 3. Scheme of *Partial duplication*.

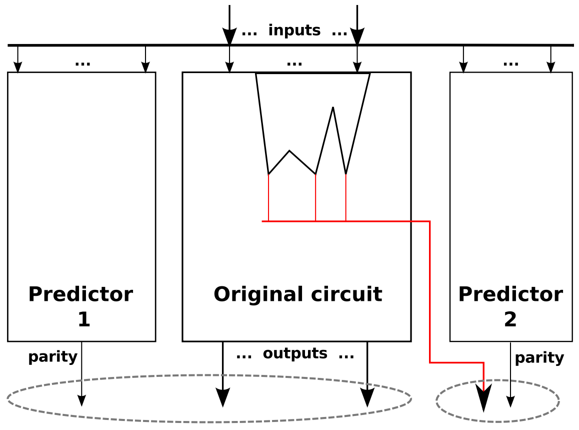

Fig. 4. Scheme of *Parity of partial duplication*.

Second-level redundancy is made as follows:

- join an original circuit and *predictor1*,
- perform fault simulation,
- connect all outputs of LUTs containing faults classified as C or D class to newly-formed test outputs,
- save a copy of the modified circuit as *predictor2*,
- remove all non-test outputs from *predictor2* and all logic that remains unused after non-test outputs removal,
- compare pairs of corresponding test outputs from the modified circuit and *predictor2*.

More details about partial duplication can be found in [9].

III. Description of Redundancy Methods

All four redundancy methods are based on partial duplication described in Section II-D. The description of the methods based on Fig. 3 follows:

- *Simple parity* – Simple parity use *original circuit* and *predictor1*. Partial duplication is not used in this case.
- *Partial duplication* – Partial duplication is exactly the same as described in Section II-D.
- *Parity of partial duplication* – This method is based on *Partial duplication*. This method uses resynthesized *predictor2* that generates parity bit only. This secondary parity bit forms secondary output code together with test outputs of *original circuit*. Block scheme of this method is shown in Fig. 4.
- *Joined parities* – This method is based on *Parity of partial duplication*. *Predictor1* and *predictor2* are resynthesized together to create one parity bit only in this method. Output code is formed by all (original and test) outputs of *original circuit* and one parity bit generated by joined *predictor1/2*. Block scheme of this method is shown in Fig. 5.

The selected redundancy method is encased to Modified Duplex System architecture to calculate *availability*. The selected redundancy method is used as *design*, so it is copied twice as shown in Fig. 1 in Section II-A. This modification allows us to achieve the whole system to be fault tolerant, so we can compare it with Duplex and TMR systems.

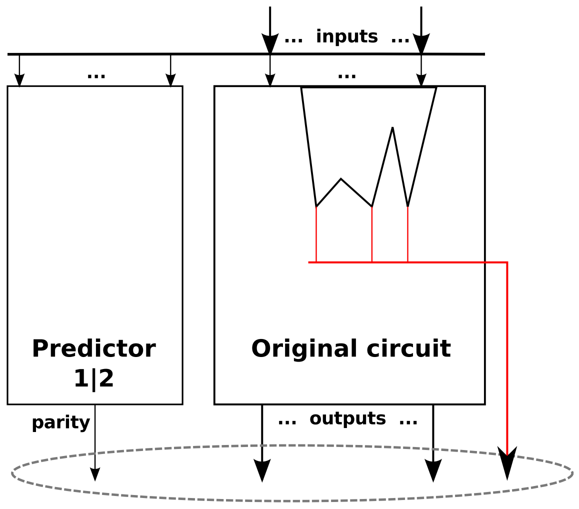

Fig. 5. Scheme of *Joined parities*.

IV. Results

We monitor two parameters: *Total size* and *availability*. *Availability* is calculated using Markov chain shown in Fig. 2 in Section II-C.

The parameters are set as follows:

- FS – Fault Security is the result of fault simulation made during *Partial duplication* described in Section II-D
- *Total size* – see Table III
- $\lambda = 1.8 \times 10^{-5}$ – the fault rate (taken from [2])
- $s = sFPGA * \textit{Total size}/2$ – the size of the one copy of design
  ($sFPGA = 1180800$ – the size of bitstream of Xilinx Virtex4 XC4VLX15[10])
- $\mu = 216 \times 10^9 = f_{rec} * 3600$ – the repair rate
  ($f_{rec} = 60 \times 10^6$ – reconfiguration frequency)

The results of standard redundancy methods using these parameters are shown in Table II.

The results of presented redundancy methods applied on MCNC benchmarks [11] are shown in Table III.

Values on dark grey background represent, that the method applied on that particular benchmark has better results than

TABLE II
FAULT CLASSIFICATION

Method name	Total size (%)	Availability
Duplex	200	0.999535
TMR	300	1.

TMR. It means that *availability* is equal to 1 and the *Total size* is lower than 300%.

Values on light grey background represent, that the method applied on that particular benchmark has significantly better results than Duplex. It means, that *availability* is at least 10x better than *availability* of Duplex (> 0.999953) and the *Total size* is lower than 300%.

The names of methods are abbreviated as follows:

- *SP* – Simple parity
- *PD* – Partial duplication
- *PoPD* – Parity of partial duplication
- *JP* – Joined parities

Two benchmarks (*alu1* and *newbyte*) can be made fully fault-secure by applying *Simple parity*. This means, that *Simple parity* have *availability* equal to 1. Partial duplication is not applicable in this case, because there are no C- or D-class faults present (see Partial duplication algorithm shown in Section III). *Newbyte* benchmark can be made fully fault-secure very effectively, but *Total size* of *alu1* benchmark using *Simple parity* is relatively high.

The results show that *Partial duplication* applied on 28% of benchmarks (5/18) has better results than TMR (equal *availability*, lower size).

Availability of *Parity of partial duplication* applied on 33% of benchmarks (6/18) is equal to 1, but *Total size* is lower than 300% in four cases (benchmarks) only.

Availability of *Joined parities* does not reach 1 in any case, so this method cannot be compared with TMR.

Availability of *Simple parity* is significantly (10x) better than *availability* of Duplex (0.999953) in three cases only, so the method is not significantly better than Duplex.

Two cases of *Partial duplication* cannot be compared with TMR, because *availability* does not reach 1. Both cases have significantly better *availability* than Duplex. One of them has *Total size* over 300%, so it is better to use TMR in this case.

Availability of *Parity of partial duplication* applied on 33% of benchmarks (6/18) is significantly better than *availability* of Duplex, but three of these cases have *availability* equal to 1, so they are even better than TMR.

Availability of *Joined parities* applied on 28% of benchmarks (5/18) is significantly better than *availability* of Duplex. One of these cases has *Total size* over 300%, so it is better to use TMR in this case.

As you can see, presented methods are appropriate for some benchmarks (especially for *newbyte*, *f51m*, *m1*, and *sqr6*), but over half of all benchmarks shows results that are not better then TMR, and/or significantly better than Duplex.

V. CONCLUSION

We presented four different redundancy methods (*Simple parity*, *Partial duplication*, *Parity of partial duplication* and *Joined parities*) in this paper and we compared them with two standard methods (Duplex and Triple Module Redundancy).

Results show that steady-state availability (*availability*) of system using *Partial duplication* method is equal to TMR in most cases (benchmarks), but only 28% of cases have *Total size* lower than the size of TMR.

Availability of 33% of cases of *Parity of partial duplication* is equal to 1, but *Total size* is lower than 300% in three cases only. This method is not better than TMR in most cases.

The main advantage of system using these two methods is that even if *Total size* is over the size of TMR, presented methods will require only two independent blocks instead of three blocks required to build TMR system.

Availability of *Simple parity* is not significantly better than *availability* of Duplex in most cases, so the method is not significantly better than Duplex at all.

Two remaining methods have significantly better *availability* than Duplex in circa 30% of cases. The remaining cases have better *availability* than Duplex, but the difference is not so high to justify the increase of *Total size* of a system.

Presented methods are appropriate for some benchmarks (especially for *newbyte*, *f51m*, *m1*, and *sqr6*), but the results of a half of all tested benchmarks are not better then TMR, and/or significantly better than Duplex.

ACKNOWLEDGMENT

This research has been partially supported by the project SGS12/094/OHK3/1T/18.

REFERENCES

[1] Borecký, J., Kubalík, P. and Kubátová, H.: Reliable Railway Station System based on Regular Structure implemented in FPGA, In Proc. of 12th Euromicro Conf. on Digital System Design, Los Alamitos: IEEE Computer Society (2009), pp. 348–354.

[2] Kubalík, P. and Kubátová, H.: Dependable design technique for system-on-chip, Journal of Systems Architecture, Vol. 2008, no. 54, (2008) 452–464. ISSN 1383-7621.

[3] Czech Technical Standards ČSN EN 50129 Drážní zařízení: Sdělovací a zabezpečovací systémy a systémy zpracování dat: Elektronické zabezpečovací systémy (2003).

[4] European Standards EN 50129:2003 Railway applications: Communication, signalling and processing systems: Safety-related electronic systems for signalling.

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

[6] Avižienis, A., Laprie, J.-C., Randell, B. and Landwehr C.: Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, Jan.–Mar. 2004.

[7] Pradhan, D., K.: Fault-Tolerant Computer System Design, Prentice Hall PTR, Upper Saddle River, New Jersey 1996, ISBN 0-7923-7991-8.

[8] Kafka, L., Kubalík, P., Kubátová, H. and Novák, O.: Fault Classification for Self-checking Circuits Implemented in FPGA, In Proc. of IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop, Sopron University of Western Hungary (2005), pp. 228–231.

[9] Borecký, J., Kohlík, M., Kubátová, H and Kubalík, P.: Faults Coverage Improvement based on Fault Simulation and Partial Duplication, In Proc. of 13th Euromicro Conference on Digital System Design, Los Alamitos: IEEE Computer Society (2010), pp. 380–386.

[10] Xilinx: Virtex 4 documentation http://www.xilinx.com/support/documentation/virtex-4.htm

[11] Yang, S.: Logic Synthesis and Optimization Benchmarks User Guide, Technical Report 1991-IWLS-UG-Saeyang, MCNC, Research Triangle Park, NC, January 1991

TABLE III
TOTAL SIZES AND STEADY-STATE AVAILABILITIES OF PRESENTED METHODS

		Total relative size (%)				Steady-state availability			
Name	Size (LUTs)	SP[1]	PD[2]	PoPD[3]	JP[4]	SP[1]	PD[2]	PoPD[3]	JP[4]
alu1	8	1325.00	N/A	N/A	N/A	1.000000	N/A	N/A	N/A
apla	44	268.10	354.36	427.00	299.88	0.999780	1.000000	0.999883	0.999779
br1	55	239.82	370.14	413.58	308.60	0.999724	1.000000	0.999697	0.999663
br2	30	253.28	373.16	399.80	353.18	0.999648	1.000000	0.999739	0.999682
b12	21	390.40	428.48	418.96	399.92	0.999939	1.000000	1.000000	0.999929
dk17	28	307.10	364.22	385.64	314.24	0.999825	1.000000	0.999905	0.999838
dk27	11	345.44	363.62	363.62	327.26	0.999970	1.000000	1.000000	0.999970
dk48	30	393.14	459.74	486.38	499.70	0.999772	0.999996	0.999862	0.999701
ex1010	1155	241.28	278.08	257.28	215.52	0.999934	1.000000	0.999973	0.999938
f51m	29	268.80	282.56	275.68	268.80	0.999987	1.000000	1.000000	0.999985
gary	197	230.00	276.00	296.00	346.00	0.999917	0.999999	0.999955	0.999872
mp2d	33	309.08	351.50	351.50	327.26	0.999867	1.000000	0.999938	0.999898
m1	24	224.96	258.24	241.60	266.56	0.999963	1.000000	1.000000	0.999959
newapla	16	300.00	387.50	387.50	312.50	0.999802	1.000000	0.999886	0.999793
newbyte	9	222.22	N/A	N/A	N/A	1.000000	N/A	N/A	N/A
newcpla1	39	322.88	425.28	409.92	333.12	0.999727	1.000000	0.999801	0.999773
newcpla2	27	251.80	355.40	318.40	281.40	0.999811	1.000000	0.999886	0.999844
p82	31	219.32	264.40	238.64	212.88	0.999938	1.000000	0.999990	0.999961
sex	20	270.00	310.00	300.00	360.00	0.999926	1.000000	1.000000	0.999910
sqr6	43	227.84	264.96	241.76	227.84	0.999953	1.000000	1.000000	0.999955

[1] Simple parity
[2] Partial duplication
[3] Parity of partial duplication
[4] Joined parities