

AKADÉMIAI KIADÓ

## International Review of Applied Sciences and Engineering

11 (2020) 1, 73-79

DOI:

[10.1556/1848.2020.00011](https://doi.org/10.1556/1848.2020.00011)

© 2020 The Author

## ORIGINAL RESEARCH PAPER



# Modeling and simulation of FPGA-based redundant digital systems

CSABA SZÁSZ\*

Technical University of Cluj, Department of Electrical Machines and Drives, Romania

Received: September 3, 2018 • Accepted: January 4, 2019

Published online: April 18, 2020

## ABSTRACT

Reliability is one of the most important criteria that characterize last generation digital systems. In a wide range of applications the required reliability level is achieved by using hardware redundant configurations. Perhaps their most common form is the triple modular redundancy (TMR) based on a majority voting structure. Researchers that use this strategy make a major assumption: in fault-free operation mode the outputs of these digital systems match in all. This paper proves that synchronization and matching in all the outputs of such systems is not such a trivial problem. In this endeavor FPGA-based (Field Programmable Gate Arrays) redundant topologies are considered for study and experiments. Upon these structures specially conceived redundant models have been developed and simulated. The results outline that synchronization of complex digital systems is a difficult engineering undertaking and any initial assumption should be managed with the adequate circumspection.

## KEYWORDS

redundant system, reliability, digital voter, FPGA processor, outputs matching

## 1. INTRODUCTION

In the related scientific literature massive redundancy is a well known strategy to improve complex digital systems reliability and fault-tolerance behaviors. The vast majority of such solutions usually are developed upon the N modular redundancy (NMR) topology requiring a single voter module that decides the right output signals. Rationality considerations impose that in such situations it is better than the used redundant modules to not exceed more than five units [1]. The triple modular redundancy (TMR) configuration uses three perfectly identical hardware modules that perform the same tasks and programmed functions inside the digital system. A majority voting strategy is performed over the outputs of these three units. The voting process means that failures in any single module are tolerated since single false output information will be outvoted by the two remaining modules. The topic of the NMR redundant systems was sufficiently discussed in the past years and a great number of high quality works have been published in international references. Among these may be found both attempts to improve TMR strategies [2, 3, 4], or researches dedicated to new topologies for robust majority voter structures [5, 6].

However, what is almost common in most of these valuable contributions is that they make two crucial theoretical assumptions. The first assumption is that the redundant modules are perfectly synchronized [7]. The second is linked with the idea that in fault-free operation mode the outputs of the three modules match in all (the signals are exactly the same). Of course, such initial suppositions may be accepted more easily in case of relatively simple digital circuits that compose the TMR structure. Unfortunately, such an approach is difficult to be accepted in case of nowadays configurations when the individual modules of the TMR topology are built upon complex programmed digital systems. In such situations the right operation of the well-known TMR architecture used to increase the general reliability coefficient of digital systems is not so trivial. Moreover, to reach the conditions that fulfill all the above mentioned assumptions becomes a difficult engineering undertaking. This paper is focused on emphasizing the difficulties that may

\*Corresponding author.

E-mail: [Csaba.Szasz@emd.utcluj.ro](mailto:Csaba.Szasz@emd.utcluj.ro)



AKJournals

occur when synchronization and matching is targeted in complex redundant systems. The inherent bottlenecks that cannot be avoided in such situations are carefully evidenced, respectively theoretical and experimental solutions to manage them are searched.

## 2. IDEAL AND REAL TRIPLE MODULAR REDUNDANCY

The well known operation principle of the TMR configuration was introduced first by John von Neumann in the reference [8]. This structure embeds three perfectly identical hardware units operating under the same conditions and executing the same task inside system, respectively a voter unit that captures the output signals generated by the above mentioned three modules. The essence of this redundancy consists of performing a majority voting strategy over the three outputs. According to this, if one of the modules enters into a faulty-state the two remaining faulty-free units perform a majority voting algorithm and the occurred fault is masked inside system. Therefore, in each case are released toward only the signals corresponding to the properly operating modules. This strategy is expressed by a very simple block diagram as shown in Fig. 1.

If the three hardware units generate the  $X_1$ ,  $X_2$ , and  $X_3$  signals the operation mode of the TMR topology is described by the equation

$$Y = X_1 \dot{\wedge} X_2 + X_2 \dot{\wedge} X_3 + X_3 \dot{\wedge} X_1 \quad (1)$$

where  $Y$  means proper the signal released (or passed) to the output. The above configuration also may be expressed via a simple 1 bit digital voter combinational logic circuit as is presented next in Fig. 2.

Let us generalize this circuit by considering the situation when the modules *Hardware\_Unit\_1*, *Hardware\_Unit\_2* and *Hardware\_Unit\_3*, are complex programmable digital systems (based even on microprocessor architectures) and they generate output signals on  $n$  bit-long buses. Figure 3 describes this state for the specific case when the operation of the above mentioned three modules has been perfectly synchronized in time. In fact, this means an ideal TMR configuration where it is assumed that the outputs (the three  $n$  bit buses) match in all.

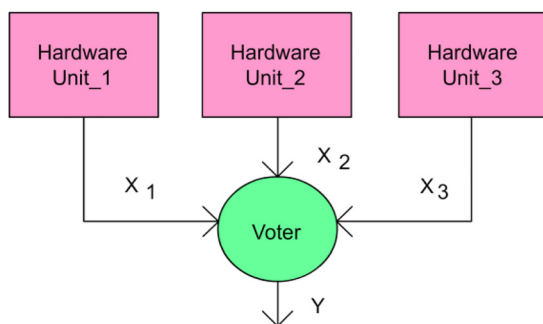


Figure 1. The TMR concept

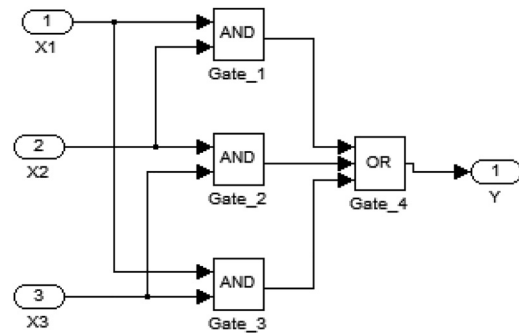


Figure 2. The 1 bit digital voter circuit

This is one of the most frequently made assumptions adopted by a wide range of authors in international references. Of course, the TMR architecture operates properly only in this ideal (or particular) case, in other situations the released signals on the  $Y$  bus will be totally compromised. Such a real case is presented next in Fig. 4, where three physically independent hardware units or modules driven by their own individual clock systems have been considered. If it is assumed that these clock systems are running independently from each other, they will generate asynchronous  $n$  bit signals on their buses. It means that inherent time delays will occur on their buses even they execute the same algorithm or programmed task in perfectly similar operating conditions.

For example, in Fig. 4 it has been evidenced that the concrete situation where the  $n$  bit signals generated by the *Hardware\_Unit\_2* and *Hardware\_Unit\_3*, are delayed with the time intervals  $t_1$  respectively  $t_2$  against the considered initial reference time moment  $t_0$ . Of course, in this case the

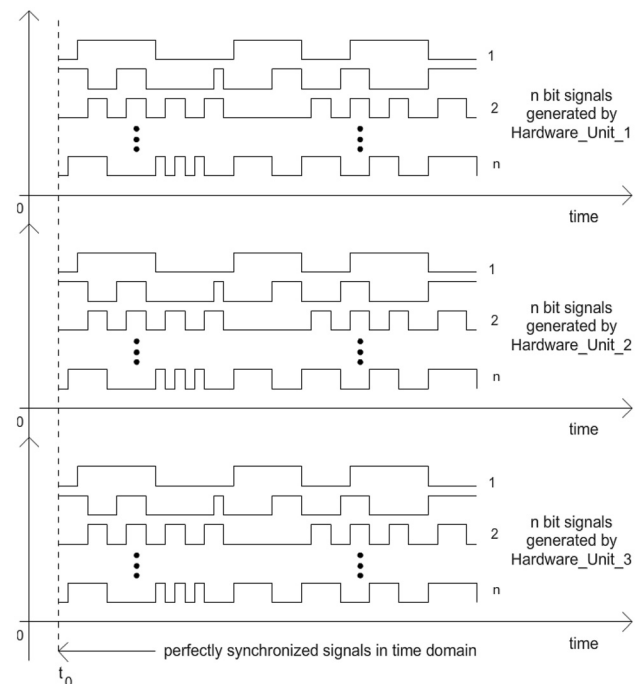


Figure 3. Ideal  $n$  bit TMR configuration operation model

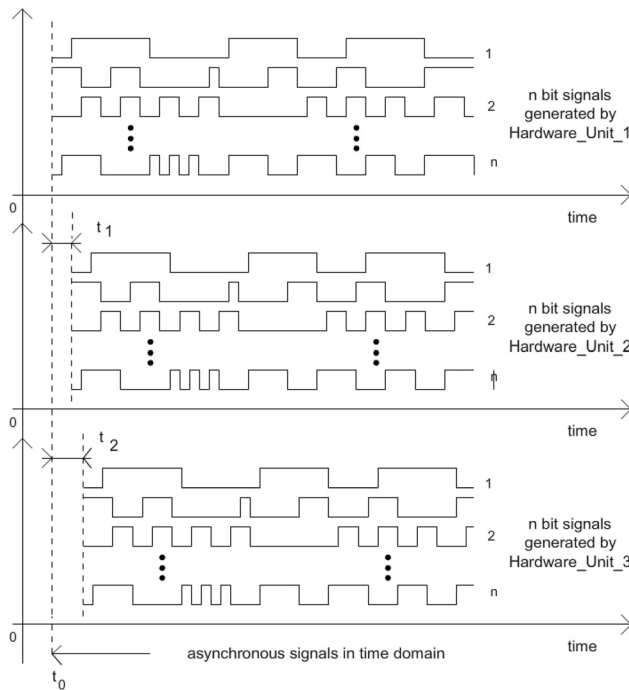


Figure 4. Real  $n$  bit TMR configuration operation mode

TMR concept presented in Fig. 1 is no longer valid and the initial assumptions that the redundant modules are perfectly synchronized and in fault-free operation mode their outputs match in all should be avoided. As a consequence, the next paragraph will be dedicated in all to the research efforts aimed to find out original hardware redundant configurations being able to reach again the desired TMR operation mode expressed above in Fig. 3. In other words, to study the specific situations for which the initial crucial assumptions may be maintained valid.

### 3. MODELING AND EXPERIMENTING FPGA-BASED TMR CONFIGURATIONS

In order to follow a concrete application example let us consider a hardware redundant system composed by three perfectly identical Field Programmable Gate Arrays (FPGA)-based development boards as shown next in Fig. 5.

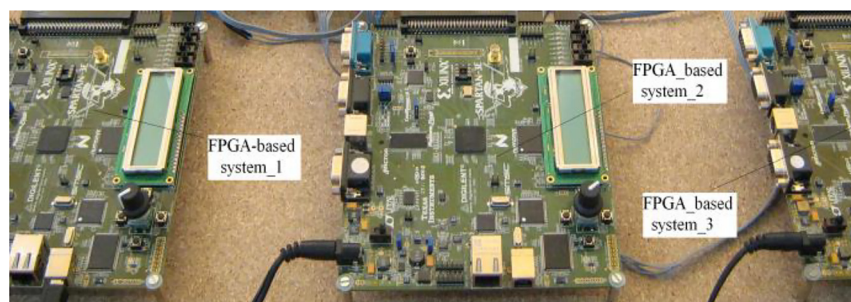


Figure 5. FPGA-based development boards TMR configuration

This is a redundant topology based on *Spartan-3E Starter Kit* ready-to-use manufacturer boards. The main hardware resources and features of such a development board are as follows: a Xilinx XC3S500E Spartan-3E FPGA processor with 232 user I/O lines, Xilinx 4 Mbit Platform Flash, 64Mbyte DDR SDRAM, Flash, 16Mbits of SPI serial Flash, 2×16 character LCD, PS/2 mouse or keyboard port, 50MHz clock oscillator, three Digilent 6-pin connectors, rotary encoder with push-button shaft, 8 LEDs, 4 slide switches, 4 push buttons [9]. The three individual modules have been connected into a TMR topology and are uploaded with the same program task developed in VHDL (Very High speed integrated circuits Description Language) technology. In order to not complicate excessively the experimental and simulation efforts let us admit that they generate information only on a 4 bit-long data bus.

For example, a two  $\pi/2$  delayed pulse trains with a given frequency and their inversed counterparts (Fig. 6).

The Matlab/Simulink model that reproduces the FPGA processors-based TMR system and simulates the waveforms presented in Fig. 6 is given in Fig. 7. The three individual development systems that generate signals on their four bit output buses (*Out1*, *Out2*, *Out3*, and *Out4*) are evidenced. These signals are plotted and monitored on *Scope1*. Then the generated waveforms are captured by 1 bit digital voter circuits. Supposing in a first attempt an ideal TMR configuration where the three hardware units operation is perfectly synchronized in time and their outputs match in all, results a proper operation of the system and the voter passes toward exactly the same signal waveforms as shown in Fig. 6.

A very different situation occurs when unsynchronized signals set are generated via the FPGA-based development systems output buses. In this case the TMR operation is totally compromised and wrong or misleading signals set is passed toward by the digital voter elements. This state may be clearly observed by studying the signals set plotted on Fig. 8. There unwanted hazard pulses or additional transient states can be detected that disturb the proper operation of the voter unit.

However, as long as the three modules are driven by their independent clock systems they will operate in an asynchronous manner in comparison to each other. Therefore, it is clear that their output buses also will remain unsynchronized in time domain. In these circumstances the initial

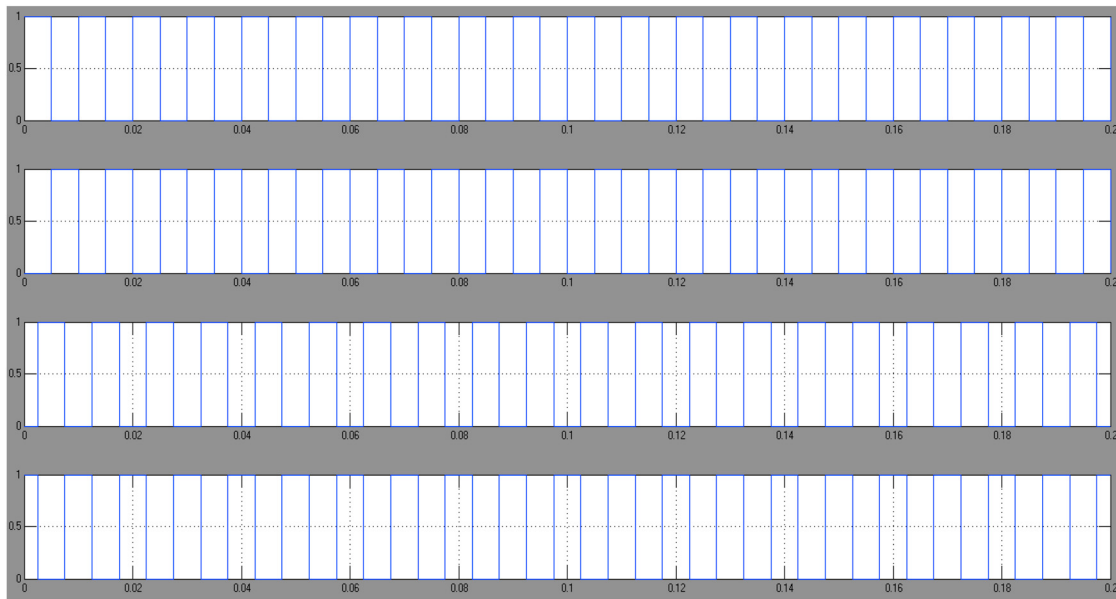


Figure 6. The 4 bit signals set generated by the FPGA-based development boards [10]

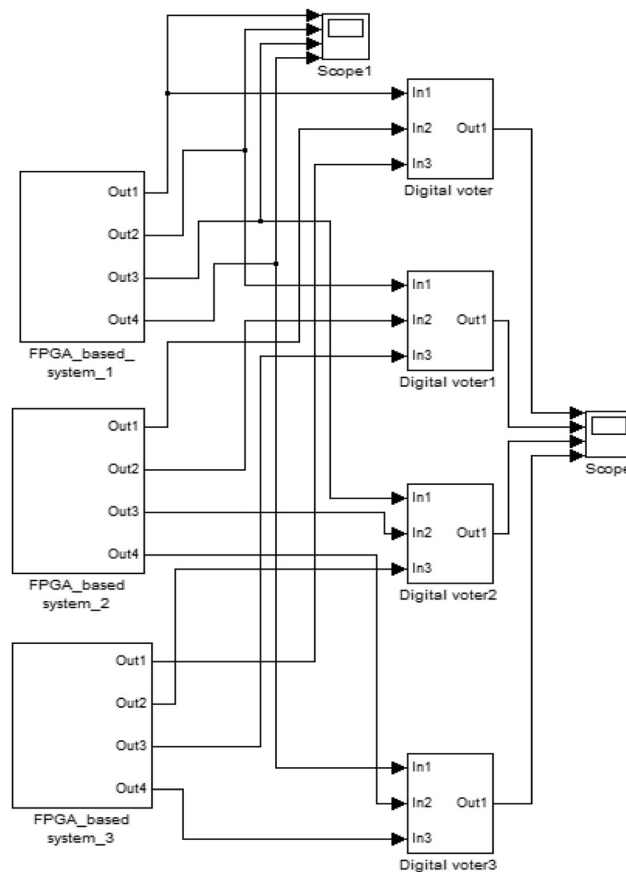


Figure 7. The Simulink model of the TMR system (Szász 2017)

crucial assumption adopted by a wide range of researchers in their works cannot be maintained at all. Of course, one welcome solution for this bottleneck could be the idea to use only one clock system that drives all the three FPGA-based

development systems. In this case the first major requirement is of course to upload the three modules with program tasks that generate the same signals waveform set and their execution time remains perfectly equal. In other words they

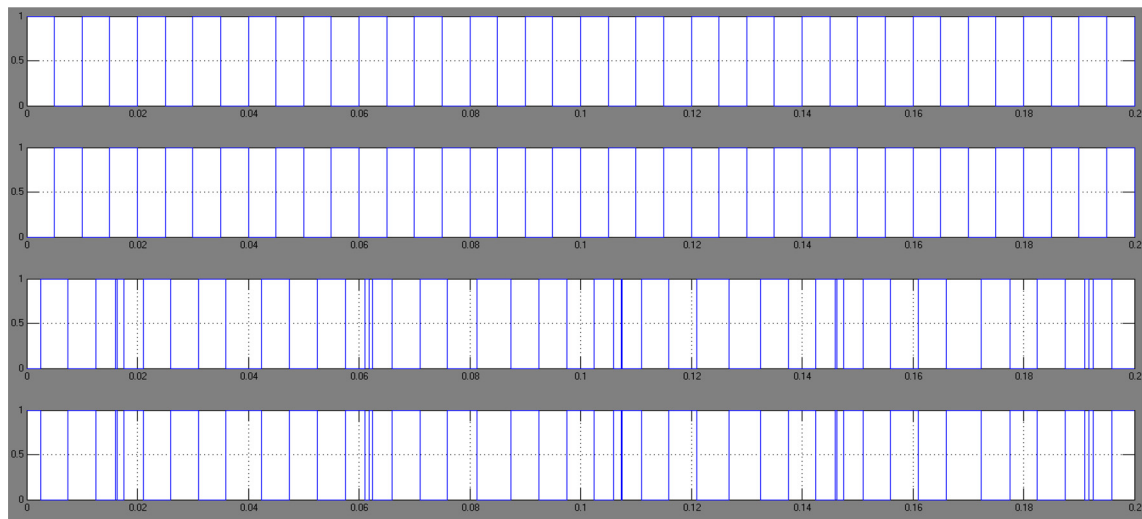


Figure 8. Wrong signals passed by the voter unit and the TMR system operation is compromised [10]

require exactly the same clock ticking. Unfortunately, in the specific case of so complex digital systems this is a very difficult programming undertaking – in most cases, even an unreachable software engineering goal. The explanation of this inconvenience is relatively simple: if the execution time of these tasks differs only by one clock tick (that means *ns* range time unit) this will be accumulated in time.

In other words, by running thousands or even billions of times the same task the initial time small shift even between the same programs will be continuously accumulated in time. Therefore, the occurred delays tend to linearly increase in time. This worsening synchronization process also may be clearly observed by plotting the waveforms set on the output buses of the three FPGA-based modules.

Because of difficulty to handle this synchronization problem, in this paper an original solution to overcome the mentioned bottleneck is introduced. The essence of this proposal is to implement a so-called “on-chip” TMR structure on each stand-alone FPGA-based development board, as shown next in Fig. 9. There is a Simulink model plotted where each module embeds a full TMR structure as it has presented in the previous paragraph in Fig. 1. Additionally, the boards supervise each other’s operation mode. It means that if one development board enters into a failure state and its operation mode fails, the next unit will take over all the functions and functionality of the faulted ones. In this way the whole hardware system becomes fault-tolerant and the occurred faults will be masked inside hardware redundant system.

In Fig. 9 a logical OR operator has been connected to each bit of the output buses, the *Scope* block plots the waveforms corresponding to proper operation of the TMR configuration, respectively *Scope1* plots the waveforms on the three buses (12 bits). As it is possible to observe from the corresponding time diagram, in the first time interval (0–0.05s) operates only the first FPGA-based development system (the *FPGA-based\_system\_1*). At the time moment

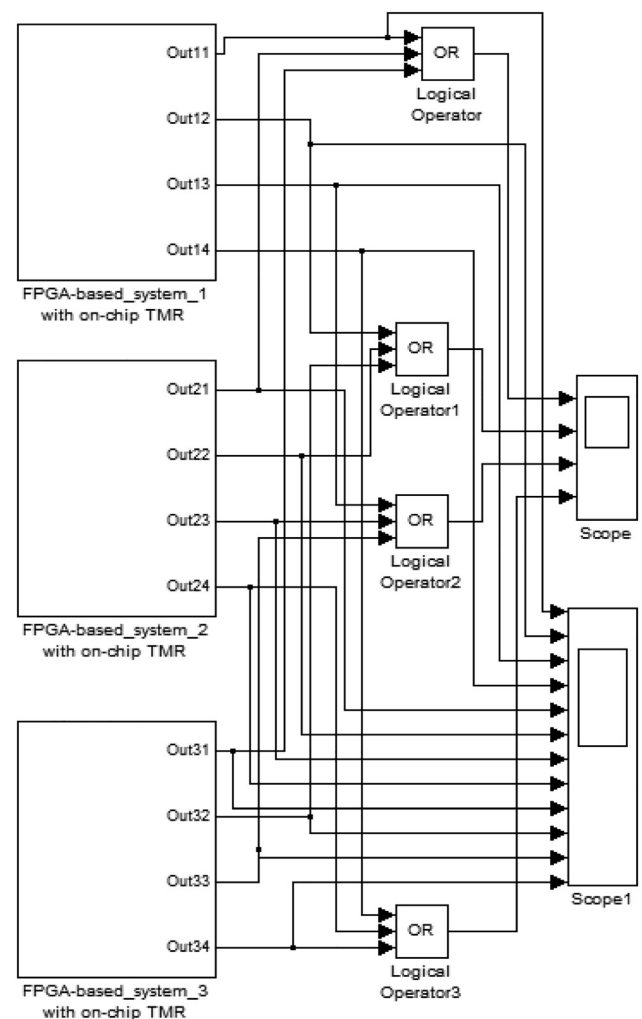


Figure 9. The proposed fault-tolerant TMR structure



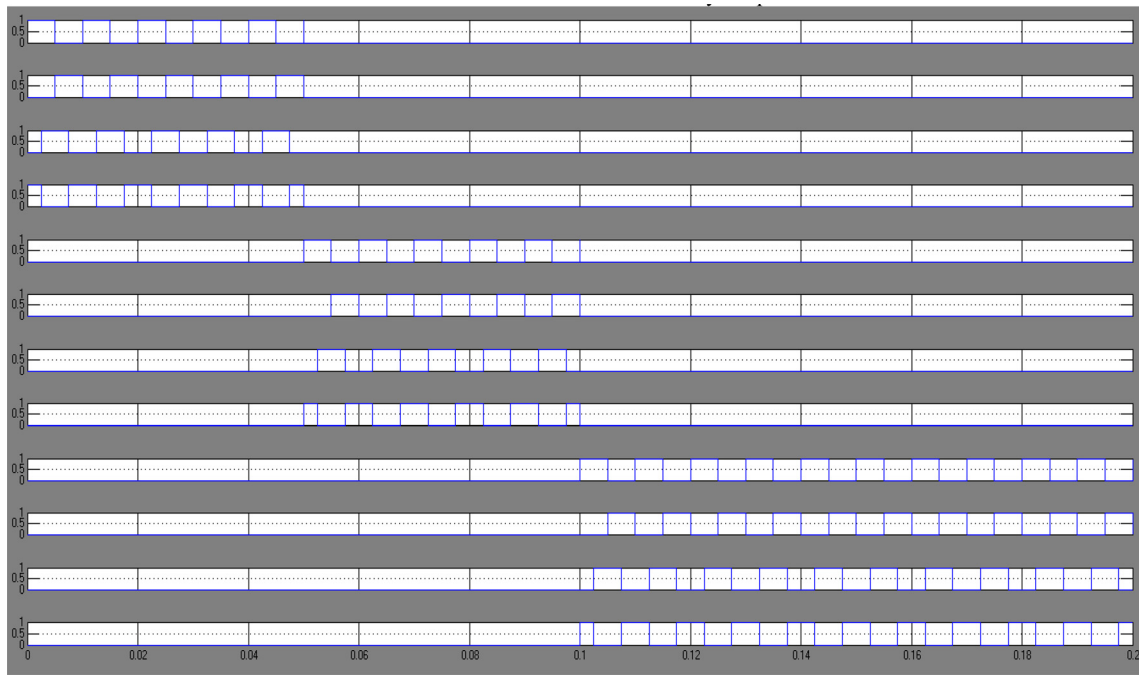


Figure 10. The waveform set corresponding to the fault-tolerant TMR system operation mode

0.05s a faulty state occurs and this system remains out of order. In this case enters into a full operation mode the second available redundant unit (the *FPGA-based\_system\_2*) and generates the same waveforms on its output buses as its previous counterpart. In a next stage, at the time moment 0.1 s, fails the second hardware module. The result is that instantaneously it activates the last remaining redundant module *FPGA-based\_system\_3*. This will operate continuously with the same functions and functionality inside the system.

As it is possible to observe from the above presented TMR configuration models and simulation results obtained by using the Matlab/Simulink software environment the synchronization in complex digital redundant systems is not such a trivial problem. Of course, by adopting crucial assumptions as has been discussed in the introductory paragraph, this problem becomes one relatively simple, as shown in Fig. 10.

Otherwise, without these important simplifications synchronization and matching in all the outputs of redundant digital systems is a difficult engineering undertaking.

## 4. CONCLUSIONS

This paper points out the main bottlenecks regarding the complex digital systems-based TMR configurations operation mode. It underlines the critical assumptions that are adopted by the vast majority of researchers in their scientific publications. It discusses the influence of these assumptions in synchronization and matching in all the outputs of redundant

digital systems. The developed models and simulation results presented in this paper evidences that it is not so trivial to solve such problems. For this reason a concrete solution has been presented on how to overcome unwanted operation modes of TMR topologies and to reach and implement highly fault-tolerant redundant digital systems.

## REFERENCES

- [1] B. W. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems*. Addison-Wesley series in electrical and computer engineering, 1989. ISBN: 0-201-07570-9.
- [2] O. Ruano, J. A. Maestro, and P. Reviriego, "A methodology for automatic insertion of selective TMR in digital circuits affected by SEUs," *IEEE Trans. Nanotechnol.*, vol. 56, no. 4, pp. 441–451, 2009.
- [3] J. Vial, A. Bosio, and P. Girard, "Using TMR architectures for yield improvement," in *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, pp. 7–15, 2008.
- [4] T. Ban and L. Lavier, "Optimized robust digital voter in TMR designs," *Researchgate Article*, v. 1, P. 1-3, June 2011. [https://www.researchgate.net/publication/267411067\\_Optimized\\_Robust\\_Digital\\_Voter\\_in\\_TMR\\_Designs](https://www.researchgate.net/publication/267411067_Optimized_Robust_Digital_Voter_in_TMR_Designs).
- [5] P. Rahman, S. Rafique, and M. Alam, "A fault tolerant voter circuit for triple modular redundant system," *J. Electr. Electron. Eng.*, vol. 5, no. 5, pp. 149–159, 2017.
- [6] P. Balasubramanian, K. Prasad, and N. Mastorakis, "A fault tolerance improved majority voter for TMR system architectures," *WSEAS Trans. Circ. Syst.*, E-ISSN: 2224-266X, vol. 15, pp. 108–122, 2017.

- [7] D. Davis, and J. Wakerly, "Synchronization and matching in redundant systems," *IEEE Trans. Comput.*, vol. 27, no. 6, pp. 531–536, 1978.
- [8] J. Neumann, "*Probabilistic logics*", *Auromafa Studies*. Princeton University Press, 1956, pp. 108–122.
- [9] Digilent Co, 2016. <http://store.digilentinc.com/fpga-programmable-logic/system-boards/>.
- [10] Cs. Szász and R. Sinca, "Synchronization strategy in complex digital voter-based systems," *J. Electr. Electron. Eng.*, vol. 10, no. 1, pp. 55–60, 2017. P-ISSN: 1844-6035.