FPGA Based Design of the Railway's Interlocking Equipments

Radek Dobias, Hana Kubatova
Department of Computer Science and Engineering, Czech Technical University Prague
Karlovo namesti 13, 12135 Prague 2, Tel.: +420 2 2435 7281, Fax.: +420 2 2492 3325
dobiasr@fel.cvut.cz, kubatova@fel.cvut.cz

Abstract

This paper describes the architecture of a safety system of the railway's interlocking equipment, which has been developed for Czech railways. The system will be used for the railway crossing gate. This system is based on FPGA blocks and has to fulfil the requirements for a fault tolerant system with a fail-safe function. The dual logic and TMR principle are used to increase its dependability. Several self-test and self-diagnostics features are used, such as an LFSR based built-in self-test, the FPGA readback and 1 out of 2 error detection codes. The functional logic uses a majority correction and the FPGA box reprogramming to precede the failure. The reliability analyses, models and reliability characteristics calculations of this system are described. Markov chain models are used for the reliability analyses. The TMR principles for fault tolerant system and the Dual-TMR logic have been used in our design and both attempts are compared.

1. Introduction

The electronics blocks are not often used for the railway's interlocking equipment at this time. The safety function in the railway's application was always based on the gravitational attraction (e.g. by relays or mechanical signals) for the stop-signals and on the mechanical pull or on the big value of the electrical current for the permit signal. It is very difficult to prove that the interlocking equipment with the electronics blocks can be safe. The railway's operators are afraid of these blocks' unreliability and dangerousness. This paper shows that the fears of railway's operators are unjustified.

Since the electronic blocks were successfully used in the space program, the railway's operators have accepted to use these blocks in railway's interlocking equipment. New designed systems usually apply microprocessors, programmable or ASIC blocks are utilised only rarely. But the features of the programmable gate arrays (FPGA, CLPD) predestine these blocks for wide use in the railway's applications.

Authors of this paper want to show advantageous features of FPGAs for the design and reliability calculations of the railway's interlocking equipment. The design of the safe interlocking equipment is shown. Various techniques for increasing the dependability are described. It is the first attempt to design and use the system with FPGA blocks in a safe railway's application for Czech railways.

The paper has the following structure: Section 2 defines the basic knowledge for the railway's interlocking plant and the main principles of our design. The safety in traffic process problems are described in Section 3. The architecture of designed interlocking plant is described in Section 4, Section 5 shows the calculation of reliability characteristics and its results. Section 6 summarises the advantage features of FPGA and Section 7 contains conclusions.

2. Safety of the railway's traffic process

The railway's traffic system has only one degree of freedom (as compare with the other traffic systems). The train can move only forward and rearward. This aspect makes possible an easy detection of the train on the track. One degree of freedom of the railway's traffic system allows to define the safety state for trains. This state is defined by the stop-signal for all trains in a controlled area. When the railway control system place an order with the stop-signal, all traffic must be stopped and can continue only by the direct orders from the human operators. After that, the traffic system is controlled by human operators without any support from the control system. In this moment, the human operators have all responsibility for safety of the traffic process.

The probability of a dangerous behaviour is higher for the human operators than for the railway's interlocking equipment. Therefore, it is very important for the railway's control and interlocking system to be highly reliable, available, maintainable and safe.

TABLE I. Type of redundancy for use in fault-tolerant railway's interlocking equipment

Type of Redundancy	***Implementation***	***Type of Detected Errors***
Time redundancy	The same software is executed on the same hardware during two different time intervals	Errors caused by transient physical fault in the hardware with a duration of less than one execution time slot
Hardware redundancy	The same software is executed on two identical hardware channels	Errors caused by transient physical fault in the hardware
Diverse hardware	The same software is executed on two different hardware channels	Errors caused by transient and permanent physical faults in the hardware
Diverse software	The different software versions are executed on the same hardware during two different time intervals	Errors caused by independent software faults and transient physical faults in the hardware with a duration of less than one execution time slot
Diverse software on the redundant hardware	The different software versions are executed on two identical hardware channels	Errors caused by independent software faults and transient physical fault in the hardware
Diverse software on the diverse hardware	The different software versions are executed on two different hardware channels	Errors caused by independent software faults and transient and permanent physical fault in the hardware

3. Basic knowledge for the railway's interlocking equipment designed with electronics blocks

A real-time computer system (e.g. railway's interlocking equipment) must react to inputs from controlled object and from the operator. The instant at which a result must be produced is called a deadline. If by missing a firm deadline a catastrophe could happen, then the deadline is called hard. A real-time computer system that must meet at least one hard deadline is called a hard real-time computer system or a safety-critical real-time computer system.

The fault tolerance is very important in the safety-critical real-time applications, if one component fails it can cause the critical failure of the complete system (missing a hard deadline). Therefore the error detection is very important in such system. The error detection needs good knowledge about the system behaviour. This knowledge is based on the regular definition of the system behaviour or on the comparison of two or more redundant systems reactions.

There are many failure types in the fault-tolerant system like a fail-silent, a fail-consistent and a malicious failure. The specific type of a fail-silent failure is power-off of the component. This reaction is often used in the railway's interlocking equipment, but it is not correct. For example, in the communication problem between hardware channels is a high probability that the connection will be restored therefore the power-off reaction is not necessary.

The general method for the high probability of the fail-safe function is described in Fig. 1. The guard block is implemented by hardware or software parts and validates the output data. The output information of the guard block is a valid/error property for the output data.

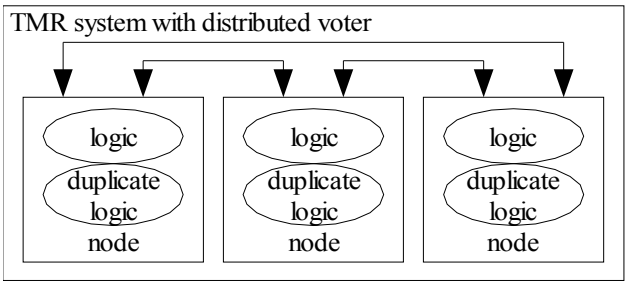

Figure 1. The general scheme for fail-safe systems

The guard block can be implemented identically as a function component. If the function component is a block of logical functions, the guard block presents the same logic. This part is than called a duplicate logic. It presents the principle ”2 out of 2”.

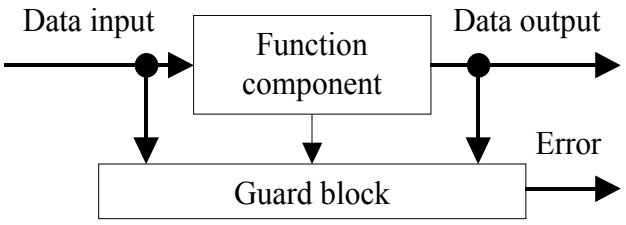

Figure 2. The Dual TMR system with duplicate logic in each node

The railway's interlocking equipment based on the principle ”2 out of 2” is not tolerant to any fault. But this system can be fail-safe, of course. The fault-tolerant systems work on the principle ”2 out of 3” (TMR). The most safety and reliable railway's systems are based on the Dual-TMR principle. There are two possibilities how to design the Dual-TMR safe systems.

Firstly two TMR parts with same function and these parts cooperate on the principle ”2 out of 2”. One of these

two TMR parts can be based on the negative logic. Secondly, only one TMR system can be used, but each node works on the principle ”2 out of 2”. It is possible if the duplicate logic block is in the negative logic.

4. Architecture of the railway's interlocking equipment with FPGA

The bases of this interlocking equipment are three hardware channels with FPGAs, located on the separate boards. The system works on the Dual-TMR principle. If 2 out of 3 hardware channels (TMR node) work, the system can signal the safety-critical commands.

A system must process and propagate data correctly even if the configuration and/or user logic fails, to be considered reliable and safety. The design has to be transient fault resistant. The techniques of the fail detection, correction and mitigation must be combined to build a reliable FPGA system.

4.1. Two-wire duplicate logic

One possible fail-safe system design is the duplicate logic utilization. In our system the duplicate logic is implemented by the two-wire logic. The two-wire logic uses the code “1 out of 2” for the variable representation. The basic logical blocks for the two-wire logic are on Fig. 3.

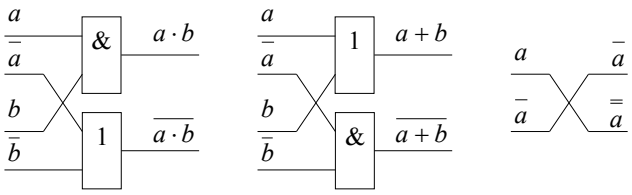

Figure 3. Basic blocks for two-wire logic (AND, OR, NOT)

When the two-wire logic is used for a general system design, the number of used blocks increases and the reliability of the system is reduced. But when the FPGAs are used, the number of blocks is stable. The designer of these FPGA blocks assigns only one value of MTTF abstractedly from the function. It means, that more logical functions implemented in one FPGA block don't increase the unreliability.

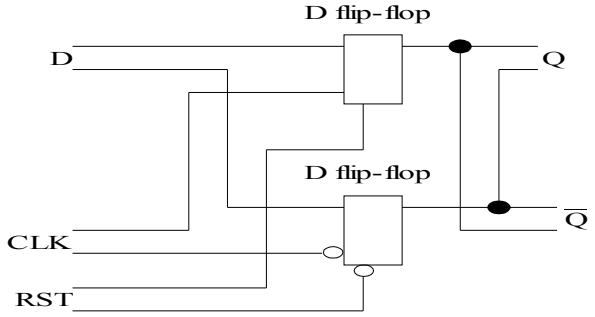

Figure 4. D flip-flop for two-wire logic

In future we will find out an optimal size of several logic blocks for various types of FPGAs. At proper optimisation would be possible to add an error detection block into every CLB block. The error detection block detects the error in the calculation of an implemented function. The next possibility is to detect these errors in the function results or only for outputs of the FPGA block.

4.2. Error detection

The error detection block is used for the outputs of several logic parts. This block is implemented as a checker for “1 out of 2” code. All signals of the logical function are checked by this block. The error detection block has a very simply structure, which is showed on Fig. 5.

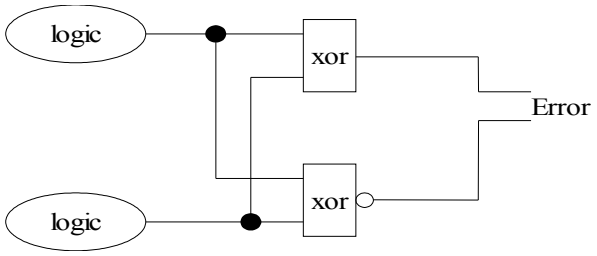

Figure 5. Error detection in duplicate logic

The Error signal is also in “1 out of 2” code. We have used the Error signals buses in the FPGA block. The Error signals buses are also checked and provide information about whole logic.

4.3. Time schedule of the system function

The railway's interlocking equipment is a hard real-time system. The main advantage of this type of real time system comparing with the other hard real-time systems is that the hard deadline is approximately up to 5 second. The railway's interlocking plant doesn't need an immediate response in a case of external event rise. It can be designed as a time-trigged real time system. The period 100-1000 ms of trigger event is used generally. In our system the used period is 500 ms, the internal clock run at 5 MHz.

It is necessary to make these base tasks during the 500 ms cycle:

- to read data from inputs of interlocking plant
- to exchange input data between TMR nodes
- to compare input data and to determine a value for computation on the base of majority
- to check validity of internal state of finite state machine (FSM)
- to determine new internal states and output values
- to exchange internal states and outputs between nodes
- to determine output values and internal states on the base of majority
- to set output values on pins of FPGA block
- the most of the time is need for communications between nodes, other tasks take a few time.

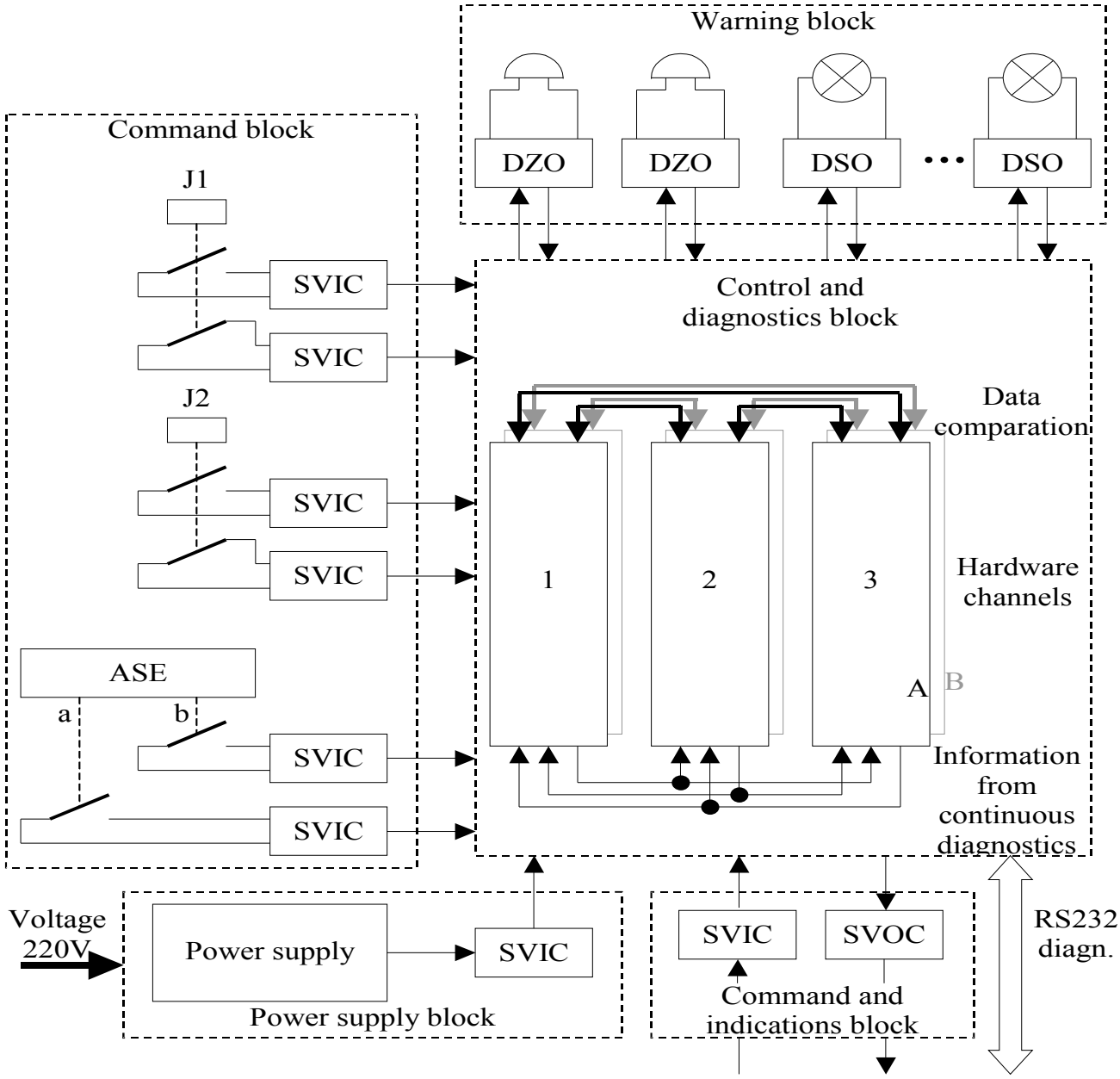

Figure 7. The architecture of the railway's crossing interlocking plant with FPGAs
J1,J2 - safety track circuits, ASE - reliable point train detector, SVIC - safe voltage input circuit, SVOC - safe voltage output circuit, 1 2 3 - hardware channels based on FPGA with A and B parts in the FPGA block , DZO - check circuits for signalling bell, DSO- check circuits for signalling lamp

4.4. Non-concurrent on-line testing

In the previous section is shown that the functional logic is mostly not in use. This is an eventual time to check the logical function. The check of logical function in such time, when the system is not used for useful function, is called non-concurrent on-line testing. Fig. 6 shows, how the test runs.

The BIST structure for on-line testing of duplicate logic behaviour is used. The test pattern generator and the response analyser are based on Linear Feedback Shift Registers (LFSR). This structure is implemented in each FPGA block.

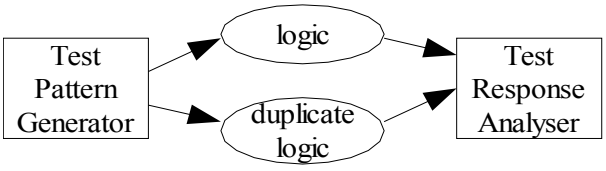

Figure 6. Built in Self Test

4.5. Reprogramming

The function of the hardware channels is programmable. When a fail in the program logic is detected via the "readback", the program is re-loaded to the FPGA block. The time needed for reprogramming of the FPGA block depends on concrete FPGA type, but it is not longer than 400 ms (5MHz) [12]. The state of the FSM is automatically restored trough the majority function from the other TMR blocks during the next function cycle. In a case when a FPGA block supports a partial reconfiguration, only part of FPGA block with a detected fault is reprogrammed.

4.6. General function description

The input data are compared between hardware channels and input values are evaluated by a majority function. These values are converted into the two-wire logic for use in an algorithm. The algorithm is executed on the hardware channels independently. The output values are also compared.

The outputs from the hardware channel can be disabled, when a channel fault is detected by self-checker or outputs are permanently different from other channels. The reset signal for this channel is set when the interlocking equipment is in the base state. The complex test is executed after a reset.

When one hardware channel is disabled and the next channel detects a fault, the system is degraded, all safety-critical commands are cancelled and only emergency commands are allowed.

All hardware channels use the same algorithm, but the logical functions are located in the different part of the FPGA block. It represents the hardware diversification. The two-wire logic is used for all logical functions, for a FSM and for a checker. All channels exploit continuous diagnosis for self-testing.

Application of this architecture in the railway's crossing interlocking plant is in the Fig 7.

5. Reliability of the interlocking equipment

The SHARPE software [11] has been used for the reliability modelling and calculations. This software tool computes the reliability characteristics from a block diagrams, Markov-chains, tree analyses, etc. The hierarchical structure of the models can be exploited.

The MTTR (mean time to repair) equals to 24 hour (it is the real value obtained from Czech railways) has been assumed for reliability modelling of this interlocking equipment, while the system is usually repaired according a demand of the service company. The XILINX FPGA blocks are used. For this FPGA block XILINX Company determines MTTF as 2.8×10^8 hours. This value was decreased to 2.5×10^5 MTTF because not only FPGA blocks have been used. It means that MTTF is just about 28 years.

5.1. TMR model for the interlocking equipment

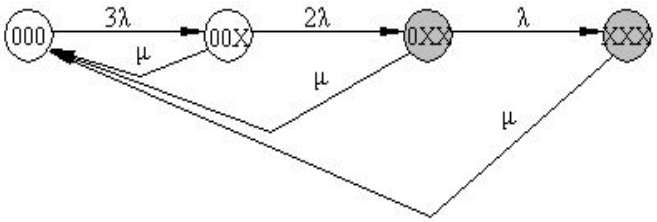

Figure 8. Markov-chain model of TMR

The Markov model in Fig. 8 describes the reliability characteristics for FPGA blocks without support of partial reprogramming feature. Then the usual TMR arrangement for interlocking equipment should be used. In comparison of our model with TMR models shown in literature is our model different, because MTTR does not depend on the failure size. This premise is true when a maintenance man repairs the equipment by the board(hardware channel) replacement.

The model in Fig. 8 represents the TMR architecture of the interlocking equipment. The value in the circle represents which hardware channels are with (X) or without (0) a failure. A white colour is used for functional states and grey one for faulty states. Fig. 8 represents full, not simplified model.

The equation system is determined for probability of a steady state. The incoming and outgoing rates must be equal for every state. The equation system is completed with a normalisation condition.

$$3\lambda\, p_{OOO} = \mu\, p_{OOX} + \mu\, p_{OXX} + \mu\, p_{XXX}$$
$$(2\lambda + \mu)\, p_{OOX} = 3\lambda\, p_{OOO}$$
$$(\lambda + \mu)\, p_{OXX} = 2\lambda\, p_{OOX}$$
$$\mu\, p_{XXX} = \lambda\, p_{OXX}$$
$$p_{OOO} + p_{OOX} + p_{OXX} + p_{XXX} = 1$$

The necessary conditions for this model are: the model reflects only probability of the stable states; the interlocking equipment in the fail-free state after power-on of the system. This model was evaluated with these results:

$$p_{OOO} = 0.999712083$$
$$p_{OOX} = 2.87861810 . 10^{-4}$$
$$p_{OXX} = 5.52641622 . 10^{-8}$$
$$p_{XXX} = 5.30535958 . 10^{-12}$$

The value of the steady-state availability ASS is a sum of probabilities for all fail-free states.

$$A_{SS} = 0.999999945$$
$$MTBF = 4.34236111 . 10^{8}\,h$$

This great value of the MTBF exceeds thinkable livetime for the railway's interlocking equipment, but this value is important, when more applications of the interlocking plant are really used in the traffic process. Then, it means the lower cost for maintenance of the interlocking plant.

5.2. Dual-TMR model for the interlocking equipment

The Dual-TMR arrangement provides a considerable improvement of reliability characteristics.

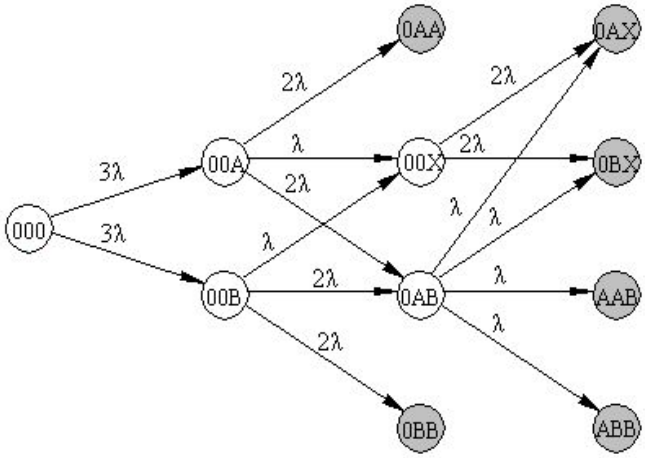

Figure 9. Markov-chain model of Dual-TMR

O letter in Fig. 9 means the same as in previous model, A and B letters denote a faulty part of the FPGA block and X letter means that the both parts are faulty. The whole equipment operates if at most one node from each TMR part is faulty. In this case there is at least one hardware channel, which can compare an output data between the logic and duplicate logic without a fault and can produce the output data. The feedback edges from each state to the OOO state (i.e. to the fail-free state) described by the repair intensity μ are not drawn in Fig. 9. But the next calculations take them into account, of course. Fig. 9 represents partly simplified model for better understanding.

The following results were obtained from this model by SHARPE:

$$p_{OOO} = 0.999712083$$
$$p_{OOA} = p_{OOB} = 1.43923998 . 10^{-4}$$
$$p_{OAA} = p_{OBB} = 1.38167038 . 10^{-8}$$
$$p_{OOX} = 1.38140515 . 10^{-8}$$
$$p_{OAB} = 2.76281031 . 10^{-8}$$
$$p_{OAX} = p_{OBX} = 2.65229789 . 10^{-12}$$
$$p_{AAB} = p_{ABB} = 1.32614895 . 10^{-12}$$

The value of the steady-state availability ASS is a sum of probability for all fail-free states.

$$A_{SS} = 0.999999972$$
$$MTBF = 8.68263913 . 10^{8} \text{ h}$$

These results show that Dual-TMR system has two times greater MTBF then TMR. This fact could be expected because 2 independent TMR systems were used. But if the second system is implemented in the same FPGA circuit and its design is only some extension of the first one, the implementation is advantageous, because the design costs and the hardware overhead are not two times greater. Therefore this system is better than a classic hot back-up [1, 2].

Next improvements (e.g. reconfiguration of TMR like TMR/S or TMR/S/S are subjects of our future research.

6. Advantages of FPGAs

FPGAs have (in comparison with microprocessor based solutions) the following advantages:

The function of the FPGA block (chip) is programmable. It means, that the checkers for continuous (on-line) testing and a guard block could be integrated inside the block. It is not necessary to add these testing and diagnostic functions to the higher layers of the system design.

The function of a microprocessor block (chip) is complicated. It should be, because microprocessors are designed for the general use. But it complicates testing and diagnosis of this microprocessors. Some functions of the processor are not used in a common process. It means that some parts of the processor are not used. But these not used parts can be important in the safety critical moments of the traffic process. The faults in these parts of a processor remain long time latency.

Next advantage of FPGAs is parallel processing of the function in FPGA, in comparison with the sequential one in a processor. It means that the reaction time on input signals is limited only by the clock frequency. This is a great advantage of the real-time systems.

The VHDL language can be utilized for the implementation of the railway's interlocking equipment with FPGAs. With the help of this tool most simulations and verifications of the logical function of the interlocking equipment are created and made before their concrete realization. These results are used in the validation process for the real using and operation by the Czech railways.

7. Conclusions

From the presented and computed values of the reliability characteristics follows that the railway's interlocking plant with FPGAs is at least so good, as other railway's interlocking plants with the processors or with the relays .

The interlocking equipment of the designed railway's crossing interlocking plant has 1000 times better MTBF that recently used railway's crossing interlocking plants and has 4 times better MTBF of the complete interlocking plant with all necessary but not too reliable peripheries.

It is interesting to know that all railway's crossing interlocking plants used by the Czech railways now, have the safe interlocking equipment (with the safety validation), but have no fault tolerant interlocking equipment. Only this new interlocking plant is designed as a fault-tolerant system.

This interlocking plant uses Dual-TMR arrangement and satisfies all standard requirements for system with safety integrity level 4 (SIL 4) [4, 5].

The first safe interlocking plant with FPGA blocks based on principles described in this paper is now in a developing process. The pilot project of the fault-tolerant equipment is planned to be validated and used in a trial run during this year (2004).

The authors are convinced that new designed interlocking plants should be designed as a fault-tolerant system. In connection with the recommendation that all faults in the redundant parts are repaired within 24 hours, the railway's interlocking plants will be more reliable and safe.

The authors suppose, that this work contributes to more expansion of FPGAs and other microelectronics blocks in safety critical applications.

Acknowledgement

This research has been in part supported by the GA102/03/0672 grant.

References

[1] Pradhan, D. K.: Fault-tolerant computer system design, Prentice Hall ptr, New Jersey, 1995, 550 p

[2] Kopetz, H.: Reel-time systems: design principles for distributed embedded applications, Kluwert Academic Publishers, Boston / Dortrecht / London, 1997, 338 p

[3] Dobias, R.: Reliability evaluation of the railway's interlocking plants, with design of the railway's crossing interlocking plant based on the FPGA, diploma thesis, CTU Prague, 2003, 64 p (in Czech)

[4] Storey, N.: Safety-critical computer systems, Prentice Hall ptr, New Jersey, 1996, 453 p

[5] EN 50126: Railway application : reliability, availability, maintainability and safety. Celenec, 1997

[6] Faran, A., Srb, S.: The railway interlocking systems, Czech technical university, Prague, in press (in Czech)

[7] Powel, D.: A generic fault-tolerant architecture for real time dependable systems, Kluwert Academic Publishers, Boston, 2001, 242 p

[8] Brinkley P., Carmichael, A., Carmichael, C.: SEU Mitigation Design Techniques for the XQR4000XL, Xilinx Application note, 2000, 14 p

[9] Fernandes D., Harris I.: Application of Built in Self Test to Interconnect Testing of FPGAs, North Atlantic Test WorkShop 2002, 5 p

[10] Itazaki N., Matsuki F., Matsumoto Y., Kinoshita K.: Built-In Self-Test for multiple CLB faults of a LUT Type FPGA, 7th. Asian Test Symposium, Singapore, 1998, 6 p

[11] Hirel C., Sahner R., Zang X., Trivedi K.: Reliability and Performability Modelling using SHARPE 2000, 11th International Conference, TOOLS 2000, Schaumburg, US, 2000, 5 p

[12] Xilinx Inc.: Virtex FPGA Series Configuration and Readback, Application Note XAPP138, v2.7 edition, 2003

[13] Entrena L., López C., Olías E.: Automatic Generation of Fault Tolerant VHDL Designs in RTL, Forum on Design Languages 2001, Lyon France, 5p