Reliable Railway Station System based on Regular Structure implemented in FPGA

Jaroslav Borecký [1,2], Pavel Kubalík [1,2], Hana Kubátová [1]
e-mail: borecj2@fel.cvut.cz, xkubalik@fel.cvut.cz, kubatova@fel.cvut.cz

[1] Dept. of Computer Science and Engineering
Czech Technical University in Prague
Prague, Czech Republic

[2] IKT Advanced Technologies, s. r. o.
Prague, Czech Republic

Abstract—**The method how to design a safety device of railway station efficiently and scalable is proposed. The safety device for any configuration of railway station can be built from five basic blocks. These basic blocks are connected together with universal interface. Each block is based on a finite state machine. The finite state machines are "Moore" type. Each state machine is divided into three basic parts, where each part is designed as a self-checking circuit ensuring fault detection. Our methodology is intended for final implementation in FPGA and hence SEU faults occurring in the system is assumed.**

Keywords: Fault Tolerant design, FPGA, Finite State Machine, SEU, Secure Device, Railway Station.

I. Introduction

Systems realized by programmable hardware like Field Programmable Gate Arrays (FPGAs) are more and more popular and widely used in more and more applications due to several advantages, like their high flexibility in achieving multiple requirements such as cost, performance and turnaround time and the possible reconfiguration and actual changes of the implemented circuit, e.g., only via wireless connections.

The FPGA circuits should be used in mission critical applications such as aviation, medicine, space missions, and railway applications as well [1, 2, 3]. Many FPGAs are based on SRAM memories sensitive to Single Even Upsets (SEUs), therefore a simple usage of FPGA circuits in mission critical applications without using any method of error detection (and then correction or minimally safe behavior when a fault or error happened) is impossible.

A change of one bit in the configuration memory leads to a change of the circuit function, often drastically. The Concurrent Error Detection (CED) techniques allow detection of soft errors (errors which can be corrected by reconfiguration) caused by SEUs [4, 5, 6]. SEUs can change also the content of the embedded memory, Look-up Tables (LUTs) and other configuration bits. These changes are not detectable by off-line testing methods therefore CED techniques have to be used. The probability of a SEU occurrence in the SRAM is described in [7, 8].

The self-checking (SC) structure is used to detect an occurrence of a fault in the tested circuit. Only one copy of the SC circuit is not sufficient to increase dependability parameters. Thus, we assume to use the Modified Duplex System (MDS) architecture [9, 10].

Nowadays the safety device of a railway station is in many cases realized by several functional blocks based on relays. Such devices have been very popular due to their high safety factor ensured by a structure corresponding with a railway scheme. Current research deals with systems based on two or more parallel working processor (instead of relays). The safety property of this system depends more on a human factor due to properties based and given by software. The safety device based on processors is described in [26].

The safety device of the railway station is based on five blocks each realized by a finite state machine (FSM). Each block is designed as a self-checking circuit. The final dependability design is based on the MDS architecture principles [9, 10].

The self checking circuit quality is determined by an area overhead and fault security (FS) parameters. These parameters are important to compare the quality of basic blocks realizing safety device of railway station.

The paper is structured as follows: the principles of the self-checking circuit are described in Section 2. The method of fault security calculations is presented in Section 3. Then the method of blocks connection with respect to FS parameter is described in Section 4, the architecture of the safety device of a railway station is stated in Section 5, section 6 describes the design methodology of the FSM self-checking blocks. Section 7 contains the experimental results and Section 8 concludes the paper.

II. THE PARITY GENERATOR

CED techniques are widely used to increase the system dependability parameters. Almost all CED techniques are based on the original circuit, an error detection code predictor and a checker. The predictor predicts checking nets of used error detection code from primary inputs.

A checker unit is used to control an output code. When an error is detected, the fail signal is generated. The self-checking basic structure of CED technique is shown in Figure 1.

It is important to say, that some hardware redundancy is required for predictor and checker.

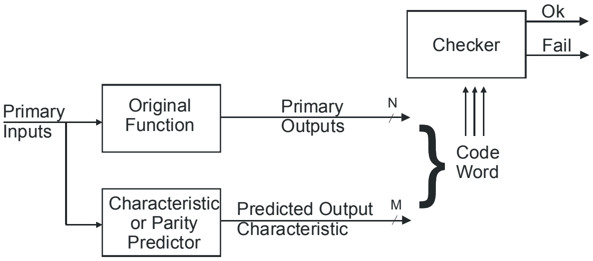

Figure 1. The self-checking circuit

Even 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, 13, 14].

III. FAULT SECURITY CALCULATION

There are three basic quantitative criteria in a field of CED: fault security (FS), self-testing (ST) and totally self-checking (TSC) properties [15]. These three aspects have to be used in the 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, the possible faults should be classified and separate into four classes, A, B, C and D [16]

□ 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. 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. They are not detectable by any other input vector. Faults from this class cause undetectable errors. If any fault in a 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 the level of satisfaction of FS or ST properties of the designed circuit and then calculate TSC properties.

IV. INTERCONNECTION OF INDIVIDUAL SC BLOCKS

Each FPGA contains a TSC circuit and a comparator in our methodology. 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 2.

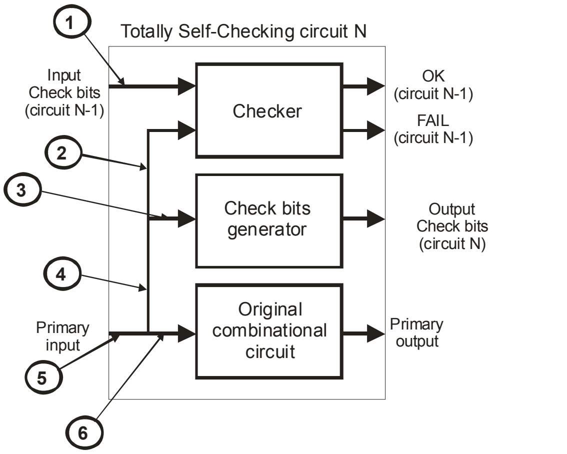

Figure 2. Proposed structure of TSC circuits implemented in FPGA

Six places where an error is observable for this compound design has assumed. It is assumed, 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 block N will detect an error if it occurs in the 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 greater detail in [10].

Not every small block (in the compound design) satisfies the TSC property to 100%. The TSC property depends on the 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.

V. RAILWAY STATION SYSTEM

The safety railway station systems are in many cases based on block containing relay unit. Security property is given by possible hardware combination. It means that one path cannot be set with using occupied block. Hardware configuration do not allows set it. This approach is different from PC based system, where possibility of setting of a free path is defined by rules. The rules are set by programmer and his mistake could lead to an accident. The relay based systems are more complex ones and they obviously occupy higher area. The proposed scheme of a simple railway station based on the relay blocks is shown in Figure 3.

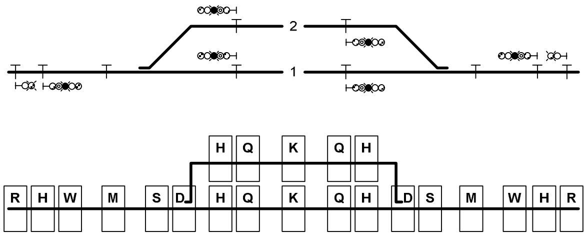

Figure 3. Simple railway station with relay based blocks

The new system based on FPGAs was investigated in our department in several diploma thesis, e.g. in [17]. First attempt to convert relay based system into FPGA hardware was difficult and after some experiments was refused. New method assumes a new approach, where original blocks were generated independently only according the knowledge of the common system functions.

The new proposed system uses the same blocks as a relay based system but the function implemented inside the blocks and communications between these blocks are completely different. Each block is based on finite state machine (FSM).

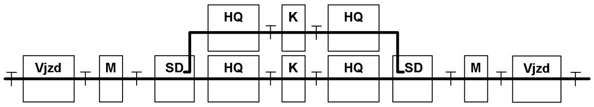

Figure 4. Simple railway station with new FSM based blocks

Some blocks from an old system were joined together, some blocks were removed and other ones were created. The simple railway station based on the new proposed blocks is shown in Figure 4.

These blocks are defined as follow:

- **Vjzd** block represents a home signal. This block is a start point of a train path. This train path is built directly from this block. This block can be also the end of a train path. Vjzd block solves a track before the home signal and signalizes whether this track is free or occupied.
- **M** block controls correctness of a train position. In a case, when the train path is divided into three parts, the train coming from left to right must firstly occupy the left segment following by the middle one and at last the right segment. In any other case, an error is signalized.
- **SD** block represents a rail switch and also controls the right position of a train.
- **HQ** block represents an exit signal. This block serves as a start point of the train path and the train path is built to the right from this block. This block can be also the end of the train path.
- **K** block represents a station track and controls the correctness of a train position.

The complex railway station safety device can be generated from these basic blocks. The inputs and the outputs of each block are divided into groups according theirs functionality. There are three types of inputs and three types of outputs.

Inputs are divided as follow:

I name – an input from track
IB name – an input from others blocks
IO name – an input from control device

Outputs are divided as follow:

V name – an output to track
VB name – an output to others blocks
VO name – an output to control device

VI. FSM WITH SELF-CHECKING ARCHITECTURE

The design technique for sequential circuits called MD-architecture has been mentioned in [20, 21, 22]. Authors did not use an error detection code for the outputs, but they have used specific properties of algorithmic state machines (ASM) for achieving the FS property. The MD-architecture is composed of four basic blocks. The outputs of these parts are coded in the 1-out-of-n code in a simple form or in a two-rail form. These parts are designed by such a way that each considered fault manifests itself at one output only, thus all the faults are detected and the architecture is FS. But the proof of ST property has not been presented.

Another technique for SC sequential circuits design is based on an inverter-free design used together with codes that detect unidirectional errors such as Berger code or M-out-of-N code. In [23, 24] the PLA description of a circuit is being modified in order to any fault can cause only a unidirectional error on the output. The modified PLA description is called PLAU. The reduced M-out-of-N code [25] is used, which leads to the FS property.

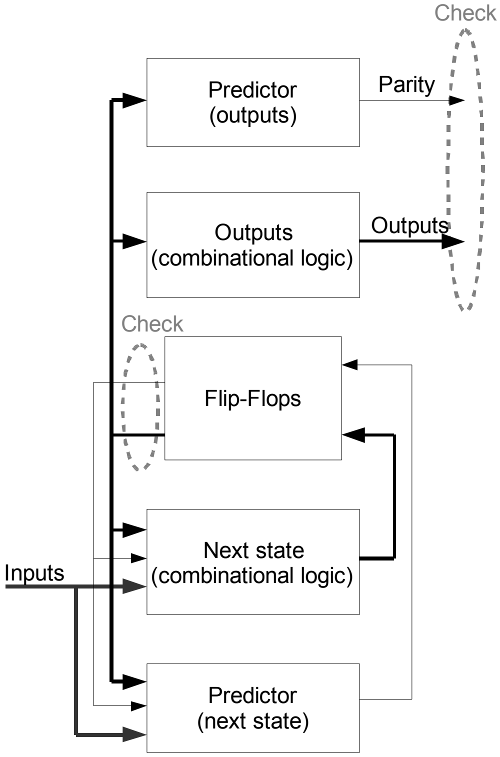

Figure 5. FSM with self-checking architecture

Our approach is based on the basic architecture of MOORE type FSM. There are two blocks of combinational logic. A set of flip-flops is assumed. The current state is stored in flip-flops and we assume its representation only as a data path in our approach. The current state is encoded by the select code (binary or 1-out-of-n code) and forms the code word. The code word is generated by one of combinational logic used to obtain the next state. Both combinational logics are designed as a self-checking ones and use an even parity code to detect a fault. An original combinational logic contains predictor to predict parity nets on outputs from inputs. The proposed architecture of a self-checking FSM is shown in Figure 5. The places where checker checks the correct function of the given combinational logic is highlighted with a dash. The checker of the next state logic is situated behind the flip-flops to keep self-checking property for the whole FSM. This architecture is derived from original rules for connecting of small circuit to compound design (see section 4)

VII. Experimental Results

The synthesis process used to obtain ST and FS parameters and the area overhead is described in the following subsection. All our experiments were performed for two error detection codes: binary and 1-out-of-n. Five basic blocks needed to design any safety device of the railway station are modified to ensure self-checking property. All these five blocks were originally described by VHDL language.

Detailed descriptions of these blocks are presented in Table 1. Here "*FSM*" is the name of the basic safety device block, "*Track I*" and "*Track V*" are the numbers of inputs and outputs nets to drive the track device, "*Other blocks IB*" and "*Other Blocks VB*" are the numbers of inputs and outputs nets which connect individual blocks together, "*Control device IO*" and "*Control device VO*" are the numbers of inputs and outputs nets connected to the operation staff, "*IS*" and "*OS*" are the numbers of sum of inputs and sum of outputs and "*S*" represents a number of states.

TABLE I. Five basic block of railway station safety device

FSM	**Track**		**Other blocks**		**Control device**		**IS**	**OS**	**S**
	I	***V***	***IB***	***VB***	***IO***	***VO***			
HQ	2	6	16	19	2	5	20	30	26
K	1	0	12	4	0	3	13	7	19
M	1	0	12	8	0	3	13	11	22
SD	3	2	44	36	0	3	47	41	37
VJZD	3	6	12	12	2	5	17	23	21

A. *The Overall Synthesis Process*

1. FSM description to be processed has to be described by KISS format. If it is not like this (as in our experiments) the description has to be transformed to KISS format manually.
2. After loading a FSM the internal states coding has done. Then two VHDL top modules are generated: one for combination logic of next state function and one for output function. Then two original circuits and two predictors in PLA format are generated. The last outputs in this step are two files "tst" containing test vectors for both functions. This step is processed with our design tools programmed in C++ language.
3. Now the orthogonal property of all PLA circuits is controlled by ESPRESSO [19]. When this property is not fulfilled (the circuits are not orthogonal) – the KISS description is wrong. It should lead to bad results therefore all PLA circuits have to orthogonal. This step must be performed due to step 1 where manual vhdl2kiss conversion is processed.

4. Now (after the orthogonal property control for all PLA circuits is processed) we use BOOM [18] to minimize the circuits and to translate them to VHDL.

5. Now the Synplicity Synplify synthesis is performed for all generated VHDL circuits. The obtained output is edif format.

6. The obtained edif circuits are connected by generated top modules using Mentor Graphic Leonardo design tool. The predictor and an original circuit are connected for both functions. Two completed edif formats which implement a next state function and output functions of FSM are the final output.

7. The last step is testing, because all data needed for our fault simulator [16] are available.

This process is structurally described in Fig. 6, where all passes are illustrate – through the tools to the testing process.

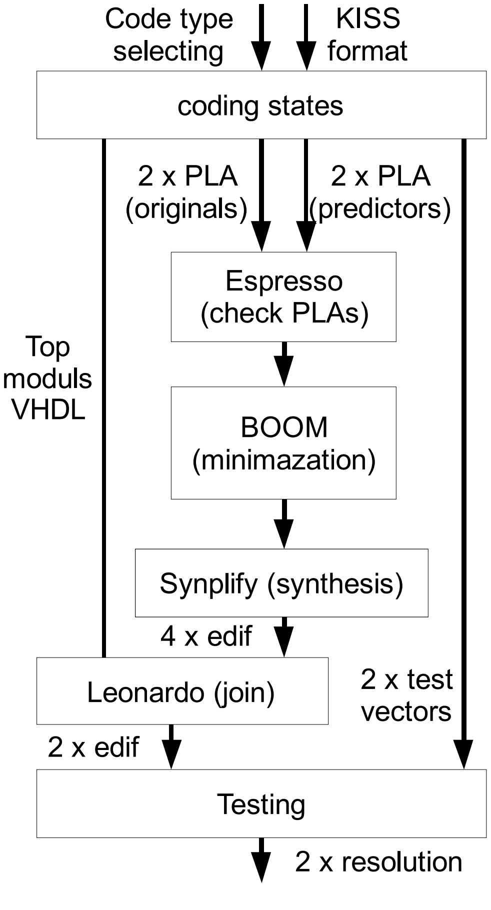

Figure 6. Synthesis process of safety device blocks creation

The described method is usable for a design of any structure of safety devices of railway station constructed from basic blocks with possible reliability parameters increasing the encapsulation.

B. *The Efficiency of the Method*

The results of our experiments were processed individually for the next state and output combinational logic. The results for next state logic are shown in Table 2 for both code (even parity and 1-out-of-n).

The results for the output combinational logic are shown in table 3 and results for the whole FSM are shown in table 4.

Here "*FSM*" is the name of one safety device block, "*Code*" is the name of the used error detection code, "*Orig*." and "*Pred.*" are the numbers of LUTs occupied with original and predictor circuit, "*Over.*" is the overhead of logic used to predict checking outputs, "*Sum*" is the number of tested faults, "*ABCD*" are the groups of faults describing the fault manifestation (described in section 3), "*ST*" indicates how much the circuit satisfies the self-testing property and "*FS*" indicates how much the circuit satisfies fault-security parameter.

These results show that 1-out-of-n code is proper for coding our safety device blocks. The parity predictor for 1-out-of-n code is only constant due to the fact that only one of "1" can by generated on outputs. It means that the predicted output is always "1". The code 1-out-of-n causes that area of original circuit is higher then for binary code. The binary coding has worse fault coverage than 1-out-of-n code. In overall score the area overhead of 1-out-of-n code is almost the same as a binary code but the fault secure parameters (FS) are better. Our results prove, that our methodology is suitable to design the safety device of a railway station but due to the fact that FS is not hundred percent, the MDS architecture must be used to keep SEU detection. Therefore our methodology allows us to create any configuration of railway station safety device.

VIII. CONCLUSIONS

An efficient design methodology for any railway station safety device was presented. The safety device for any configuration of railway station can be built from five basic blocks. Each block is designed as Moore type FSM. The FSM is divided into 2 combinational blocks connected together by data path. Data path for the next state function contains flip-flops in order to store an actual state. Each data path is ensured by the error detection codes. All combination circuits are implemented as self-checking ones. The connections between blocks have to follow TSC architecture. FS parameter shows that FS parameter of each block is more than 80 percent of the whole FSM (next state logic and output logic). If we compare tested codes by their FS parameters and the area overhead, the 1-out-of-n code reaches higher score. Therefore this 1-out-of-n code is suitable for safety device design for a railway station implemented in FPGA. The described method is efficient for a design of any structure of safety devices of the railway station constructed from basic blocks with possible increasing, control and prediction of reliability parameters of the whole design and components, too. It can be stated that our methodology can lead to the design of a safety railway station device. To ensure FS as high as possible (up to 100%), the MDS architecture has to be used.

ACKNOWLEDGMENT

This research has been partially supported by MSMT under research program MSM6840770014, GA102/09/1668, and FI-IM4/149 grant.

REFERENCES

[1] R. Dobiáš, H. Kubátová, “FPGA Based Design of Raiway's Interlocking Equipment”, In Proc. of EUROMICRO Symposium on Digital System Design, Rennes (FR), 31.8. - 3.9. 2004, pp 467-473.

[2] D. Ratter, ”FPGAs on Mars”, www.xilinx.com,Xcell Journal Online, 2004.

[3] Actel Corporation, ”Historic Phoenix Mars Mission Flies Actel RTAX-S Devices”, www.actel.com, 2007.

[4] L. Sterpone, M. Violante, “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.

[5] QuickLogic Corporation, “Single Event Upsets in FPGAs”, 2003, www.quicklogic.com.

[6] M. Bellato, P. Bernardi, D. Bortalato, et al., “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.

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

[8] Actel Corporation, “Single-Event Effects in FPGAs“, http://www.actel.com/documents/FirmErrorPIB.pdf, 2007.

[9] P. Kubalik, R. Dobias, H. Kubatova, “Dependable Design for FPGA based on Duplex System and Reconfiguration”, In Proc. of 9th Euromicro Conference on Digital System Design, Los Alamitos: IEEE Computer Society, 2006, pp. 139-145.

[10] P. Kubalík, H. Kubátová, “Dependable design technique for system-on-chip“, Journal of Systems Architecture. 2008, vol. 2008, no. 54, p. 452-464. ISSN 1383-7621.

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

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

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

[14] P. Kubalík, P. Fišer, and H. Kubátová, “Fault Tolerant System Design Method Based on Self-Checking Circuits”, Proc. 12th International On-Line Testing Symposium 2006 (IOLTS'06), Lake of Como, Italy, July 10-12, 2006.

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

[16] L. Kafka, P. Kubalík, H. Kubátová, and O. Novák, “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.

[17] M. Zatřepálek, “Zabezpečovací zařízení pro železniční stanici založené na FPGA“, Diploma Thesis, Czech Technical University in Prague, Faculty of Electronic Engineering, 2009. (In Czech)

[18] P. Fišer, and J. Hlavička, “BOOM - A Heuristic Boolean Minimizer,” Computers and Informatics, Vol. 22, 2003, No. 1, pp. 19-51.

[19] R.K. Brayton, et al. “Logic Minimization Algorithms for VLSI Synthesis”, Boston, MA, Kluwer Academic Publishers, 1984.

[20] I. Levin, V. Sinelnikov, M. Karpovsky, “Synthesis of ASM-based Self-Checking Controllers,” Euromicro Symposium on Digital Systems Design (DSD'2001), Warsaw, 2001, pp.87-93.

[21] M. Karpovsky, I. Levin, V. Sinelnikov, “New architecture for sequential machines with self-error detection,” International Conference on New Information Technologies (NITe'2000), Minsk, 2000, pp.87-93.

[22] I. Levin, M. Karpovsky, V. Sinelnikov, “Architecture of FPGA-based Concurrent Checking FSM,” The 3rd International Electronic Circuits and Systems Conference (ECS2001), Bratislava, 2001. pp.63-68.

[23] A. Matrosova, S. Ostanin, “Self-Checking FSM Networks Design. The 4th IEEE International On-line Testing Workshop (IOLTW'98), Capri, 1998. pp.162-166.

[24] A. Matrasova, I. Levin, S. Ostanin, “Self-checking Synchronous FSM Network Design with Low Overhead. International Journal of VLSI Design, vol. 11, 2000. pp.47-58.

[25] I. Levin, V. Ostrovsky, S. Ostanin, M. Karpovsky, “Self-checking Sequential Circuits with Self-healing Ability. The 12th Great Lakes Symposium on VLSI (GLSVLSI 2002), New York City, 2002. pp.71-76.

[26] V. Chandra, M.R. Verma, “A Fail-Safe Interlocking System for Railways”, IEEE Design & Test of Computers, 1991, pp. 58-66.

TABLE II. FS FOR NEXT STATE COMBINATIONAL LOGIC

FSM	Code	Orig. [LUT]	Pred. [LUT]	Over	Sum	A	B	C	D	ST	FS
HQ	1-out-of-n	169	1	0,6%	1572	12	1427	0	133	100,0%	91,5%
HQ	binary	156	40	25,6%	1788	48	1510	33	197	98,1%	86,8%
K	1-out-of-n	157	1	0,6%	1428	6	1265	0	157	100,0%	89,0%
K	binary	104	39	37,5%	1310	17	1088	52	153	95,8%	84,1%
M	1-out-of-n	164	1	0,6%	1540	6	1397	6	131	99,6%	91,1%
M	binary	128	46	35,9%	1574	27	1330	37	180	97,5%	86,0%
VJZD	1-out-of-n	142	1	0,7%	1296	6	1177	0	113	100,0%	91,2%
VJZD	binary	115	44	38,3%	1622	5	1328	55	234	96,5%	82,1%
SD	1-out-of-n	311	1	0,3%	2836	638	2006	18	174	99,2%	91,3%
SD	binary	646	112	17,3%	6654	1625	4197	160	672	96,7%	83,5%

TABLE III. FS FOR OUTPUT COMBINATIONAL LOGIC

FSM	Code	Orig. [LUT]	Pred. [LUT]	Over	Sum	A	B	C	D	ST	FS
HQ	1-out-of-n	34	6	17,6%	388	14	260	50	64	84,6%	69,5%
HQ	binary	37	2	5,4%	402	4	273	78	47	75,6%	68,6%
K	1-out-of-n	12	5	41,7%	168	2	121	17	28	88,6%	72,9%
K	binary	9	1	11,1%	110	2	100	2	6	98,1%	92,6%
M	1-out-of-n	17	6	35,3%	240	15	159	32	34	83,4%	70,7%
M	binary	17	2	11,8%	196	4	169	12	11	93,3%	88,0%
VJZD	1-out-of-n	21	4	19,0%	256	2	188	24	42	89,6%	74,0%
VJZD	binary	27	3	11,1%	414	5	368	12	29	97,0%	90,0%
SD	1-out-of-n	38	2	5,3%	414	10	250	62	92	81,9%	61,9%
SD	binary	60	5	8,3%	622	7	422	81	112	84,8%	68,6%

TABLE IV. FS FOR WHOLE STATE MACHINE

FSM	Code	Orig. [LUT]	Pred. [LUT]	Over	Sum	A	B	C	D	ST	FS
HQ	1-out-of-n	203	7	3,4%	1960	26	1687	50	197	97,3%	87,2%
HQ	binary	193	42	21,8%	2190	52	1783	111	244	94,5%	83,4%
K	1-out-of-n	169	6	3,6%	1596	8	1386	17	185	98,9%	87,3%
K	binary	113	40	35,4%	1420	19	1188	54	159	96,0%	84,8%
M	1-out-of-n	181	7	3,9%	1780	21	1556	38	165	97,8%	88,5%
M	binary	145	48	33,1%	1770	31	1499	49	191	97,1%	86,2%
VJZD	1-out-of-n	163	5	3,1%	1552	8	1365	24	155	98,4%	88,4%
VJZD	binary	142	47	33,1%	2036	10	1696	67	263	96,6%	83,7%
SD	1-out-of-n	349	3	0,9%	3250	648	2256	80	266	96,8%	86,7%
SD	binary	706	117	16,6%	7276	1632	4619	241	784	95,5%	81,8%