INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2012, VOL. 58, NO. 4, PP. 419–424
Manuscript received October 25, 2012; revised Decemer, 2012. DOI: 10.2478/v10177-012-0057-8

A New Conception of Safety Logic Microcontroller

Marek Sałamaj

***Abstract*—In this paper a new conception of safety logic microcontroller (BML) is described, together with its physical hardware realization. The unit has various mechanisms which increase its safety and reliability, so that it can satisfy rigorous requirements of safety-critical systems. Thus, the BML unit uses some untypical and innovative technical solutions. The new approach to safety systems development allowed to propose a new conception. The paper describes also physical realization of small multiprocessor BML unit for management of decision-control systems adopted to critical usage.**

***Keywords*—Safety logic microcontroller, critical systems, master-slave architecture, handshake, GALS, conception.**

I. Introduction

SAFETY logic microcontroller (BML) [1] is proposed for usage in critical systems, especially in critical real-time systems [2], [3], but also in other domains. By default it was assumed, that the unit can diagnose and control not only "normal", common technical solutions with much higher precision, but also all kinds of critical real-time systems. Therefore, the conception of a new BML unit was developed for critical real-time systems. Such systems have to satisfy high requirements regarding their structure, functionality and management units (logic controllers and microcontrollers). Requirements for such systems and their management units are in details specified in various standards and technical publications [4].

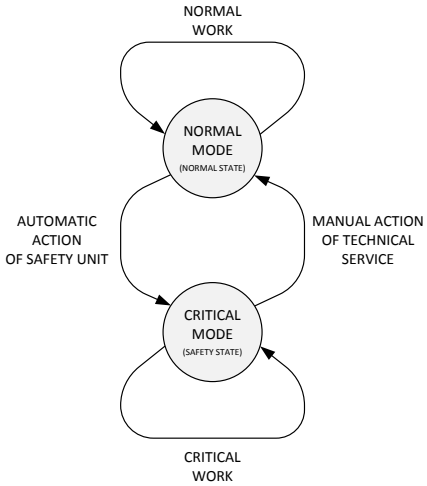

Fig. 1. The modes of the work of a BML unit.

M. Sałamaj is with the Institute of Computer Science and Production Management, Faculty of Mechanical Engineering, University of Zielona Góra, 65-516 Zielona Góra, Poland (e-mail: M.Salamaj@iizp.uz.zgora.pl).

Abnormal functionality of real-time systems may be caused by faulty, incorrect or erroneous operations of internal executing units or the management system itself. Executing units in such systems have clear structure and functioning rules. Control unit, much more complex than managed executing units, may operate incorrect (erroneous). Errors lead to critical mode in real-time systems functionality, which then change into safety states [5]–[7]. Therefore, the new conception of proposed BML unit introduces much more safety and reliability in comparison to other conceptions of this type of units. Construction errors which may appear in safety system development process, can be successfully detected and eliminated using various methods. However, these errors influence critical system functionality less than random errors or errors caused by external aspects from the environment. Thus, safety systems are designed in such a way, that they operate correctly in real time, according to project assumptions. They should detect hardware and software errors automatically on the fly and without any unnecessary delays. Moreover, they should be resistant to external factors. Proposed safety logic microcontroller (BML) meets above requirements.

II. Safety

Considering safety systems design, it should be noted that it has been never possible to propose and develop completely safe system (controller or logic microcontroller). Taking into account the technology, it is not possible to reach an absolutely safe state [5]–[7], and thereby a completely reliable system. In fact, in the real environment it is always possible that something unwanted influences system functionality. Physical realization of safety systems (including the BML unit) with other methods and tools supported by the newest technology and CAD (Computer Aided Design) solutions allows only to develop them precisely. However, it is not possible to protect them against errors in the functionality. Therefore, the BML unit is treated in such a way, that its functionality in real time can be described in any of the two modes: critical mode or normal mode (Fig. 1).

BML normal mode specifies in details all desired functionality schemas and changes of input signals logic states. The schemas are initially defined as project assumptions, and then taken into account in hardware and software of realized control unit. Whenever the BML unit detects a functioning error during self-diagnosis, then it immediately changes from normal mode into critical mode – the so-called safe state [6]. A safe state of the BML unit is a state, where in an extreme situation all output signals are set to the previously defined logic state. This state depends on controlled critical real-time system and does not threat humans life. The BML unit was

designed in such a way, that if necessary, it can initialize internal change from normal mode into critical mode in case of detected:

- construction faults (structure),
- errors connected with data processing (algorithm),
- random errors,
- errors caused by external factors (noise).

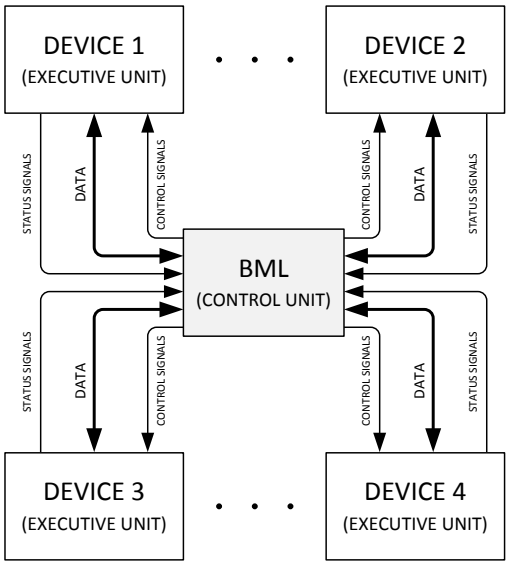

Fig. 2. Example of a simple critical real-time system.

As the result, it is possible to quickly and automatically detect and serve all kinds of above specified errors or faults. Thus, the proposed BML unit conception (in both modes) completely satisfies all requirements for control units, which manage critical real-time systems. By the fact itself, the solution can be considered as a system with much more safety and reliability of its own functionality.

III. BML Conception

Critical real-time systems usually consist of a single safe control unit and at least one controlled executive unit (device). The proposed BML unit can be equivalent to the control unit. BML is adapted to communicate with peripheral (executive) devices, like other units of this type.

In this way, the BML unit has the possibility to precisely diagnose and control internal processes and tasks of supervised devices. It means, that the more objects are controlled by the BML unit, the more complex decision module it becomes. It processes then more data, control and status signals (Fig. 2). Unfortunately, the increase of structure complexity and functionality rules of a safe unit controlling critical system causes, that it becomes more difficult to maintain at a very high level its safety and reliability. Sometimes it is even not possible. Therefore, safe solutions have various diagnostic mechanisms, which should prevent faulty management of critical systems. Thus, the BML unit also has various specialized technical solutions. The priority of these solutions is to analyze in details and verify the correctness of its functionality in real time. It should be noted, that too many safety solutions and mechanisms implemented in safety control systems can lead to situation, when they are considered as units with unpredictable functionality. So, these units cannot be treated as safety systems. Safety systems should have the simplest structure, functioning rules and as simple as possible control mechanisms.

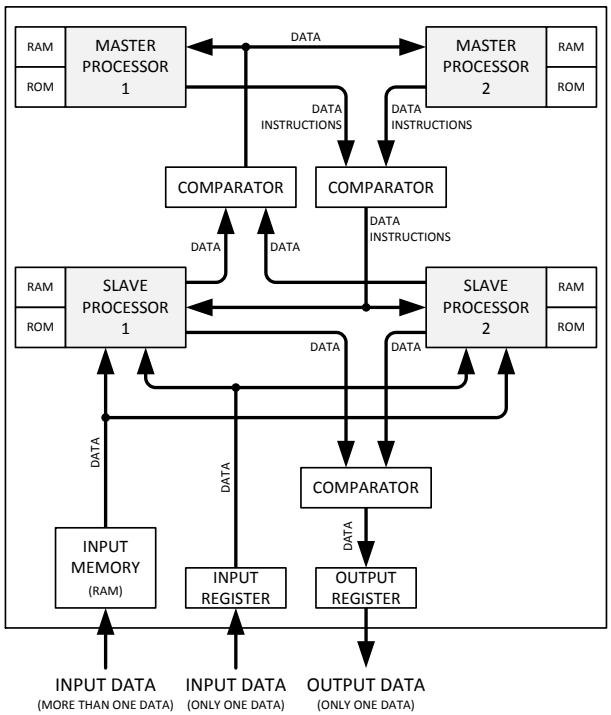

Fig. 3. The simplified construction of Halang-Sniezek controller.

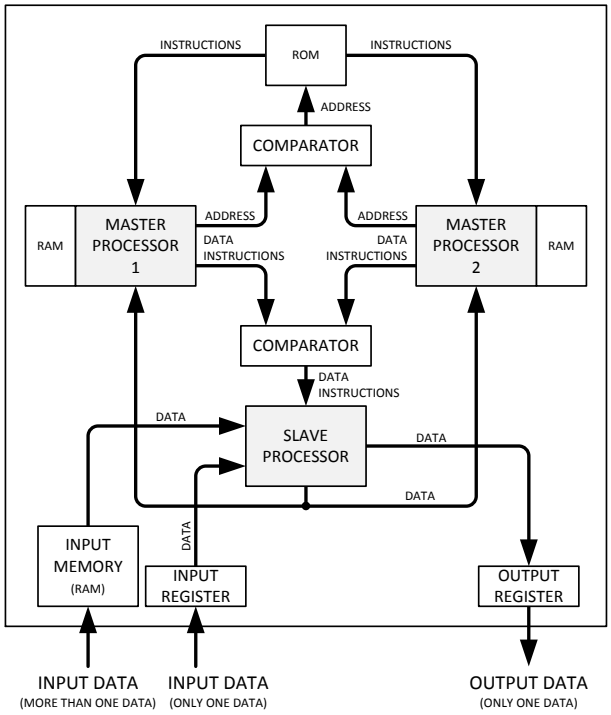

Fig. 4. The simplified construction of BML unit.

Therefore, an optimal choice of the simplest structure and functioning rules taking into account used safety mechanisms is a very important task in safety systems design. An optimal proposition of this type of safety systems is the described BML unit. As a multiprocessor simplified decision-control system, it has various precise and fail-safe mechanisms which ensure system safety. Finally, the proposed solution of control unit was implemented in a reconfigurable device of FPGA (Field Programmable Gate Array) type. It allowed to further test it in a real physical environment.

Safety logic microcontroller (BML) was proposed for fail-safe management of critical real-time systems. Structure of the unit is based on the safety controller construction proposed by Halang-Sniezek [5]–[7] (Fig. 3), which also operates with high computational complexity and is resistant to various types of errors. Unit proposition of this type was initially analyzed in details. Then it was verified against the functionality. As the result, it allowed to propose a completely new construction of BML unit – (Fig. 4). A generalized BML conception in form of function blocks is shown in (Fig. 5). A block schema of the new BML conception has six main blocks. These blocks realize completely different tasks, which have nothing in common with each other, but are complementary in the functionality.

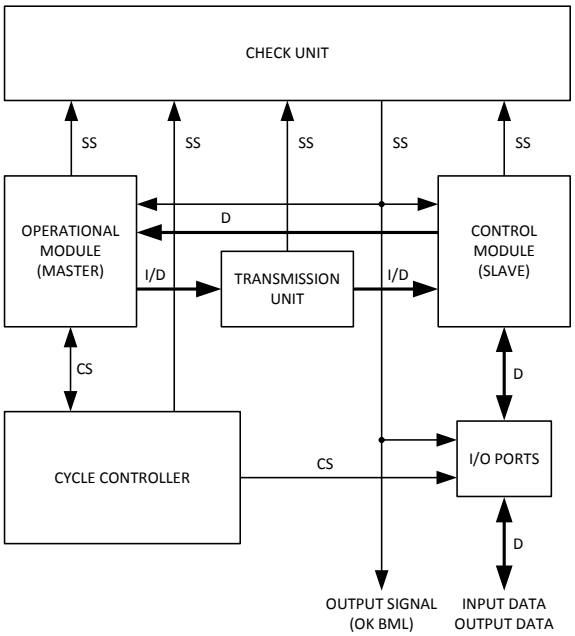

Fig. 5. The block diagram of a BML conception.

Control module MASTER manage and control the whole BML unit, which consists of two concurrently working control processors (MASTER1 and MASTER2), program memory (ROM) and data memory (RAMM1 and RAMM2). Operational module SLAVE in the BML unit (only on MASTER control module request) performs all types of computational operations. It includes only one, but tri-piped mathematical coprocessor SLAVE together with data memory (RAMS). Transmission unit is used only for communication between control and executive modules. Its functioning mechanism is based on safe communication protocols implemented in nets of Handshake type [8], [9].

Input-output ports consist of two general-usage registers (REG_IN and REG_OUT) and data memory (RAMS) shared with operational processor SLAVE. They are used for communication between BML unit and external objects (peripherals). Controller of cycle directly controls the correctness of signal generation, which is responsible for specifying microcontroller (BML) work cycle and input-output ports serving cycle.

Control unit supervises directly the functionality correctness of all dependent (declared before) blocks and function modules in real time. Only if an error a malfunction in dependent units is notified to control block, a specified signal is generated. The signal changes BML work from normal mode into critical mode and, at the same time, it sets the (previously defined) specified logic states of output ports.

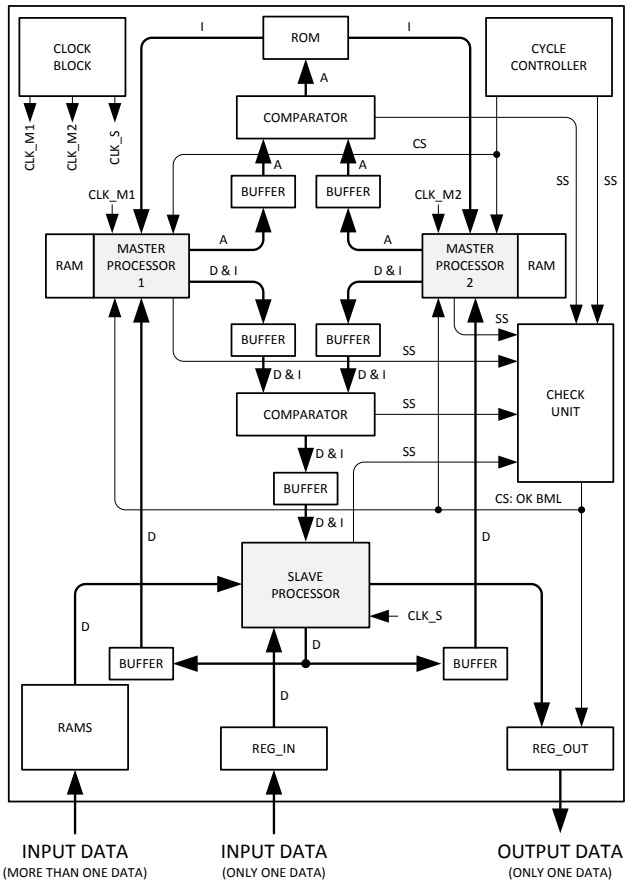

Fig. 6. Details of an BML architecture.

IV. Design Assumptions

At the initial stage of new conception development and safety logic microcontroller (BML) design, some main project assumptions were specified. They allowed to focus on the particular research area. It was inter alia assumed, that:

- BML unit is a single-unit solution implemented completely in a single reconfigurable structure of FPGA

type (low costs, great hardware possibilities, prototyping speed),
- BML unit is a reconfigurable unit, where program can be changed on-line (unlike the Halang-Sniezek solution adapted to particular usage) – a much more flexible solution in comparison to Halang-Sniezek solution,
- BML unit has i.e. only one common power supply (safety is increased at unit conception level, and not on construction level, like in Halang-Sniezek solution),
- considering the functionality, BML unit conception should be similar to Halang-Sniezek unit conception,
- safety level of BML unit with various mechanisms and technical solutions should be possibly high,
- there is a compromise between BML unit complexity and unit methods ensuring safety of its work.

V. Structure and Physical Realization of BML Unit

BML unit was proposed as a three-processor decision module, where three independent and separated processors are connected in MASTER-SLAVE configuration [3], [5], [6] [10], [11] and precisely cooperate. Components of proposed unit (Fig. 6) communicate with each other using the inside implemented safe asynchronous communication protocol of Handshake type.

Control processor MASTER (Fig. 7) and computational processor SLAVE (Fig. 8) are based on Harvard architecture processors. In this case, their structure was initially verified. Then it was reduced to necessary elements. Therefore, the processors could be provided with additional safety solutions. As the result, three locally separated and independent synchronous computational units (GALS technology – Globally Asynchronous Locally Synchronous [12]) were distinguished inside the BML structure by the processors. Usage of additional function blocks was necessary to ensure the correct functionality of these blocks.

Various types of memory – RAMM1, RAMM2, RAMS and ROM can be classified as additional function blocks (used to physically separate data and program), as well as asynchronous comparators and coupling (matching) units with very simple structure and functioning rules. In contrast to used RAM data memory, the ROM program memory was used as a two-ports memory. It is shared between two concurrently working MASTER control processors and has two independent address spaces.

Both address spaces have exactly the same BML control program, but are individually served by different control processors MASTER1 or MASTER2. RAMS data memory together with input register REG_IN and output register REG_OUT as input-output ports was adapted to communication with external executing devices. Control unit verifies on the fly all received signals with information about correct/incorrect functionality of chosen components of the BML structure. Whenever an error inside any component is reported, then control unit immediately changes the BML work mode into safety mode, where previously defined logic states values of all output ports are set. Cycle controller generates

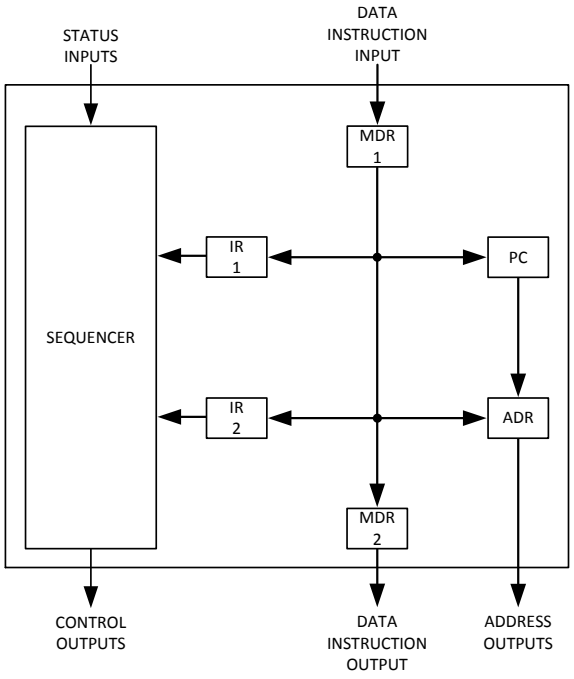

Fig. 7. Architecture of a MASTER processor.

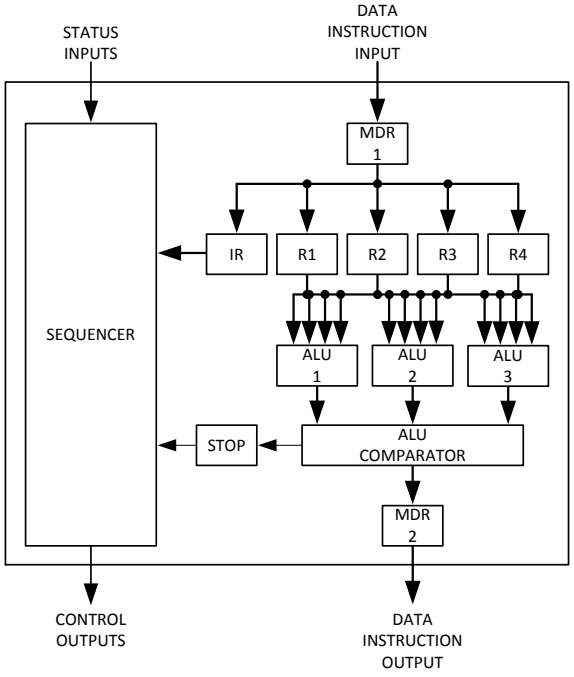

Fig. 8. Architecture of a SLAVE processor.

in this case only signals, which initialize the safety system. Additionally it informs control unit about critical delays in decision system functionality.

Various simple, but also specialized in a particular domain technical solutions were used inside the BML unit. It was a motivation for further researches on its structure and functioning rules in the real physical environment. Therefore, a new conception of BML unit was successfully completely

implemented in a reconfigurable structure of FPGA type, namely in VIRTEX-II PRO device of XC2VP30 type with VIRTEX-II PRO device of 94V-00523 type of the XILINX company. The whole synthesis process of proposed solution (BML) was performed successfully without any problems, taking into account both available optimization types. Partial results of synthesis process are presented in Tab. I and Tab. II.

TABLE I
RESULTS OF SYNTHESIS OF SELECTED BML COMPONENTS – OPTIMIZATION: AREA

BLOCK	F-F	Latch	4 input LUTs	F_{CLK}	Gates
SEQ_M1	36	–	513	184,3 MHz	3563
SEQ_M2	170	–	1007	35,6 MHz	7519
SEQ_S	40	–	429	300,3 MHz	3141
MASTER1	140	–	592	46,4 MHz	5803
MASTER2	244	–	1329	26,6 MHz	11202
SLAVE	204	–	3182	30,3 MHz	23727
BML	939	1255	6437	26,6 MHz	58403

TABLE II
RESULTS OF SYNTHESIS OF SELECTED BML COMPONENTS – OPTIMIZATION: SPEED

BLOCK	F-F	Latch	4 input LUTs	F_{CLK}	Gates
SEQ_M1	36	–	539	424,4 MHz	3674
SEQ_M2	186	–	1126	115,5 MHz	8487
SEQ_S	40	–	426	492,5 MHz	3001
MASTER1	182	–	841	83,2 MHz	7468
MASTER2	289	–	1643	59,1 MHz	13419
SLAVE	299	–	4354	57,4 MHz	31940
BML	1035	1255	7782	56,9 MHz	67610

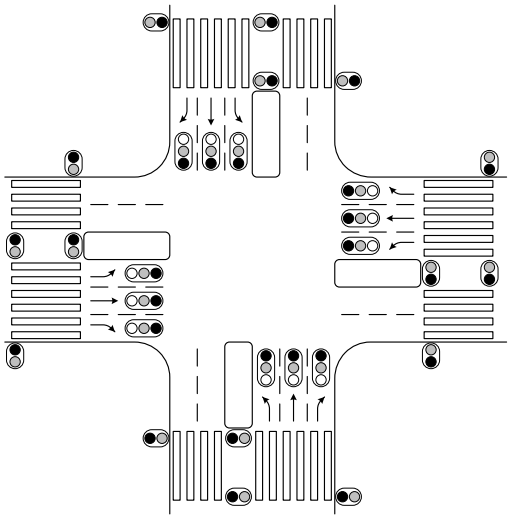

Fig. 9. A model of crossroads.

VI. TEST OF PROTOTYPE UNIT

Safety Logic Microcontroller unit has been designed and physically made to control various critical real-time systems. The implementation of this unit in a reconfigurable structure of FPGA (device: VIRTEX-II PRO FPGA type XC2VP30 with VIRTEX-II PRO type 94V-0 0523 XILINX) allowed to test his work in the physical environment. In this case, a critical system used in the explorations was represented by a model of crossroads. This mock-up was prepared on the pattern of the intersection of two multi-lane roads with traffic lights, which the view is shown in (Fig. 9). In this proposed system (in the model), traffic light of the intersection was managed by BML unit, which main task was only controlled the turning on and off lights in proper sequence. Arrangement of elements of traffic lights on the layout crossroads is shown in (Fig. 9). In this particular application, the control unit (BML) did not collect and verified any information from the system, but only controlled it. By reason of it, the control system of the intersection realized only specified control algorithm and analyzed the correctness of the work of safety unit. The main and most important task of BML unit was to manage the traffic lights so that traffic of cars on the model of intersection was smooth and grade-separated. Therefore, in the work of the considered system are specified four major drive cycles, which details are shown in (Fig. 10).

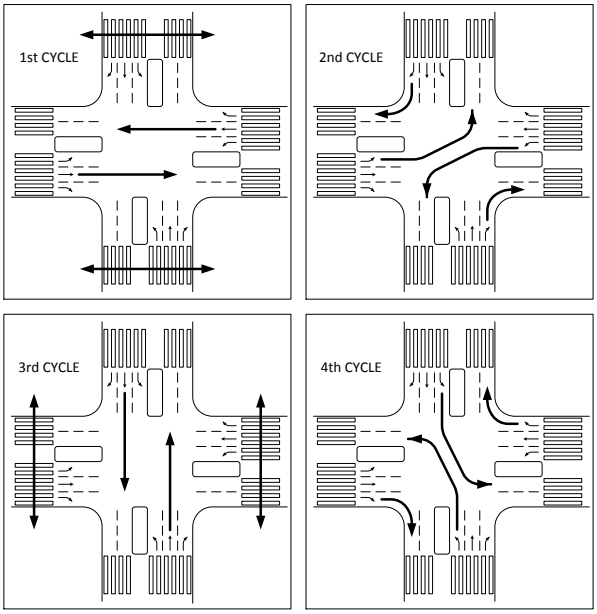

Fig. 10. Four major drive cycles on the crossroads.

According to the concept of the BML, this unit can operate in one of two possible states of the work: in the normal state and in the critical condition. In the normal state, the control unit (which was used to manage critical system of traffic lights) can operate only in two modes: day and night mode. During the day mode, the BML unit turns on and off a three-color traffic lights at the crossroads to obtain street traffic in accordance with the cycles as shown in (Fig. 10). Cycles occur one after another. However in the night mode, the same microcontroller turns on and off only yellow blinking lights. The automatic switch between a day and nigh mode can only appears, while signal from both sensor or clock programmed by technical service was occurred. In this case, the day and the night mode determines the status of the normal operation

of the BML unit. Accidentally, the critical state, in the same system coincides with the night mode in the normal state of the BML work. Depending on the time of day, the safety microcontroller switches the state of his work between day and night mode, according to the algorithm shown in (Fig. 11). However, every little mistake, an anomaly, defect or failure of the BML work immediately switches its state from the normal state to the critical state. In this case of critical state, blinking yellow lights warning about occurring system failure and street traffic is determined based on traffic signs.

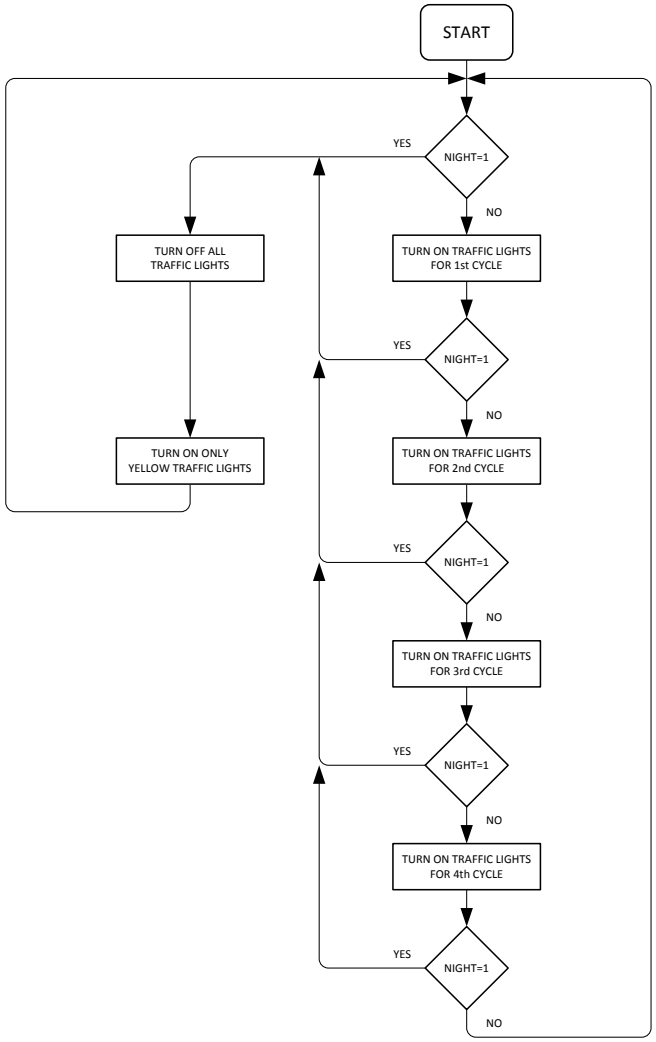

Fig. 11. Algorithm of work of the BML unit.

VII. Conclusion

A new BML unit conception was successfully physically implemented in a reconfigurable structure of FPGA type. Then, the functionality of realized control unit was successfully tested, this time in a specified critical usage. In this way it has been proved, that it is possible to develop at a low cost the designed BML conception. It is possible with all simplifications as well as with various types of specialized technical solutions, safety mechanisms and technics. Such solutions have great hardware possibilities and fast prototyping phase. However, some designers still think, that FPGA devices are not appropriate to realize safety solutions. It is caused by the fact, that certification and design process of safety solutions should undergo reverse engineering. Unfortunately, while designing solutions with FPGA devices, this condition cannot be satisfied. Each programming of reconfigurable device creates (each time) a completely new unique connection structure, i.e. of internal logic blocks. Although the new BML conception was also implemented in FPGA and does not satisfy the condition mentioned before, it certainly may be treated as a safety system, especially considering the conception level (not construction level) and taking into account used solutions and technologies.

Additionally, there is a belief that technical innovations should not be used in safety solutions design. Old, proved and as simple as possible methods and solutions seem to be better suited. This assumption may be misleading, as in fact reconfigurable devices are not taken into account. This electronics domain is a dynamically developing technic area, which only recently gives us the possibility to pack great amounts (quantitatively) of the smallest and simplest already known technical solutions in one project. Necessity of complex projects development will enforce changes. Thus, reconfigurable devices with time will eliminate from the market obsolete and ineffective solutions. Thereby, reconfigurable solutions (like the BML unit) will be adapted to use in critical systems.

References

[1] M. Adamski and M. Salamaj, “Programowalny sterownik logiczny,” Polish Patent P388 721, Aug. 03, 2009.
[2] M. Colnaric, D. Verber, and W. A. Halang, *Distributed Embedded Control Systems: Improving Dependability with Coherent Design*, 1st ed., ser. Advances In Industrial Control. Springer Publishing Company, Incorporated, 2008.
[3] M. Wegrzyn and A. Bukowiec, “Design of safety critical logic controller using devices integrated microprocessors with FPGA,” in *Proceedings of SPIE’05*, vol. 5775, 2005, pp. 377–384.
[4] J. Rinaldi, “Control IEC 61131-3 - The Fast Guide to IEC 61131-3 Open Control Standard & Software,” *Real Time Automation*, 2010.
[5] W. A. Halang and M. Sniezek, “A safe programmable electronic system,” *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 58, no. 3, pp. 423–434, 2010.
[6] M. Sniezek and W. A. Halang, *Bezpieczny programowalny mikrosterownik logiczny*. Rzeszow, Polska: Oficyna wydawnicza Politechniki Rzeszowskiej, 1998, in Polish.
[7] M. Sniezek and W. A. Halang, “Electronic system for safety tasks programmed with logic diagrams and flow charts uses two computers for processing function block references and converting data flow between function blocks and signal sequences specified by flow charts,” Patent DE19 861 281, 2008.
[8] A. Julius, “GALS – global asynchronous local synchronous circuits,” *Humboldt-Universität (HU-Berlin)*, 2004.
[9] M. Krstic and E. Grass, “New GALS technique for datapath architectures,” *Springer-Verlag Berlin Heidelberg*, vol. 2799, pp. 161–170, 2003.
[10] M. Sniezek and J. Stackelberg, “A fail safe programmable logic controller,” *Annual Reviews in Control*, vol. 27, pp. 63–72, 2003.
[11] S. Zaba, “Analiza czasowa cyfrowych interfejsow mikroprocesorowych z architektura master-slave,” *Pomiary Automatyka Robotyka*, vol. 4, pp. 13–17, 2005, in Polish.
[12] S. Smith, “An asynchronous GALS interface with applications,” in *Proceedings of the 2004 IEEE Workshop on Microelectronics and Electron Devices (WMED’04)*, Boise, Idaho, United States, 2004, pp. 41–44.