www.czasopisma.pan.pl PAN www.journals.pan.pl
INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2013, VOL. 59, NO. 4, PP. 363–368
Manuscript received November 11, 2013; revised December, 2013. DOI: 10.2478/eletel-2013-0044

Conception of a Control Unit for Critical Systems

Marek Sałamaj

***Abstract*—The article regards critical systems precisely controlled by safety units. A new idea of safety logic microcontroller is proposed. It is built on the basis of the simplest mechanisms and technical solutions. This approach allowed to obtain a complex decision-making and control microsystem, in which the applied mechanisms and solutions increased its reliability. As a result, the proposed control unit proved to be a universal solution, which can be used in any critical system.**

***Keywords*—Safety logic microcontroller, critical systems, master-slave architecture, FPGA, conception**

I. Introduction

THE current standard of living, as well as the industry requires constant progress and automation of various technical solutions that surround us. The continuous process of their upgrading and extension is a reason for creation of ever newer and much more specialized control units, which are able to control and manage them. Such type of technical solutions combined with managing them control units form the specialized technical devices, which generally are identified as real-time systems [1]–[3].

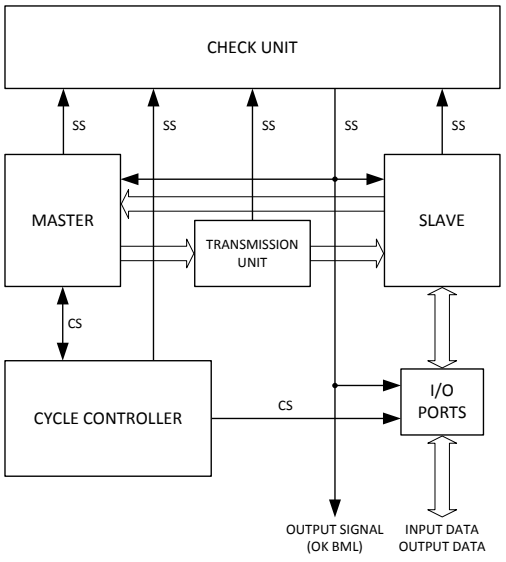

Fig. 1. The block diagram of a BML conception.

Real-time systems are used in various domains where they perform some specified actions in an automated way. An example of such system can be a very simple system used in a guarded paid parking lot. Various types of equipment are here connected together – a specific group of sensors (photocells, timers), executing units (entry and exit gate) and the most important control unit (logic controller, microcontroller, computer). In this case, the conditional payment of a fee causes the control unit to open the entrance gate to the parking lot, otherwise the entrance gate remains closed. Therefore, any damage of the control unit managing gates functionality can result in a situation when a gate is permanently closed or open. Then, the vehicles are allowed to enter the parking lot without any restrictions, or the same parking is blocked and cannot make profits until the fault has been noticed, identified and removed by the technical support team. Response time of the technical support team is not limited, and the possible delay only leads to financial losses, which have to be calculated into in the functioning of the system (parking lot).

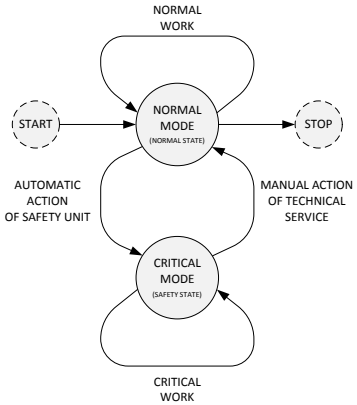

Fig. 2. The modes of the work of a BML unit.

Currently, real-time systems are used in almost every domain that is surrounding us, but also in critical applications. They are being used more and more in military and civilian critical real-time systems, which often are referred to as critical systems [3]–[7]. The word critical means here, that all included elements should definitely meet all the rigorous requirements for such systems. In addition, this type of elements should be reliable (faultless), because any error in their functioning could lead to the risk of human life, environmental pollution or to some financial losses. Therefore, critical systems must be supported (controlled and diagnosed) by specialized safety control units.

II. Safety Control Unit

Because of the fact that in each critical system all sensors and actuators (objects) are usually managed by a single control unit, the unit becomes responsible for monitoring (diagnosing) the correctness of their operation. Therefore, it is necessary to create newer control and diagnose unit solutions that

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

would be able to correctly manage any objects regardless of the operating conditions (environment). Depending on the complexity and functionality of critical systems, such units can be represented by specialized processors, logic controllers, microcontrollers or computers, provided that they are safe [6], [8]–[11]. In this case, the safety of control unit has to be understood as the trouble-free functionality (operating). It can be obtained by applying to a control unit various mechanisms and technical solutions significantly increasing the reliability.

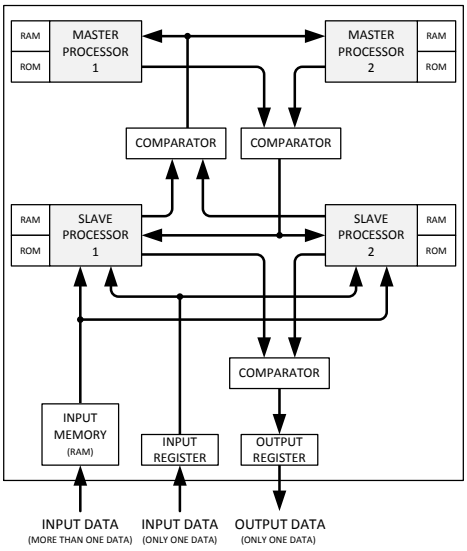

Fig. 3. Conception of a Halang-Śnieżek controller.

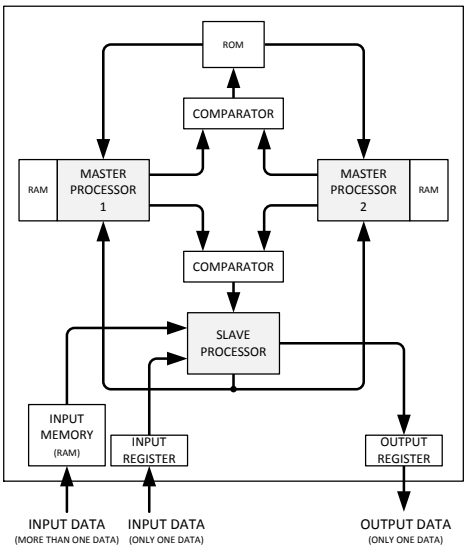

Fig. 4. Conception of a Adamski-Sałamaj controller.

An example of a safety control unit is the proposed and implemented Safety Logic Microcontroller (BML) [6], [7], which can be used as a safety system for critical applications [12] (depicted in Fig. 1). This system (BML) can be adapted to manage any critical system, managing i.e. traffic lights or long-range rockets. A correctly working BML unit should control the subordinate objects according to the design assumptions regarding the control unit itself as well as corresponding elements of critical system. If any fault (error) is detected in the BML control unit (or in the system), it should make the functionality of the whole system safe.

Reliable work of the system after a failure enforces switching the control unit into a safety mode (Fig. 2) and activating the appropriate alarm signalization. The signalization should force the user to increase the patience, pay special attention or take some specific actions. As a result, the stuck and turned into safety mode unit (the whole critical system) is waiting for technical support team that can diagnose and remove the cause of the failure. Very fast error detection and its identification in the BML system is obtained by automating the switching of operation mode from working (normal) mode into safety (critical) mode (Fig. 2). This type of automation in the BML system at the current level of technological complexity is crucial and necessary, because any delay in reaction to a detected fault, or even the lack of it, can lead to serious consequences. Then, any delay in safety mode functionality could lead to a threat to human life, environmental pollution or to specific financial losses. For this reason, human interaction is completely eliminated from the process of critical system management to make the system operate much more efficiently and faster responding to occurring events. Therefore it is necessary to use in control units of critical systems the ever-increasing amount of various mechanisms and technical solutions, that will enhance their computing capabilities in terms of identification and interpretation of errors and failures. In the proposed BML system the above mentioned characteristics have been successfully significantly improved.

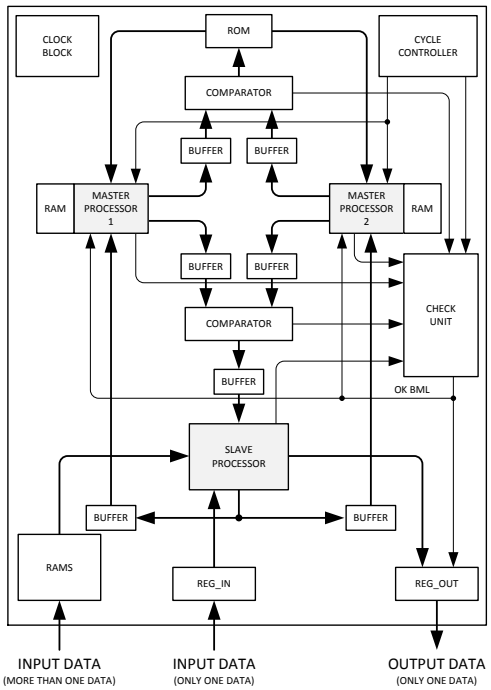

Fig. 5. Details of an BML architecture.

III. Safety Logic Microcontroller

It is very important that technical solutions used in various domains are continuously extended, modernized and improved. Therefore, the researches regarding a new concept of Safety Logic Microcontroller (BML) architecture [6], [7] are still going on in the Faculty of Mechanical Engineering at the University of Zielona Góra. The studies were initiated by Halang-Śnieżek (Fig. 3) in the safety logic controller approach [8]–[10] and then extended by Adamski-Sałamaj (Fig. 4) [6], [7].

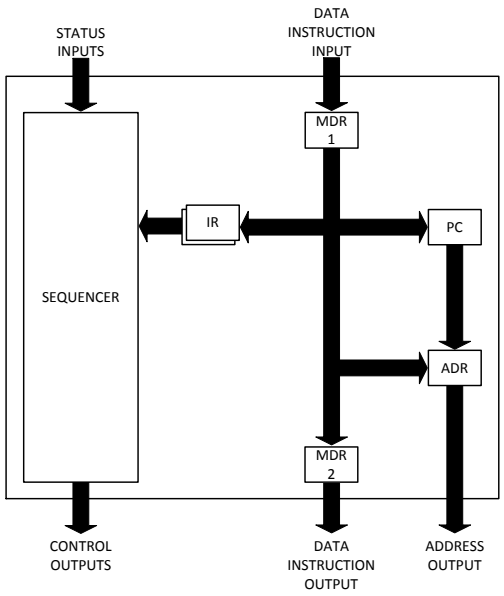

Fig. 6. Architecture of a MASTER processor.

The BML system has been designed so far to be used in the critical real-time systems [9], [13]–[16] and to control them with much higher precision and reliability in comparison to standard solutions. The BML system is designed as a 32-bit control unit, where the three-processor MASTER-SLAVE architecture (Fig. 5, Fig. 6, Fig. 7) [8], [9], [11], [14], [15] with all necessary functional modules is fully implemented in a single reprogrammable structure of FPGA (Field Programmable Gate Array) [17] (depicted in Fig. 5). On the other hand, commonly used logic controllers and microcontrollers are implemented on the basis on the processor placed in the vicinity of various functional modules (depicted in Fig. 8).

Additionally, the realization of the BML unit in a reprogrammable FPGA structure (as opposed to standard logic controllers) helped to implement in its architecture some specific solutions and mechanisms that finally quite significantly increased the safety and reliability level. The goal during the BML implementation was to make the proposed system meet rigorous requirements for units used in critical systems [18], in terms of construction and functioning principles. As a result, a new conception of the BML unit was proposed and implemented, prepared on the basis of a number of unusual, yet innovative technical solutions.

When considering the safety systems design it should be noticed, that it has never been and it will not ever be possible to propose and implement a completely safe control unit (including the BML). A state of absolute safety [8]–[10], and the more a state of extreme and total reliability of the system, is impossible to achieve in terms of implementation technology, as in the real environment the system functionality may be anytime undesirably disturbed. On the other hand, the implementation of safety systems (including the BML) with another methods and tools supported by the latest technologies and CAD-solutions, allows only for their precise realization, but not for protection against errors or faults in the functionality. Therefore, by the BML functioning principles it is taken into account that it can work in any of two possible modes of operation: in safety mode or in normal mode [6], [7], [9] (Fig. 2).

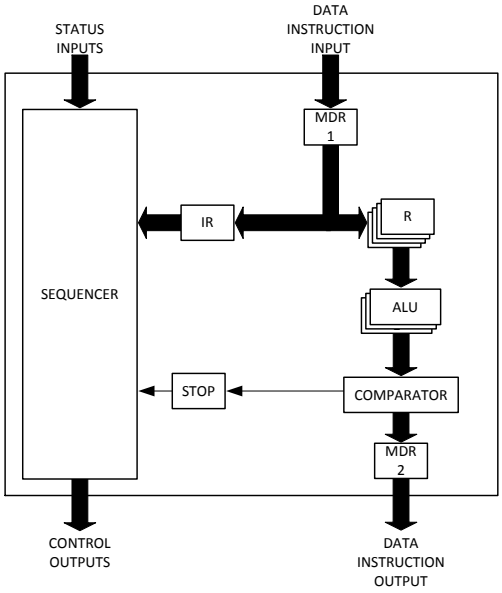

Fig. 7. Architecture of a SLAVE processor.

The BML normal mode defines in details all the desired patterns of unit behavior after changes of logical states that can occur on its inputs. These patterns are initially defined in the early design phase (when assumptions are made), and then reflected in the hardware and software of an implemented control unit. Whenever the BML system detects an error in the functionality while automatically self-diagnosing, it is immediately switched from the normal mode into the critical mode – the so-called safety state [6], [7], [9], [10], [14]. The safety state of a device is a state where in an extreme situation all its outputs (in this case BML's outputs) have assigned some strictly specified logical states dependent on the real-time critical system which manage the device. Switching working mode of the BML from the critical mode (the safety state) into the normal mode requires in turn an outside intervention of the technical support team which will diagnose the reason why the control of the system was stopped.

The BML unit was designed in such a way, that if necessary, it can initialize an 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).

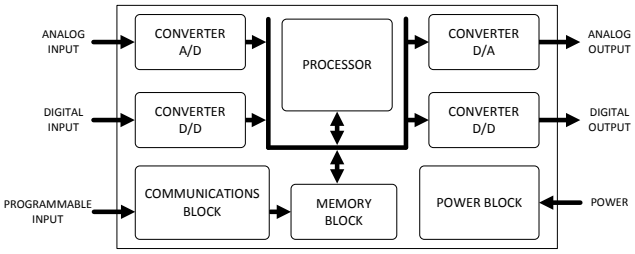

Fig. 8. The block diagram of a logical controller.

In this case, construction faults that may occur in the process of safety system design can be effectively detected and eliminated using various methods. Therefore, these faults do not threaten the critical system functionality so much as random errors or errors caused by external factors in the place of its (later) usage. This is why safety systems are designed in such a way that they operate correctly in real-time according to the design assumptions. Moreover, they should on-the-fly automatically and without any unnecessary delays detect hardware and software faults or errors and be resistant to external factors. In this way the experimental model of the BML unit was designed and implemented.

Implementation of measuring and recording devices which use in the architecture reconfigurable structures of the FPGA-type (as it is in the BML system) is much more expensive than universal solutions using programmable logic controllers. In spite of high costs of realization of such devices, using the FPGAs offers incomparably much more technical possibilities of implemented hardware. The reason for this is the rapid progress of computer science and electronics in the field of reconfigurable structures, which allowed in recent years to implement in a single FPGA not only single-processor solutions, but even complex decision-making and control systems. Thus, short development time and implementation simplicity of measuring and recording devices using FPGAs is only limited by the engineer's imagination and the need to adjust the signal levels of the control unit (current and voltage) to external extension modules and sensors.

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-Śnieżek solution adapted to particular usage) – a much more flexible solution in comparison to Halang-Śnieżek 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-Śnieżek solution),
- considering the functionality, BML unit conception should be similar to Halang-Śnieżek 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 [8], [9], [11], [14], [15] and precisely cooperate. Components of proposed unit (Fig. 5) communicate with each other using the inside implemented safe asynchronous communication protocol of Handshake type.

Control processor MASTER (Fig. 6) and computational processor SLAVE (Fig. 7) 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 [13], [16], [19]) were distinguished inside the BML structure by the processors. Usage of additional function blocks was necessary to ensure the correct functionality of these blocks.

TABLE I
The Results of the Synthesis of Selected Components of BML – 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

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

TABLE II
THE RESULTS OF THE SYNTHESIS OF SELECTED COMPONENTS OF BML – 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

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

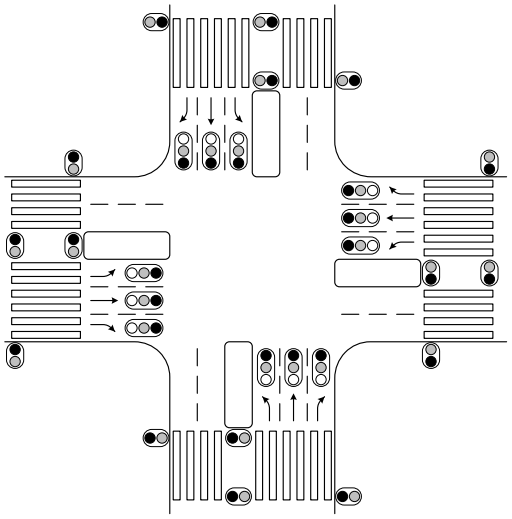

Fig. 9. A model of crossroads.

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.

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 [17]) 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).

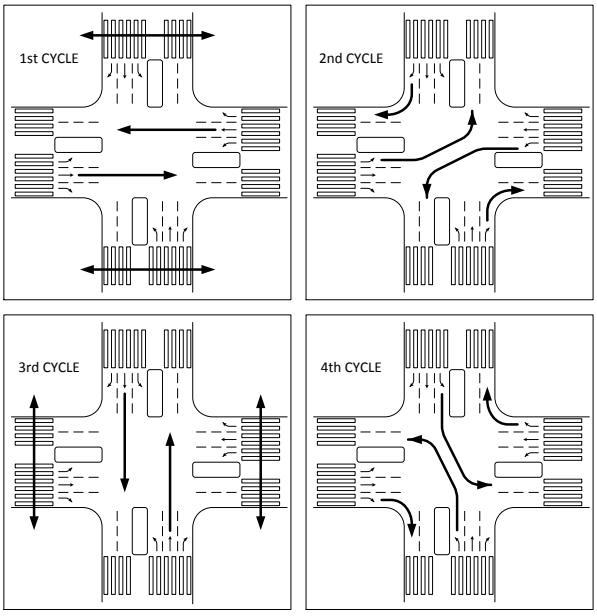

Fig. 10. Four major drive cycles on the crossroads.

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 threecolor 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 appear, 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.

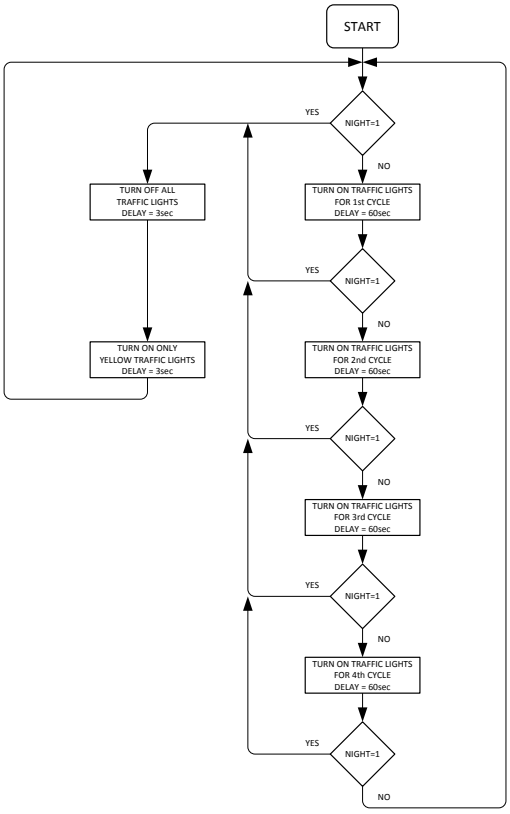

Fig. 11. Algorithm of work of the BML unit.

VII. Conclusion

The proposed Safety Logic Microcontroller (BML) is certainly not the only one system that can be used to manage critical systems. However, a wide range of various implemented solutions and mechanisms allowed the control unit to become much more efficient and function much more precisely than previously used systems. Furthermore, the proposed control unit is quite well protected against the occurrence of various types of faults and errors, and thus it is much more safe (fault-free). Increased safety level in the BML was obtained thanks to the detailed optimization and significant simplification of the complexity of technical solutions implemented in its structure. The process of simplification of the BML system involved mainly its whole architecture, as well as all the architectures of specialized function processors. In this way, the complexity of the final BML version has become so simple that its work (functionality) can be easily controlled. According to design assumptions, the proposed system was supposed to be and it is the most versatile (standard) solution adapted for use in a variety of critical systems in which BML's functionality depends entirely on the program execution.

In summary, safety units for managing of critical systems must be continuously extended and innovated. It can be confirmed by the fact that various fields of industry and everyday life are dynamically changing. Therefore, a very rapid technological progress of our civilization requires that decision-making and control units should be created that would be able (in the future) to manage the more complex and well-developed critical systems.

References

[1] W. Halang and K. Sacha, Eds., *Real-Time Systems*. London, Great Britain: Implementation of Industrial Computerised Process Automation, World Scientific, 1992.
[2] K. Sacha, "Fault analysis using petri nets," *Proc. IEEE Real-time Embedded Systems Workshop*, pp. 130–133, 2001.
[3] D. V. M. Colnaric and W. Halang, Eds., *Distributed Embedded Control Systems: Improving Dependability with Coherent Design*. London, Great Britain: Springer-Verlag, 2008.
[4] M. W. M. Adamski and A. Węgrzyn, Eds., *Safety reconfigurable logic controllers design*. Warszawa, Polska: Wydawnictwo Komunikacji i Łączności, 2007.
[5] J. Biernat, *Architektura komputerów*, Wrocław, Polska, 2005, wyd. 4 rozszerzone.
[6] M. Adamski and M. Sałamaj, "Programowalny sterownik logiczny," Patent P388 721, Aug., 2009.
[7] M. Sałamaj, "Bezpieczne sterowniki logiczne w automatyzacji procesów produkcyjnych," *Zeszyty Naukowe Uniwersytetu Zielonogórskiego, Mechanika*, no. 132, pp. 55–60, 2006.
[8] W. Halang and M. Śnieżek, "A safe programmable electronic system," *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 58, no. 3, pp. 423–434, 2010.
[9] M. Śnieżek and W. Halang, *Bezpieczny programowalny sterownik logiczny*. Rzeszów, Polska: Oficyna wydawnicza Politechniki Rzeszowskiej, 1998.
[10] ——, "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.
[11] M. Węgrzyn 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.
[12] K. Sacha, "Verification and implementation of dependable controllers," in *Proceedings of the 2008 Third International Conference on Dependability of Computer Systems DepCoSRELCOMEX, IEEE Computer Society*, 2008, pp. 143–151.
[13] M. Krstic and E. Grass, "New gals technique for datapath architectures," *Springer-Verlag Berlin Heidelberg*, vol. 2799, pp. 161–170, 2003.
[14] M. Śnieżek and J. Stackelberg, "A fail safe programmable logic controller," *Annual Reviews in Control, PERGAMON, Elsevier Ltd*, vol. 27, pp. 63–72, 2003.
[15] S. Zaba, "Analiza czasowa cyfrowych interfejsów mikroprocesorowych z architekturą master-slave," *Pomiary Automatyka Robotyka*, vol. 4, pp. 13–17, 2005.
[16] 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.
[17] "Xilinx - all programmable technologies and devices," 2013. [Online]. Available: www.xilinx.com
[18] R. T. Automation, "Control iec 61131-3," in *The Fast Guide to IEC 61131-3 Open Control Standard & Software*, 2010.
[19] A. Julius, "Gals - global asynchronous local synchronous circuits," *Humboldt-Universitat (HU-Berlin)*, 2004.