A Novel Comparator with Hamming Code Correction for Safety Programmable Logic Controller

Bai Zhanyuan, Xu Aidong, Liu Mingzhe and Song Yan
Industry Informatics Laboratory
Shenyang Institute of Automation, Chinese Academy of Sciences
Shenyang, China
breezyon@sia.cn

Abstract- **Safety Instrumented Systems (SIS) are automatic systems designed for the purpose of taking an action to avoid an accident or minimize its consequences. Safety Instrumented System relies on many devices that must work as designed at a specific point in a hazard scenario to stop propagation of the hazardous event. Safety Programmable Logic Controller (PLC) plays a more and more important role in the SIS. This paper introduces a quad architecture Safety PLC, which provides complete integration within a single control architecture, where safety and standard control functions reside and work together. A novel comparator is designed with a FPGA chip, which performs hamming code correction and data bus comparison. The safety integrity level and the evaluation approach are introduced.**

I. INTRODUCTION

Safety Instrumented Systems (SIS) are used to implement Safety Instrumented Functions (SIF) [1].A Safety Instrumented System is composed of any combination of sensors, transmitters, valves, valve positioners and logic solvers and final control elements for the purpose of taking a process to a safe state when predetermined conditions are violated [2].

Safety PLC is a core part in a Safety Instrumented System; such computers will be applied in various fields which require simultaneously both: availability and maximal safety. They are applied where human lives need to be protected and/or safed, either in material handling, energy production/distribution, in the medical field or in future industrial power plants in space.

As a helpful way to improve the system's safety, hamming code correction technique is used in Safety PLC architecture introduced below [3].

II. 2OO4- ARCHITECTURE SAFETY PLC

A novel comparator was designed for a 2oo4-system Safety PLC, please refer to Figure 1 for additional information. The 2oo4- architecture contains four independent channels, provides four (4) processor-two per channel, and remedies problems associated with dual processor architectures, as regards the dangerous undetected failure of one of the two (dual) processors. Both pairs of active processors operate synchronously with the same user program. A hardware comparator and a separate fail-safe watchdog monitor the operation of each pair of processors to diagnose and resolve anomalies.

In the architecture, a novel comparator designed in FPGA connects to the data bus, and compares the real-time data from two different data bus, to detect errors. In the FPGA chip two SRAM controllers and Hamming Code Correction Module are designed by VHDL language to perform data comparison and data correction.

The multiplexer is used to connect the different data bus together, and make the data transmit to other data bus become possible.

The dual port rams (DPR) are the communication part for the two same boards, this architecture can switch the 2oo4-system to other channels seamlessly.

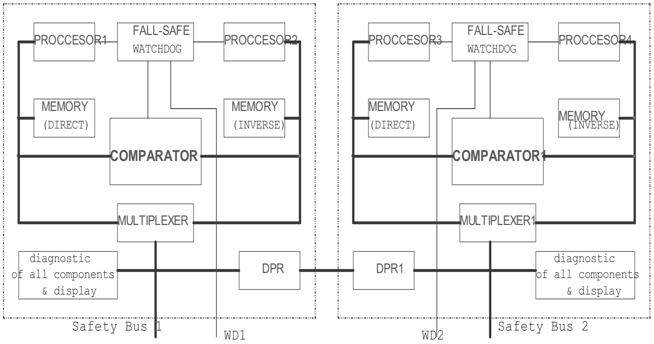


Figure 1. A 2oo4-architecture for Safety PLC

The four channels are connected one with another. In order to trigger the safety function at least two of the four channels must work correctly. Even if two failures in two different channels occur the system can be transmitted into the safe state.

A dangerous breakdown of the system is generated if three of the four channels have dangerous failures themselves. Figure 4 shows a reliability block diagram of a 2oo4-architecture. Each single channel contains of an input circle, a safe processing unit and two serial output circles.

As such architecture, the safety PLC can operate at the SIL3 level on either one or both channels, for an unrestricted period of time. It achieves a significant increase in both safety and availability which exceeds that provided by TMR architectures by a factor of three. In addition, it has significantly less susceptibility to common cause failure because of the absolute separation, isolation and operation of the redundant channels [4].

III. SAFETY PLC RELIABILITY ANALYSIS

A. Safety integrity levels

IEC 61508 "Functional safety of electrical/electronic/ programmable electronic safety-related systems" defines the requirements for programmable electronic systems used in the safety related parts of controls systems. This standard is driving the direction for future safety PLC developments.

Safety Integrity Level (SIL) is defined as a relative level of risk-reduction provided by a safety function, or to specify a target level of risk reduction. In simple terms, SIL is a measurement of performance required for a Safety Instrumented Function (SIF).The requirements for a given SIL are not consistent among all of the functional safety standards.. Within the European Functional Safety standards four SILs are defined, with SIL 4 being the most dependable and SIL 1 being the least [5].

The standard details the requirements necessary to achieve each safety integrity level. These requirements are more rigorous at higher levels of safety integrity in order to achieve the required lower likelihood of dangerous failure. In order to have measurable parameters it was defined the widely used parameters "mean time to failure" (MTTF) and "probability of failure on demand" (PFD). The PFD characterizes the quality of a faultless system. The different requirements of the altered safety integrity levels (SIL) are dependent on PFD-value, the smaller the value the better the safety of the system.

In low demand operation and continuous operation mode, the PFD_{avg} value and the corresponding safety integrity levels is shown in TABLE I:

TABLE I
SAFETY INTEGRITY LEVEL FOR SAFETY FUNCTIONS

Safety Integrity Level	Low Demand Operation PFD_{avg}	Continuous/High Demand PFD_{avg}
1	$10^{-2} \sim 10^{-1}$	$10^{-6} \sim 10^{-5}$
2	$10^{-3} \sim 10^{-2}$	$10^{-7} \sim 10^{-6}$
3	$10^{-4} \sim 10^{-3}$	$10^{-8} \sim 10^{-7}$
4	$10^{-5} \sim 10^{-4}$	$10^{-9} \sim 10^{-8}$

Combing all elements of a system in a safety architecture the system can be classed with a defined safety level, safety integrity level (SIL).Table 1 shows the various classifications of safety systems. The norm IEC /EN 61508 defines two different criterions for the classification of the safety systems into the individual safety levels. [6]

The International Electrotechnical Commission's (IEC) standard IEC /EN 61508, defines SIL using requirements grouped into two broad categories: hardware safety integrity and systematic safety integrity. A device or system must meet the requirements for both categories to achieve a given SIL.

The SIL requirements for hardware safety integrity are based on a probabilistic analysis of the device. To achieve a given SIL, the device must meet targets for the maximum probability of dangerous failure and a minimum Safe Failure Fraction. The concept of 'dangerous failure' must be rigorously defined for the system in question, normally in the form of requirement constraints whose integrity is verified throughout system development [7]. The actual targets required vary depending on the likelihood of a demand, the complexity of the device(s), and types of redundancy used.

B. *PFD_{avg} for a 2oo4-System*

The probability of failure for the 2oo4-system for normal failures is:

$$PFD_{avg,\,normal}(T) = (\lambda_D)^3 \cdot T^3 \qquad (1)$$

The *PFDavg* value for common cause failures as

$$PFD_{avg,\,\beta} = \frac{\beta \cdot \lambda_{DU}}{2}(T_1 + MTTR) + \frac{\beta_D \cdot \lambda_{DD}}{2} \cdot MTTR \qquad (2)$$

In the equation (2), β is a weight factor,T1 means the proof time interval, MTTR means the mean time to repair.

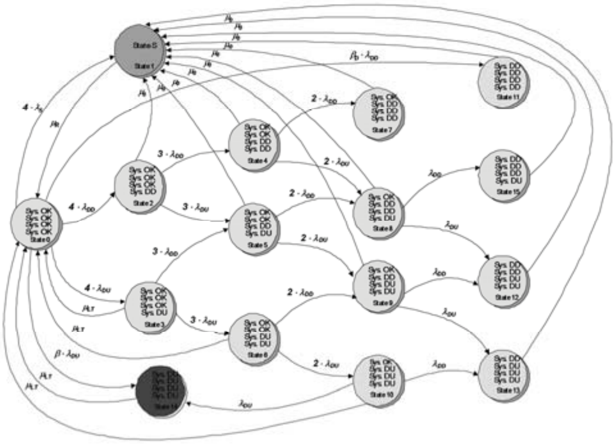
The PFDavg equation of a 2oo4-system taking into account the normal failures, equation (2), and the common cause failure, equation (1), is therefore:

$$PFD_{avg} = (\lambda_D)^3 \cdot T^3 + \frac{\beta \cdot \lambda_{DU}}{2}(T_1 + MTTR) + \frac{\beta_D \cdot \lambda_{DD}}{2} \cdot MTTR \quad (3)$$

The probability of a common cause failure is the same in a 1oo2-, 2oo3- and in a 2oo4-system. If the probability of a normal failure in a 2oo4-system is compared with the probability of a 2oo3-system, then the probability in a 2oo4-system is several dimensions smaller than in a 2oo3-system [8].

C. *Markov-model of a 2oo4-architecture*

Markov analysis is a powerful and flexible technique to assess the reliability measurements of safety instrumented systems. Figure 2 is a Markov model for reliability assessment of 2oo4-architecture safety instrumented systems. Many safety related factors, such as failure modes, self-diagnostic, restorations, common cause and voting, are included in Markov models [9].



State S
State 1
Sys. OK Sys. OK Sys. OK Sys. OK State 0
Sys. OK Sys. OK Sys. OK Sys. DD State 2
Sys. OK Sys. OK Sys. OK Sys. DU State 3
Sys. OK Sys. OK Sys. DD Sys. DD State 4
Sys. OK Sys. OK Sys. DD Sys. DU State 5
Sys. OK Sys. OK Sys. DU Sys. DU State 6
Sys. OK Sys. DD Sys. DD Sys. DD State 7
Sys. OK Sys. DD Sys. DD Sys. DU State 8
Sys. OK Sys. DD Sys. DU Sys. DU State 9
Sys. OK Sys. DU Sys. DU Sys. DU State 10
Sys. DD Sys. DD Sys. DD Sys. DD State 11
Sys. DD Sys. DD Sys. DU Sys. DU State 12
Sys. DD Sys. DU Sys. DU Sys. DU State 13
Sys. DD Sys. DD Sys. DU Sys. DU State 14
Sys. DD Sys. DD Sys. DD Sys. DU State 15
$4 \cdot \lambda_S$
μ_R
μ_S
$\beta_D \cdot \lambda_{DD}$
$4 \cdot \lambda_{DD}$
$3 \cdot \lambda_{DD}$
$3 \cdot \lambda_{DU}$
$2 \cdot \lambda_{DD}$
$2 \cdot \lambda_{DU}$
$4 \cdot \lambda_{DU}$
λ_{DD}
λ_{DU}
μ_{LT}
$\beta \cdot \lambda_{DU}$

Figure 2. 2oo4-system Markov chain model

Basically is the Markov-model of a 2oo4-"Single-Board System" accomplished with conventional calculation methods [10]. The single transitions are shown in figure 2.

IV. COMPARATOR DESIGN

A. *Comparator internal architecture*

In the architecture, comparator connects to the data bus, and compares the real-time data from two different data buses, to implement data validation and error-correcting. The novel comparator detects errors by hamming code and decodes the data in the memory. This operation can deduce data errors, correct 1 bit error. So this architecture of the comparator can improve the safety integrity level [11].

As in figure 3 showed, in the comparator there are two SRAM controllers, thorough these two SRAM controllers, the comparator reads in the data from the two direct and inverse data buses.

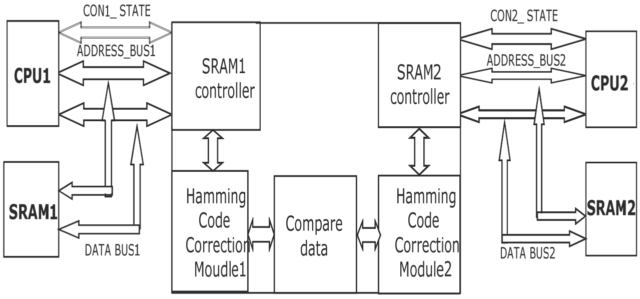


Figure 3. comparator for 2oo4-architecture schematic

The comparator uses SRAM controllers to control the data bus between the CPU and SRAM, When data is write out from CPU, comparator will code the data by hamming code, and when data is read out from SRAM, the comparator will perform a decode operation ,this operation can detect two bit errors and correct one bit error.

B. Hamming code correction

Error Correction Coding has been a crucial part of data transmission or storage. In high-reliability applications.In telecommunication and data storage, when data is transmitted from one location to another there is always the possibility that an error may occur. When digital data is stored in a memory, it is crucial to have a mechanism that can detect and correct a certain number of errors. Error Correction Codes (ECC) encode data in such a way that a decoder can identify and correct certain errors in the data. Hamming code is linear block code, A Hamming code is an error-correcting code, Hamming codes can detect single and double-bit errors, and correct single-bit errors as well. As a helpful way to improve the system’s safety, hamming code correction technique is used in Safety PLC architecture introduced below.

In this design, a FPGA is suitable to implement the hamming code correction as the complex algorithm and cell density, a FPGA has more than millions of cells to achieve system demands [12].

V. Conclusion

The 2oo4 Safety PLC provides high reliability for Safety Instrumented System by its advanced architecture. The more safe 2oo4-architecture will be established within high safety class computers in future. The comparator is a core part to connect the two channels which are running simultaneously, and compares the data from the two dependent channels. The novel comparator can also perform code correction when errors are found by Hamming Code Correction Module. And comparator is also an important part for judging fail mode to switch channel. To verify whether SCS had conformed SIL3, the technique known as FMEDA (failure modes effects and diagnostic analysis) was used and analyzed the failure rate, failure mode, and effects caused by the failure of a component on all of the constituting components [13].

References

[1] IEC 61508. Functional Safety of Electrical /Electronic / Programmable Electronic Safety-Related Systems. International Electrotechnical Commission, Geneva, Switzerland.

[2] Mohamed Sallak, Christophe Simon, and Jean-Francois Aubry, “A Fuzzy Probabilistic Approach for Determining Safety Integrity Level,” *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, vol. 16, NO. 1, February 2008, pp.239-248.

[3] Zhao Jianwu Shi Yibing Li Yanjun, “Software Implementation of a Novel Approach to Improving Burst Errors Correction Capability of Hamming Code,” The Eighth International Conference on Electronic Measurement and Instruments, ICEMI’2007, 2007, pp.499-503.

[4] Laihua Fang, Zongzhi Wu, Lijun Wei, etc. “Design and Development of Safety Instrumented System,” *Proceedings of the IEEE International Conference on Automation and Logistics*, China, 2008, pp.2685-2690.

[5] Shuo Li, Zheying Li, Li Luo, etc. “Processor Architecture Design for Operation Information Processing Based on Safety Model of Instruments,” ICSP’04 Proceedings, pp.2687-2690.

[6] PROF. DR.-ING. HABIL. JOSEF BÖRCSÖK, “ Modern 2oo4-processing architecture for safety systems,”.

[7] Torres-Echeverrı́a A, Martorell S, Thompson H, “Design optimization of a safety-instrumented system based on RAMS+C addressing IEC 61508 requirements and diverse redundancy,” Reliability Engineering and System Safety, vol. 94(2)2009, pp.162-179.

[8] Julia V. Bukowski, “A Unified Model for Evaluating the Safety Integrity Level,” *IEEE*, 2008,

[9] Wolfgang Velten-Philipp, Dr. Michel Houtermans, “The Effect of Diagnosic and Periodic Testing on Safety Related Systems,”

[10] Evzudin Ugljesa, Josef Börcsök, “Evaluation of sophisticated hardware architectures for safety applications,” *IEEE*, 2009.

[11] Josef Börcsök, Ali Hayek, Muhammad Umar, “Implementation of a 1oo2-RISC-Architecture on FPGA for Safety Systems,” IEEE, 2008, pp.1046-1051.

[12] Graham Cormode, Mayur Datar, Piotr Indyk, etc. “Comparing Data Streams Using Hamming Norms,” *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, vol. 15, 2003, pp.529-540.

[13] Ryotaro Shishiba1, “Implementation of a Safety Instrumented System,” SICE Annual Conference 2007, Kagawa University, Japan, September, 2007, pp.2493-2496.