International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:6, No:10, 2012

Study the Effect of Soft Errors on FlexRay-Based Automotive Systems

Yung-Yuan Chen, and Kuen-Long Leu

***Abstract*—**FlexRay, as a communication protocol for automotive control systems, is developed to fulfill the increasing demand on the electronic control units for implementing systems with higher safety and more comfort. In this work, we study the impact of radiation-induced soft errors on FlexRay-based steer-by-wire system. We injected the soft errors into general purpose register set of FlexRay nodes to identify the most critical registers, the failure modes of the steer-by-wire system, and measure the probability distribution of failure modes when an error occurs in the register file.

***Keywords*—**Soft errors, FlexRay, fault injection, steer-by-wirer

I. Introduction

REPLACING the traditional mechanical and hydraulic control systems by electronic control systems is the inevitable tendency in the automotive industry worldwide [1]. Such replacements have the benefits of saving the cost/energy and improving the performance as well as safety. However, electronic control systems have higher probability of incurring fatal interferences such as electromagnetic interference (EMI) or radiation-induced error than mechanical and hydraulic systems. Therefore, the safety and robustness issues must be addressed during the development of safety-critical electronic automotive systems.

It is well known that the rate of soft errors caused by single event upset (SEU) increases rapidly while the chip fabrication enters the very deep submicron technology [2-4]. Since system-on-chip (*SoC*) becomes prevalent in the intelligent automotive applications, which require a stringent dependability while the systems are in operation. When *SoCs* are applied to safety-critical applications, fault-robust designs with the dependability validation are required to guarantee that the developed *SoCs* are able to comply with the safety requirements defined by the international norms, such as IEC 61508 [5, 6] or ISO 26262 [7].

For the complicated IP-based *SoCs* or embedded systems, it is unpractical and not cost-effective to protect the entire *SoC* or system. Analyzing the vulnerability of *SoCs* or systems can help designers not only invest limited resources on the most crucial region but also understand the gain derived from the investment. The failure mode and effects analysis (FMEA) [8] and fault tree analysis (FTA) [9] are two effective approaches that are used to validate the robustness/safety of the *SoCs* or systems and to identify the critical components and major failure modes for protection if the measured robustness/safety cannot meet the system requirements. The results of FMEA and FTA can be exploited to help us develop a feasible and cost-effective risk-reduction process.

FlexRay – as the next generation of in-vehicle network standard – provides not only high bandwidth but also fault-tolerant features. A number of practical fault-tolerant mechanisms (FTMS) have been introduced in the FlexRay protocol specification [10]. According to that, system developers can utilize the provided FTMS to enhance the communication robustness of the FlexRay systems. FlexRay protocol specification [10] focuses mainly on the reliable data communication. As we know, a FlexRay cluster is a communication system of multiple nodes, which consists of at most two channels and each node in the cluster may be connected to either or both of the channels. So, it is clear that in addition to the reliable data communication, the reliable node operation plays another crucial role for a FlexRay system to comply with the safety requirements. Validating the robustness of a node in the FlexRay system becomes imperative in the robustness validation process.

In this study, we are going to investigate the effect of soft errors on the nodes of FlexRay-based systems. We employ the fault injection method to inject the faults/errors into the nodes to examine the behaviors of the system, to measure the robustness of the nodes and system, and to locate the vulnerability of the node. We use a FlexRay-based steer-by-wire system to demonstrate the robustness validation process, where the soft errors were injected into the register file of a selected FlexRay node.

The remaining paper is organized as follows. In Section 2, we propose a robustness validation and fault-tolerant design process. A FlexRay-based steer-by-wire system is presented in the following section. The experimental platform and robustness validation results are described and discussed in Section 4. The conclusions appear in Section 5.

II. Robustness Validation and Fault-Tolerant Design Process

We propose a robustness validation and fault-tolerant design process as shown in Fig. 1 to develop the safety-critical electronic systems. The process contains three phases described as follows.

Phase 1 (fault hypotheses): this phase is to identify the potential interferences and develop the fault injection strategy to emulate the interference-induced errors.

Yung-Yuan Chen is with the Department of Electrical Engineering, National Taipei University, New Taipei City, Taiwan (phone: 886-2-86741111; fax: 886-2-26736500; e-mail: chenyy@mail.ntpu.edu.tw).

Kuen-Long Leu is with Department of Electrical Engineering, National Central University, Jhongli City, Taoyuan County, Taiwan (phone: 886-3-4227151; fax: 886-3-4255830; e-mail:: 945401025@cc.ncu.edu.tw).

Phase 2 (FMEA): this phase is to perform the fault injection campaigns based on the Phase 1 fault hypotheses. Throughout the injection campaigns, we can identify the failure modes of the system, which are caused by the faults/errors injected into the system while the system is in operation. The probability distribution of failure modes can be derived from the fault injection campaigns. The risk-priority number (RPN) [8] is then calculated for the components inside the electronic system. A component's *RPN* aims to rate the risk of the consequence caused by component's failure. RPN can be used to locate the critical components to be protected. The robustness of the system is computed based on the adopted robustness criterion, such as safety integrity level (SIL) defined in the IEC 61508 [5]. If the robustness of the system meets the safety requirement, the system passes the validation; else the robustness/safety is not adequate, so Phase 3 is activated to enhance the system robustness.

Phase 3 (fault-tolerant design): This phase is to develop a feasible risk-reduction approach by fault-tolerant design to improve the robustness of the critical components identified in Phase 2. The enhanced version then goes to Phase 2.

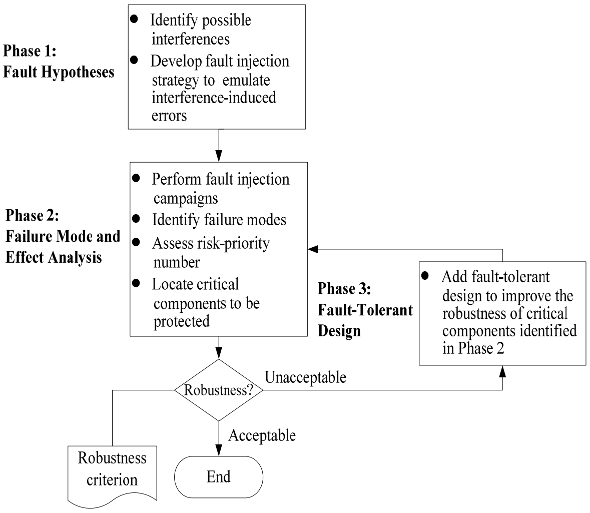

Phase 1:
Fault Hypotheses
Identify possible interferences
Develop fault injection strategy to emulate interference-induced errors
Phase 2:
Failure Mode and Effect Analysis
Perform fault injection campaigns
Identify failure modes
Assess risk-priority number
Locate critical components to be protected
Phase 3:
Fault-Tolerant Design
Add fault-tolerant design to improve the robustness of critical components identified in Phase 2
Robustness?
Unacceptable
Acceptable
Robustness criterion
End

Fig. 1 Robustness validation and fault-tolerant design process

III. FlexRay-Based Steer-by-Wire System

A. Soft Errors in FlexRay Nodes

FlexRay network – the next generation automotive drive-by-wire communication system – provides a new communication infrastructure for safety-critical automotive applications, such as steer-by-wire and brake-by-wire. Fig. 2 illustrates a node architecture that consists of a host processor, a communication controller and bus drivers [10]. A node can be connected to either or both of the channels and a FlexRay cluster can be configured as a bus topology, star topology, or hybrid combinations of bus and star topologies.

As stated before, radiation-induced soft errors could cause a serious dependability problem for *SoCs*, electronic control units, and nodes used in the safety-critical applications. The soft errors may happen in the flip-flop, register file, memory system and combinational logic. The reliable node operation plays an important role for FlexRay-based systems to achieve the stringent safety requirement, such as SIL 4 in IEC 61508. As a result, we need to adopt the robustness validation and fault-tolerant design process as shown in Fig. 1 in the design of FlexRay node to guarantee its robustness. A FlexRay-based steer-by-wire system was constructed to demonstrate the robustness validation (Phases 1 and 2 in Fig. 1) of a node.

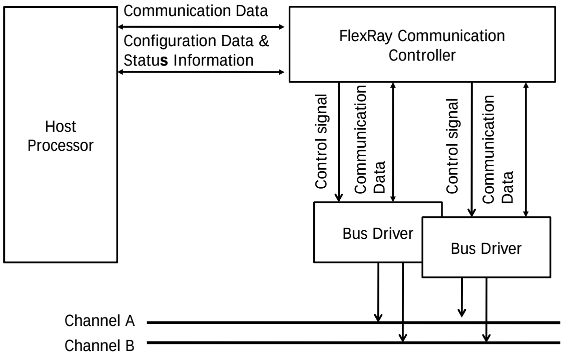


Fig. 2 A node architecture

B. Steer-by-wire System

Fig. 3 shows the architecture of simplified steer-by-wire system, where two nodes are clustered by bus topology to implement the steering control law. We basically utilize the Ackermann steering geometry as displayed in Fig. 4 to decide the steering angles of front wheels [11]. The expression (1) can be used to calculate the turning angles of front wheels according to the angle of steering wheel. The concept of active steering can also be implemented in steering control to increase the vehicle maneuverability, stability and safety. The steering ratio may be varying with speed data. For example, at low/high speed, we could use low/high steering ratios to increase the maneuverability/stability of the wheel control.

$$\cot\alpha = \cot\beta + \frac{DO - CO}{L} = \cot\beta + \frac{B}{L} \qquad (1)$$

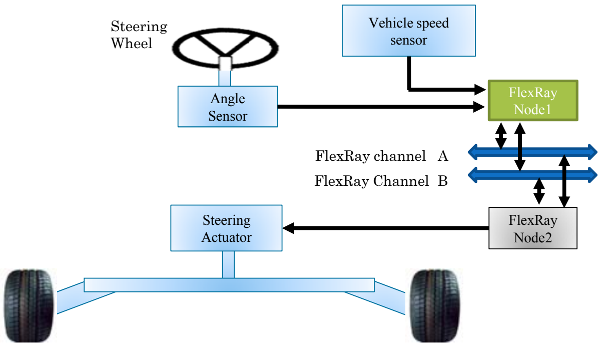
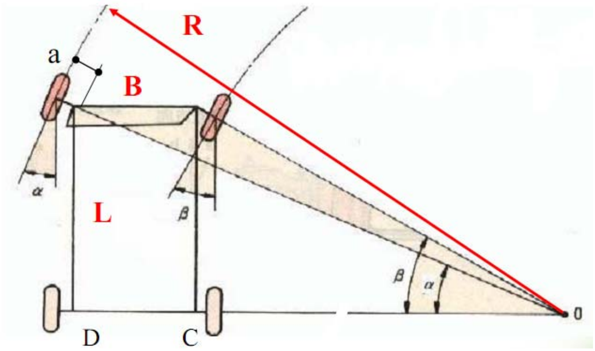


Fig. 3 Simplified steer-by-wire architecture



β, α: turning angle of inner and outer front wheels, respectively;
B: width of front axle;
L: distance between front and rear axles.

Fig. 4 Ackermann steering geometry

IV. Experimental Results

A. Experimental Platform

According to the depiction of Section III.B, an experimental FlexRay-based steer-by-wire system was created. The platform with task assignments in Node1 and Node2 is illustrated in Fig. 5, where the nodes are implemented by TTTech Universal FlexRay Control Unit with Infineon Tricore TC1796 host CPU. An experiment contains one thousand and eight hundred input patterns from sensors. A pattern is 5-byte long and comprises the data of steering wheel, brake and throttle. The sampling rate of input patterns is 5*ms*.

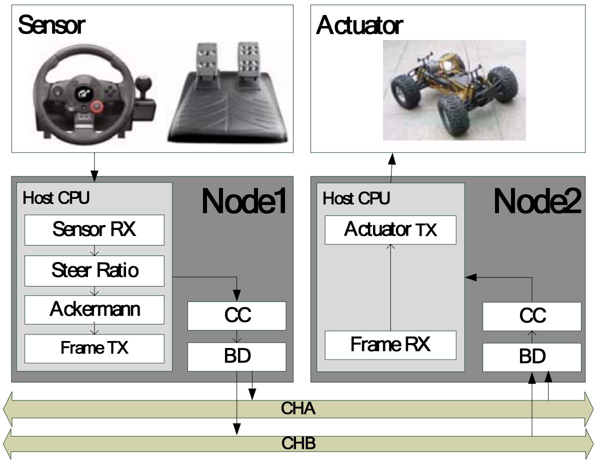


Fig. 5 Steer-by-wire platform, where CC: communication controller & BD: bus driver

B. Robustness Validation

We injected the soft errors into register file within the Node1 host CPU to identify the failure modes of the steer-by-wire system and assess the probability distribution of failure modes when an error occurs in the register file. The experimental data can be employed to locate the vulnerability of register file as well. Besides that, we adopt the safety criterion termed as SIL defined in IEC 61508 to demonstrate the robustness validation. If the robustness of the system fails to meet the safety requirement, the fault-tolerant design will be utilized to improve the system robustness. At this phase, the vulnerability analysis of register file provides valuable information for Phase 3.

C. Results and Discussion

The potential failure modes can be identified from the fault injection campaigns. We have conducted one hundred and ninety fault injection campaigns for experimental platform as illustrated in Fig. 5. In the experiments, nineteen registers named as 'D0' ~ 'D15', i.e. sixteen 32-bit data registers, and 'PC', two 32-bit address registers 'A2' and 'A15' are selected as the fault injection targets. We performed ten fault injection campaigns for each register and each injection campaign injected a single bit error into that register. Time instant of fault injection is randomly chosen in the time range of the fault-free experiment. The possible failure modes classified from the fault injection process could be silent data corruption (SDC), correct data/incorrect time (CD/IT), and deadlock (DL) as depicted in Fig. 6. No Effect (NE) in Fig. 6 means that a fault/error happening in a component has no impact on the system operation at all. The failure probability (FP) is equal to one minus NE.

We note that the bit errors occurring in the register set won't cause damage to the system if one of the following situations occurs:

- Situation 1: The benchmark never reads the affected registers after the bit errors happen.
- Situation 2: The first access to the affected registers after the occurrence of bit errors is the 'write' action.

Otherwise, the bit errors could cause damage to the system. Clearly, if the first access to the affected registers after the occurrence of bit errors is the 'read' action, the bit errors will be propagated and could finally lead to the failures of system operation. So, whether the bit errors will become fatal or not, it all depends on the occurring time of bit errors, the locations of affected registers, and the benchmark's register read/write access patterns after the occurrence of bit errors.

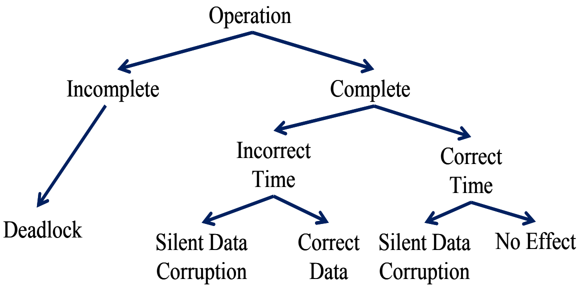


Fig. 6 System behaviors resulting from fault injection campaigns

Table I gives the experimental results of fault injection campaigns. According to the data, we can rank the vulnerability of registers in register set, which prioritized the registers to be protected. Consequently, the precious resources can be invested in the most SEU-critical registers so as to effectively reduce the SEU resilient design overhead, the system failure rate and risk.

There is evidence that the most critical registers could be 'PC' and 'A15' based on the Table I data.

TABLE I
RESULTS OF FAULT INJECTION CAMPAIGNS

	DL	CD/IT	SDC	NE	Rank
D0			2	8	4
D1	1			9	5
D2	2		1	7	3
D3	1		1	8	4
D4				10	6
D5			2	8	4
D6				10	6
D7			1	9	5
D8	1		1	8	4
D9			2	8	4
D10	2		1	7	3
D11	1		1	8	4
D12			2	8	4
D13			1	9	5
D14			1	9	5
D15	1		2	7	3
PC	7			3	1
A2			1	9	5
A15	3		1	6	2
Total	19	0	20	151	

TABLE II
PROBABILITY DISTRIBUTION OF SYSTEM BEHAVIORS

	SDC	CD/IT	DL	FP	NE
Platform	10.5%	0%	10%	20.5%	79.5%

Table II provides the probability distribution of system behaviors when soft errors occur in the register set of Node1 host CPU. The probability distribution shown in Table II can be easily derived from the data of 'Total' row provided in Table I. This probability distribution shows the possibility of each failure mode resulting from a single soft error occurring in the register set. From the results of fault injection campaigns, we observe that a single soft error occurring in the register set has around 20.5% probability to cause the system failure. One thing should be pointed out that the probability of system failures caused by soft errors could vary for different workloads. There is evidence that the workload which uses heavily the register set should have higher failure probability than the workload with light use of register set.

TABLE III
SYSTEM FAILURE RATE AND ROBUSTNESS

	SER/H	FP	SFR/H	SIL
Platform	1.0e-5	20.5%	2.05e-6	1
Platform	1.0e-5	9.6%	9.6e-7	2

TABLE IV
SAFETY INTEGRITY LEVEL (SIL)

	PFH
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

Table III shows the system failure rate per hour (SFR/H) due to the errors happening in register set and the corresponding SIL defined in IEC 61508. The SFR/H can be derived from the multiplication of raw soft error rate per hour (SER/H) of register set and the failure probability (FP) caused by the soft errors. According to IEC 61508, if a failure will result in a *critical effect* on system and lead human's life to be in danger, then such a failure is identified as a *dangerous failure*. IEC 61508 defines a system's safety integrity level (SIL) to be the Probability of the occurrence of a dangerous Failure per Hour (PFH) in the system. For continuous mode of operation (high demand rate), the four levels of SIL are given in Table IV [5]. The SIL data in Table III can be obtained from Table IV. The data of SIL can be used to validate the system robustness/safety due to the register errors.

We use Table III to explain the robustness validation and fault-tolerant design process as shown in Fig. 1. We assume the SER/H of register set is 1.0e-5, and the SIL requirement is level two. So, we found that the robustness of system cannot meet the safety requirement. We now need to activate the phase 3 of Fig. 1 to develop a feasible risk-reduction approach by fault-tolerant design to improve the robustness of the critical registers identified in Table I. Since we use the commercial CPU, it is hard to add the robust design to the register set. Therefore, we cannot perform the fault injection campaigns for the fault-tolerant version. Here, we just assume the FP of fault-tolerant version is reduced to 9.6% due to the addition of soft error protection in the register set, so the SIL now satisfies the safety requirement. From this case study, we demonstrate how to perform the robustness validation, vulnerability analysis and fault-tolerant design to achieve the safety requirement for safety-critical automotive applications.

V. CONCLUSION

Characterizing the effect of soft errors caused by SEU on FlexRay-based steer-by-wire system is presented. We injected the soft errors into general purpose register set of FlexRay nodes to identify the most critical registers, the failure modes of the steer-by-wire system, and measure the robustness of the system. A FlexRay-based steer-by-wire system is used to demonstrate how to perform the robustness validation, vulnerability analysis and fault-tolerant design to achieve the safety requirement for safety-critical automotive applications. In the future, the impact of workload on failure probability and system failure rate will be addressed in more details.

ACKNOWLEDGMENT

The authors acknowledge the support of the National Science Council, R.O.C., under Contract No. NSC 100-2221-E-305-004.

REFERENCES

[1] R. Makowitz and C. Temple, "FlexRay – A communication network for automotive control systems," *IEEE Int. Workshop on Factory Communication Systems*, Page(s):207 – 212, June 2006.

[2] C. Constantinescu, "Impact of deep submicron technology on dependability of VLSI circuits," *Proc. IEEE Int. Conf. on Dependable Systems and Networks*, pp. 205-209, 2002.

[3] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design & Test of Computers*, vol. 22, issue 3, pp. 258 – 266, May-June 2005.

[4] Y. Zorian et al., "Impact of soft error challenge on *SoC* design," *Proc. 11th IEEE Int. On-Line Testing Symposium,* pp. 63 – 68, 2005.

[5] IEC 61508: Functional safety of electrical/electronic/ programmable electronic safety-related systems, International Electro-technical Commission IEC, 1998.

[6] S. Brown, "Overview of IEC 61508 design of electrical/electronic/programmable electronic safety-related systems," *Computing & Control Engineering Journal*, pp. 6-12, February 2000.

[7] ISO 26262: Road vehicles – functional safety, November 2011.

[8] A. H. Mollah, "Application of Failure Mode and Effect Analysis (FMEA) for Process Risk Assessment," *BioProcess International*, pp. 12–20, November 2005.

[9] M. Stamatelatos et al., Fault Tree Handbook with Aerospace Applications, version 1.1, NASA, 2002.

[10] FlexRay Consortium, "FlexRay Communications System – Protocol Specification," v2.1 Revision A, December 2005.

[11] Y. Kai et al., "Optimum Design and Calculation of Ackerman Steering Trapezium," *Int. Conf. on Intelligent Computation Technology and Automation*, pp. 1248-1252, 2008.

Yung-Yuan Chen received the MS degree in computer science and the PhD degree in electrical and computer engineering from State University of New York at Buffalo in 1987 and 1991, respectively. He is currently a professor and the chairman of the Department of Electrical Engineering, National Taipei University, New Taipei City, Taiwan. His research interests include fault-tolerant computing, safety-critical automotive system, FlexRay x-by-wire system, VLSI system design, computer architecture, reliable processor and SoC design with FMEA process and dependability assessment and validation.

Kuen-Long Leu received the MS degree from the Department of Computer Science and Information Engineering, Chung-Hua University, Taiwan. Currently he is the PhD student with the Department of Electrical Engineering, National Central University, Taiwan. His research interests include fault-tolerant processor, safety-critical automotive system, FlexRay x-by-wire system and SoC dependability verification platform development.