EasyChair Preprint

№ 6453

Estimating Residual Error Probability of Data Communication in Safety Critical Systems

Srikanth Srinivasan Kaniyanoor, Sivasubramanian Srinivasan, Laura Spinella, Elisa Spano, Gabriele Boschi and Nabajit Deka

August 28, 2021

Estimating Residual Error Probability of Data Communication In Safety Critical Systems

Srinivasan, Srikanth Kaniyanoor
Intel Corporation
India
srikanth.kaniyanoor.srinivasan@intel.com

Srinivasan, Sivasubramanian
Intel Corporation
India
sivasubramanian.srinivasan@intel.com

Spinella, Laura
Intel Corporation
Italy
laura.spinella@intel.com

Spano, Elisa
Intel Corporation
Italy
elisa.spano@intel.com

Boschi, Gabriele
Intel Corporation
Italy
gabriele.boschi@intel.com

Deka, Nabajit
Intel Corporation
India
deka.nabajit@intel.com

***Abstract*—Today's high-end cars have complex system architecture involving 50 to 100 electronic control units working together in order to achieve a common safety goal and comply with safety standards like ISO 26262. A complex SoC for such applications has multiple IP's that often implement black channel communication mechanism posing a challenge where a failure in communication can potentially compromise the safety goal. Hence risk reduction in communication channels is a vital component in the overall design for safety. However, any risk reduction approach always leaves behind a residual risk. In this regard, the permitted residual error rate for a communication channel is specified in IEC 61784-3. This paper provides a methodology of calculating the residual error rate as a function of failure rate of the communicating medium, the diagnostic coverage claimed by implementing the safety mechanism and the effectiveness of the CRC polynomial used with a case study.**

Keywords— functional safety, residual error, failure rates, diagnostic coverage, black channel communication, probability of failure, risk reduction techniques.

I. INTRODUCTION

Modern vehicles use multiple on-board controllers to manage different safety functions. Advance Driver Assistance System (ADAS) in modern vehicles receive inputs from multiple sensors viz cameras, LIDAR and radars, processes such data and drives actuating elements to enhance safety considering varying real-time situations. With the emergence of systems that demand multi inputs, multi-processors and multi-actuator configurations to augment safety, systems have migrated from federated architecture to integrated architecture [1]. Thus, multicore single die SoC are preferred for workload consolidation [2]. The use of software platforms like AUTOSAR[3] and virtualization are evolving to provide isolation between applications enhancing safety.

In order to meet stringent safety goals, both software and hardware system architectures should incorporate fault tolerant features in design. The ISO 26262 standard introduced in November 2011 for automotive electrical and electronic systems emphasizes safety practices right from the product design until product retirement. This standards is an adaptation from IEC 61508 used for industrial applications. Since it is not practically feasible to achieve zero risk in a system, the overall goal is to ensure that the residual risk (defined for a given ASIL) that exist after taking into account the safety mechanisms is well below the tolerable risk.

The standards [6] define the metrics required for achieving a given ASIL. Figure 1 shows the classification of faults for a hardware element.

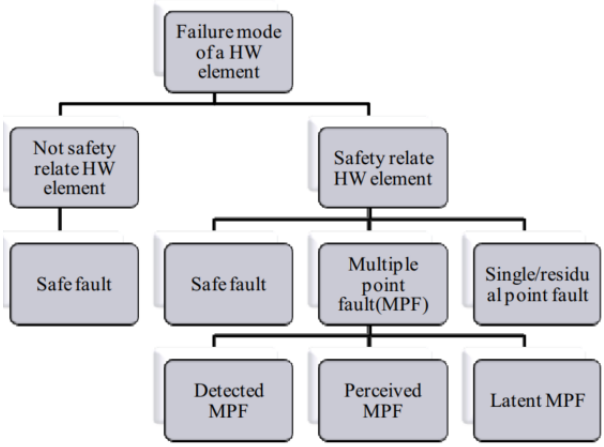

Figure 1 Classification of faults

Residual faults: They are a portion of faults in a system design that by itself can lead to the violation of safety goal. In safety architectures, critical elements that often demands attention for meeting safety objectives are the residual faults especially in communication channels between subsystems or IPs in SoC.

This paper presents a methodology to estimate the residual faults present in such black channel communication. Section II describes the safety mechanisms that needs to be considered for communication channels. Section III describes the residual error calculation through a case study. Section IV describes the related work done in the area of bit error estimation. Section V presents the conclusion and future work.

Note: The failure rates and the failure mode distribution reported in this paper are for illustration only. The actual failure rates and the error distribution will vary based on the SoC.

II. SAFETY MECHANISMS FOR COMMUNICAITON CHANNELS

A. Black Channel communication

A communication channel that is neither designed nor validated for safety according to the IEC61508 standards is called as black channel. Typical examples include TCP/IP, CAN, legacy protocols like SPI and I2C. In a multicore SoC, we can have similar communication channels within a die for data exchange between different IP's. They could be legacy IP's that may not be designed according to the safety standards. Thus in such black channels, often the approach of

implementing an End-to-End (E2E) safety protocol is adopted as a strategy for obvious reasons like limited insights in to safety compliance of the design. The typical End-to-End protocols include safety mechanisms to protect the safety telegram (message exchanged between two communicating peers). Of late, AUTOSAR is coming up with a specification for E2E protection of data using the E2E communication libraries.

B. CRC as a safety mechanism

Statistics in [9] shows that 79% of the data corruption failures are caused by single bit errors and can be easily detected if a good CRC polynomial is used. Table **I** shows an example of two dissimilar payloads having the same CRC.

Table I Different payloads with same CRC

CRC Polynomial	0x04C11DB7
Test Payload 1	0x706c756d6c657373
Test payload 2	0x6275636b65726f6f
Computed CRC	0x4ddb0c25
No of bits flipped	22

Therefore, there exists a non-zero probability for two different payloads i.e. corrupted and un-corrupted payloads to have the same CRC. This situation can lead to acceptance of a corrupted payload for processing which then could possibly lead to a dangerous failure. The probability for such a corrupted payload to have a valid CRC could be estimated and is referred as the "Residual Error Probability".

The overall goal is to do keep the residual risk for a communication channel not more than 1% of the IEC61508 requested PFH as shown in Figure 2.

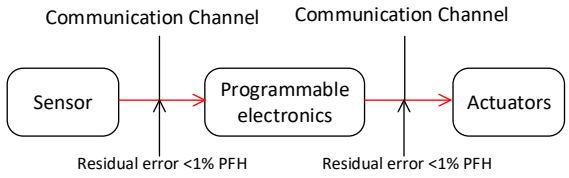

Figure 2 Recommended residual risk for communication channel

C. Considerations concerning CRC polynomial

Safety telegrams are usually transmitted in blocks of certain length (n). During the transmission, if we have k perturbed bits, then we can represent the bit error probability as

$$R_n = \sum_{k=d}^{n} \binom{n}{k} P_e^k * (1 - P_e)^{n-k} \quad (1)$$

Where d is the hamming distance. Further as described by the authors of [7], a weighing factor of 2^{-r} can be considered for a particular class of polynomials called as "proper CRC polynomials". The authors of [11] have provided a list of such proper CRC polynomials in their research.

Equation 1 can be refined for safety telegrams having a "proper CRC polynomial" as

$$R_n = \frac{1}{2^r} * \sum_{k=d}^{n} \binom{n}{k} P_e^k * (1 - P_e)^{n-k} \quad (2)$$

D. Failure modes and safety mechanisms in a communicaiton channel

The standards [5] and [6] describe the following communication failures.

a) **Corruption:** message corruption due to errors within a bus participant, errors on the transmission medium, or due to message interference.
b) **Unintended repetition:** repetition of the same message due to a fault or interference. Repetition of a message can be a normal procedure as apart of retry.
c) **Incorrect sequence:** Due to an error, fault or interference, the predefined sequence associated with messages, is not followed leading to out of order message delivery.
d) **Loss:** Missing message acknowledgement.
e) **Unacceptable delay**: Delay beyond an acceptable message window. This can happen due to a congestion or a message incorrectly queued in the FIFO.
f) **Insertion**: A message received from an unintended source.
g) **Masquerade**: Due to a fault or interference, a message from a non-safety element, is consumed by a safety related element as if the message is from a safety element.
h) **Addressing fault**: Incorrect delivery of a safety telegram, to an unintended recipient.

A subset of these faults is defined in the ISO 26262 standard [6] in table D.1.

In order to optimize the bandwidth, it is possible to transmit a safety telegram and a non-safety telegram on the same bus. It is recommended to use different data integrity systems for the safety telegrams and the non-safety telegrams. Since a non-safety telegram does not affect the safety function, it is also possible to send the non-safety telegram without any safety mechanisms.

The IEC61784-3 standards [5] recommend using safety mechanisms to minimize the residual risk as shown in Table II

Table II Safety mechanisms for communication errors

Safety Mechanisms / Failure Modes	Sequence number	Timestamp	Time expectation	Connection authentication	Feedback message	Data Integrity	Redundancy with checking	Different data integrity assurance
Corruption					X	X	X	
Unintended repetition	X	X					X	
Incorrect sequence	X	X					X	
Loss	X				X		X	
Unacceptable delay		X	X					
Insertion	X	X		X	X		X	
Masquerade				X	X			X
Addressing				X				

III. ESTIMATING THE RESIDUAL ERROR FOR COMMUNICATION WITHIN AN SOC – A CASE STUDY FOCUSED ON PERMANENT FAILURES

In the following we propose an example on how to estimate the residual error in a simple model of a Network on Chip (NoC). Even if it is a simple test case, the process can easily be adapted to a real design. We focus on permanent failure

distribution and detection. The total FIT would be a summation of the transient FIT and the permanent FIT.

An example of a NoC switch architecture is considered for this case study as illustrated in Figure **3**

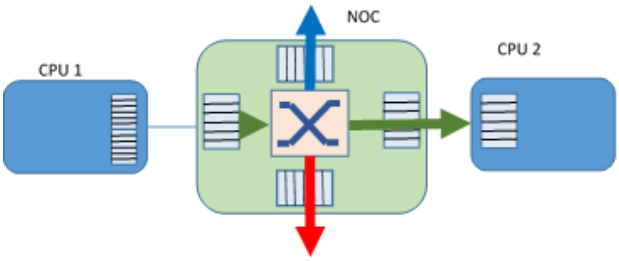

Figure 3 Simplified block diagram of NoC switch architecture

In this example, we have three input/output ports connected to one switch through a buffer and one input port that connects the switch to the processor. The switch determines the output port to which an incoming data must be sent based on routing information.
The NoC in the example can be divided into two different kinds of logic

- Data path (all the units that perform data processing)
- Control path (is the path responsible that manage the operation providing timing and control signals).

The failure modes reported in the **Table III** are extracted from the ISO 26262 part 5 table D.1.
Not all failure modes refer to data path, but actually, failure modes have been associated to data path or control path.

Table III Example of the NoC failure modes and main source

Failure mode	Main logic affected	Example
Corruption of information	Data path	Bit errors in data payload. (e.g. data-corruption may happen in FIFO, switch)
Insertion of information	Data path	Error in destination address (e.g. A message can be received from an unintended source.
Loss of information (drop packets)	Control path	Switch receives data but never sends it to output. (e.g. congestion can cause loss of packet in case queues are too small)
Delay of information	Control path	Data transfer too early/too late (e.g. congestion can cause increase of delay in case queues are too big and congestion lasted short time)
Incorrect sequence of information	Control path	Switch (e.g. faulty state machine operations)
Repetition of information	Control path	Faulty empty signal from FIFO inside switch – old data will be resent
Incorrect addressing	Control path	Error in routing matrix – misrouted packets

The Failure Mode Distribution (FMD) provides the mean to account for how the component failure rate is apportioned across different ways the component can fail.
As per ISO-26262:5, note 2 in Table D.1, if an element has the failure modes x, y and z (e.g. fault in fifo, fault in switch, fault in router, fault in bus, etc) with a failure mode distribution of X Y and Z then the effective (resulting) diagnostic coverage can be calculated as shown in (3):

$$K_{dc} = K_{fmc,x} * X + K_{fmc,y} * Y + K_{fmc,z} * Z \qquad (3)$$

Where:

- K_{dc} is the resulting diagnostic coverage
- X is the failure mode distribution for failure mode x
- Y is the failure mode distribution for failure mode y
- Z is the failure mode distribution for failure mode z
- $K_{fmc,x}$ is the failure mode coverage of failure mode x
- $K_{fmc,y}$ is the failure mode coverage of failure mode y
- $K_{fmc,z}$ is the failure mode coverage of failure mode z

E2E protocol including the following safety mechanisms selected from that listed in Table II is considered.

- Data Integrity
- Sequence number
- Timing expectation
- Sender and receiver ID

For this example, only the path from the CPU1 to CPU2 (green arrow in Figure 3 is covered by E2E, we can estimate the coverage of the whole NoC taking into account the following assumptions:

1. The percentage of logic of NoC relative to the green path (eg 50% green path, 30% red path and 20% blue path)
2. The percentage of area of the data path with respect to the control path (eg 80% data path and 20% control path)
3. The failure distribution of each failure mode both for the data and control path (an example shown in Table IV).
4. The coverage of the different safety mechanism applied to each failure mode (as shown in Table V)

Table IV: Example of Failure mode distribution (FMD) both for data path and for control path

FM	FMD Data Path	FMD Control Path
Corruption of information	80%	0%
Insertion of information (from unintended)	20%	0%
Loss of information (drop packets)	0%	20%
Delay of information	0%	20%
Incorrect sequence of information	0%	20%
Repetition of information	0%	20%
Incorrect addressing	0%	20%

Table V: Example of Safety mechanism(SM) Diagnostic Coverage (DC) for each Failure mode (FM)

FM	SM	SM DC
Corruption of information	Data integrity (CRC)	From 90 to 99.99%
Insertion of information (from unintended)	Sender/Receiver ID	90%
Loss of information (drop packets)	Sequence number	99%
Delay of information	Timeout	60%
Incorrect sequence of information	Sequence number	90%
Repetition of information	Sequence number	90%
Incorrect addressing	Source and destination address	90%

In case of 80% FMD for data path and 20% FMD for Control Path is considered, the overall E2E diagnostic will change from 89.16% to 94.92%, as a function of claimed CRC coverage.
The diagnostic coverages have been obtained by applying formula (3) with the following assumptions

- FMD of the data path is 80%
- FMD of the control path is 20%

- The DC of the CRC was varied from 90% to 99.9% keeping the remaining DC as constant for this example

The effective DC calculation can be expressed as a pseudo code as shown below

K_{dc} = 0

For each I in FM

$K_{dc} = K_{dc} + K_{fmc,x} * x(i) * FMD(i) *$ Fabricsuage(i)

Next I

The resultant diagnostic coverage K_{dc} is obtained for the NOC shown in Figure 3 considering a 50% logic covered by the green path as stated in the assumptions. In case we need to estimate the diagnostic coverage of additional paths (blue, red), the corresponding logic usage needs to be applied.

A. *Method of converting the failure rate to an equivalent BER*

This section provides a method of converting the failure rates into an equivalent BER.

A typical safety telegram will contain information as shown in Figure 4

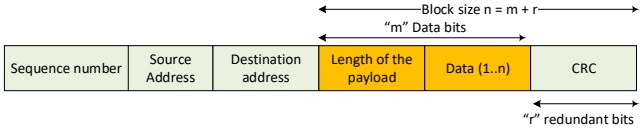

Figure 4 Typical contents of a safety telegram

Table VI lists the abbreviations used in this section.

Table VI List of abbreviations

Term	Description
$R_n(k)$	Maximum residual error probability
r	Number of bits in the CRC
d	Maximum hamming distance
nCk	Combination of n things taken k at a time
$(P_e)^k$	Probability of "k" bits getting flipped in a safety telegram
$(1-P_e)^{(n-k)}$	The probability of the remaining bits (n – k) not being flipped
V	Number of safety telegrams exchanged per hour
m	Number of sinks (Number of notes receiving the safety telegram)
Λsc	Failure rate of a black channel communication. This parameter should be <= 10^{-9} for achieving SIL2 and 10^{-10} for achieving SIL3
CRC	Cyclic redundancy check

The maximum residual error probability is given by (4)

$$R_n(k) = \frac{1}{2^r} * \sum_{k=d}^{n} \binom{n}{k} P_e^k (1 - P_e)^{(n-k)} \quad (4)$$

B. *Calculation of the residual error rate*

Starting from example in the previous paragraph the failure rates for NoC is computed as a part of the detailed FMEDA calculations.

The likelihood of the bits getting flipped while transported through the black channel is a function of the number of bits transferred, the failure rates of the path and the duration the data resides in a given fabric.

For this case study, we assume:

- 0% of safeness
- NoC FIT rate = 280 FIT
- 1024 bytes transmitted every 10ms.

NoC Residual FIT (CRC DC=99%) = 14.3 FIT
No of bits exchanged per payload = 1024 * 8 = 8192
Periodicity = 10ms
Number of bits exchanged per second = 819200
Fabric usage per hour = 0.012288 hour
P_e = Total fabric usage per hour * FIT = 1.76 $\cdot 10^{-10}$
$R_n(k)$ can be calculated using equation (4)
$R_n(k) = 1.176 \cdot 10^{-28}$

The residual error rate is given by the formula in (5) as stated in [5].

$$\boldsymbol{\lambda_{RE} = R_n(k) * V * m} \quad (5)$$

Assuming that we have only 1 sink the residual error will be 4.23 $\cdot 10^{-23}$. Which is well within the requirements of 10^{-10} (1% of the permitted PFH).

In case of a broadcast message that can reach multiple safety nodes, the number of sinks will be equal to the number of nodes receiving the safety telegram.

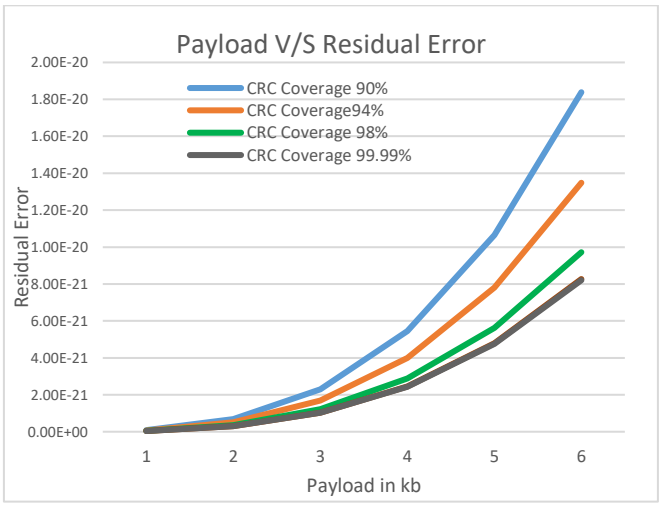

Figure 5 Residual error as a function of payload and DC coverage

Figure 5 shows the relationship between the residual errors with respect to the payload and the CRC coverage (which is a function of HD, number of crc bits, etc). Since the residual error rate is low, it permits exchange of higher data rates between IP's within the SoC.

IV. RELATED WORK

During the literature survey, the authors observed three different approaches followed to estimate the bit error rate in digital communication.

The authors of [8] have used OpenSafety protocol and estimated residual error as a function of burst error rate of the communication channel. The payload in this protocol is limited to 250 bytes, which uses transmission redundancy as described in the ISO standards [4]. Thus, it cuts down the bandwidth of the safety telegram to 125 bytes. The Burst error rate calculations make use of IEC 610004 EMI and EMC standards. The key take away from this work is that for SoC where we need to consider high data transfer, use of

transmission redundancy can negatively affect the throughput.

The authors of [9] in their work determine the communication bit error rate (BER) as a function of the BER of the cables as per equation B.4 mentioned in IEC61784-3[5].

The authors of [10] describe bit error rates for power line communication and discuss modelling methods for low speed (19200 bps) communication typically used for wiper and windscreen actuators.

While all the above authors consider BER as a critical parameter in safety systems design, they have not discussed estimating the bit error rate for inter die communication.

V. CONCLUSION AND FUTURE WORK

The above work explains how it is possible to build a fault tolerant communication system to support higher throughput for a single die multicore SoC. It identified the shortcomings in the existing literature regarding this topic and proposed a methodology to estimate the BER given the FIT rate.

Care should be taken to select a good CRC polynomial as described in [11] taking into consideration the hamming distance which is a function of the payload size. If a weak polynomial is selected, then the equation (3) cannot be applied to calculate the residual error rate.

With a proper selection of the CRC polynomial, identification and implementation of the safety mechanisms and the FIT rates of the communicating medium, we can calculate the residual error rates and fine tune the payload length that can be transmitted.

The maximum permitted residual error rate for a given communication link should be limited to 1% of the permitted PFH and can be estimated from the BER as a function of FIT.

The methodology explained in this paper can be verified experimentally by conducting a fault injection test. The methodology can be extended to incorporate the failure rates in the communicating medium due to transient faults.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support provided by Mondreti, Srinagesh, Chiavacci, Monia, Pillai Manikandan K, Rajesh Banginwar and Prasad Ganesh for motivating us to pen our findings in the form of this paper.

REFERENCES

[1] M. Di Natale, A. L.S Vincentelli, Moving From Federated to Integrated Architectures in Automotive: The Role of Standards, Methods and Tools, Proceedings of the IEEE Vol. 98, No. 4, April 2010.

[2] M.Geier, M.Becker, D.Yunge, B.Dietrich, Reinhard Schneider, Dip Goswami, Samarjit Chakraborty, Let's put the Car in your Phone! DAC '13, May 29 - June 07 2013.

[3] C. Avasalcai, D.Budhrani and P.Pop, Work-in-Progress: Towards Industry Strength Mapping of AUTOSAR Automotive Functionality on Multicore Architectures.

[4] IEC 61508 Functional safety standards, 2010

[5] IEC 61784-3 Safety standards for communication, 2016

[6] ISO 26262 Safety standards for automotive, 2011

[7] S. LEUNG-YAN-CHEONG AND M. HELLMAN, Concerning a bound on undetected error probability, IEEE Transactions on Information Theory, Volume 22, Issue 2 March 1976.

[8] A. Platschek, B.Thiemann, H.Zeilinger, T.Sauter, An Error Model for Safe Industrial Communication, November 2015.

[9] M. Rahmani, W.Hintermaier, B.Mu¨ller-Rathgeber,E.Steinbach, Error Detection Capabilities of Automotive Network Technologies and Ethernet - A Comparative Study 2007 IEEE Intelligent Vehicles Symposium, June 13-15, 2007.

[10] M. Wilson, H.Ferreiray, R.Heymanny and A.Emlehy, Bit Error Recording and Modelling of In-Vehicle Power Line Communication, 2014 18th IEEE International Symposium on Power Line Communications and Its Applications

[11] Dr P. Koopman, Carnegie Mellon University, Best CRC Polynomials https://users.ece.cmu.edu/~koopman/crc/ , 2015