UNIVERSITÄT
HEIDELBERG

International Workshop on the
Design of Dependable Critical Systems

ECOMODIS

Proceedings

of the
International Workshop on the
Design of Dependable Critical Systems
“Hardware, Software, and Human Factors
in Dependable System Design”

DDCS 2009

September 15, 2009
Hamburg, Germany

In the framework of
The 28th International Conference on
Computer Safety, Reliability and Security
SAFECOMP 2009

Edited by

Achim Wagner[1], Meike Jipp[1], Colin Atkinson[2] and Essameddin Badreddin[1]

[1] **Automation Laboratory, Institute of Computer Engineering,**
University of Heidelberg
[2] **Chair of Software Engineering, University of Mannheim**

ziti

International Workshop on
the Design of Dependable Critical Systems
September 15, 2009, Hamburg, Germany

Fault Propagation Analysis on the Transaction-Level Model of an Acquisition System with Bus Fallback Modes

Raul S. Fajardo Silva[*], Jürgen Hesser, and Reinhard Männer

Department for Application Specific Computing, University of Heidelberg,
B6, 68159 Mannheim, Germany
{raul.fajardo@ziti.uni-heidelberg.de,
juergen.hesser@medma.uni-heidelberg.de}

http://li5.ziti.uni-heidelberg.de/

Abstract. The early fault analysis is mandatory for safety critical systems, which are required to operate safely even on the presence of faults. System design methodologies tackle the early design and verification of systems by allowing several abstraction for their models, but still offer only digital bit faults as fault models. Therefore we develop a signal fault model for the Transaction-Level Modeling. We extend the TLM generic payload by the signal characteristics: Voltage level, delay, slope time and glitches. In order to analyze and process these, a TLM bus model is created, with which signal faults can be detected and translated to data failures. Furthermore, inserting this bus in an acquisition system and implementing fallback modes for the bus operation, the propagation of the signal faults through the system can be assessed. Simulating this model using probability distributions for the different signal faults, 5516 faults have been generated. From these, 5143 have been recovered, 239 isolated and 134 turned into failures.

Key words: Signal faults, mixed signal verification, system design, fault modeling, system model

1 Introduction

Safety critical systems have to operate safely even on the presence of faults. It means that malfunctioning components have to be located and its faults isolated, so that it does not propagate to its user. The behavior of the system on the presence of faults can be analyzed using a model of the system. For that, faults and methods for localization, isolation and correction have to be modeled. In the

[*] Raul S. Fajardo Silva and Reinhard Männer are with the Department for Application Specific Computing. Jürgen Hesser is with the Institute of Experimental Radiotherapy, University of Heidelberg, Theodor-Kutzer-Ufer, 1-3, DE 68167-Mannheim, Germany.

case of an acquisition system the communication buses connected to the sensors are influenced externally by the environment and by each communicating node, being a critical point of the design.

The early design of complex hardware systems including software and hardware parts, interfacing with the real world and user is aided nowadays by system design methodologies. System design [1] abstracts the behavior of the system components by the specification of its function. In order to effectively design system communication, the Transaction-Level Model has been developed [2]. It allows the design of the communication to be independent from the components or architecture design. Furthermore the detail of the model can span from function calls to pin signaling.

In this paper we analyze the propagation of signal faults through a synchronous bus in a Transaction-Level model of an acquisition system. This system is composed by multiple sensors connected to a bus, a bus master and a CPU, which pools the data. First the bus, its modes and operating characteristics are modeled. The selection algorithm of fallback mode is placed on the communication controller, the bus master. For the fault injection, probability distributions are defined for the characteristics of the signal: Delay, slope level, voltage level and glitches, thus statistically generating faults. This faults are traced by the model so that their propagation results can be later evaluated.

The next section explains the bus model, its operating modes, the fault analysis and fault procesing modules. Section 3 presents the acquisition architecture simulated in this paper, the fallback selection algorithm and the fault generation, followed by the simulation results. In section 4 the conclusion of the work is presented.

2 Bus Model

In order to model a signal fault aware bus and its fallback modes, the TLM library is used. The actual standard considers performance issues related to the communication, but does not include operating characteristics to assure communication. The standard comprehends standard blocking and non-blocking transport interfaces and defines a standard payload[1] which includes performance characteristics, such as delay and latency [2]. We extend this standard payload to include the signal quality factors: Delay, slope level, voltage level and glitches, which are directly related to the bus operating capability. Furthermore, the model of the synchronous bus holds its operation mode: Operating frequency, clock phase and connected nodes.

Payload extensions contain both the value of signal characteristics of the transmission and a record of the violation of their limit (i.e. signal failure). When forwarding read calls, the extension is ignored, the signal characteristics are set by the callee. When the callee sends back the payload, the bus analyzes these, assigning the correspondent signal failure if the bus operation limits are

[1] An instance of the standard payload corresponds to a packet, when modeling a regular communication protocol.

exceeded. Furthermore the bus modifies the payload transmitted data, according to the occurred signal failures. At last, the complete payload is sent to the caller, fig. 1. Based on the failure record, the caller can then decide to change the operation mode in order to avoid further failure. On write calls, the bus first analyzes the signal characteristics and processes the data, then forwards these to the callee, ignoring the extension on return, fig. 1.

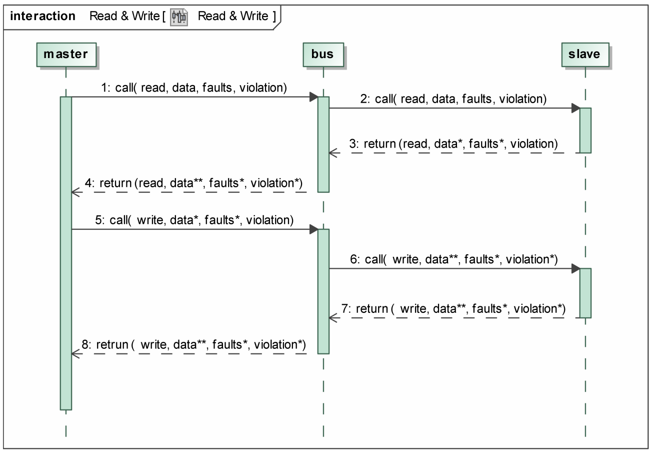
interaction Read & Write [Read & Write]
master
bus
slave
1: call(read, data, faults, violation)
2: call(read, data, faults, violation)
3: return (read, data*, faults*, violation)
4: return (read, data**, faults*, violation*)
5: call(write, data*, faults*, violation)
6: call(write, data**, faults*, violation*)
7: return (write, data**, faults*, violation*)
8: retrun (write, data**, faults*, violation*)

Fig. 1. Write and read calls to the bus (* represent that the variable has been set, ** modified)

2.1 Modeling Signal Faults

Prototype based communication monitoring techniques of [3] and [4] categorize bit faults, glitches and delays. [5] define possible signal faults of car sensors, as abnormal magnitudes (voltage levels), rolling (slope times), noise and dependency faults (context dependent). We categorize signal and bit faults related to digital hardware communication as a combination of both. A digital signal is ideally represented by two voltage levels, with instantaneous switching between both levels. For a real electronic component to drive its output from one logic level to the other, the resulting signal has a ***slope time***, which can be measured as a time degraded behavior. The same applies for ***delays***, which represent the response time of a component. Degraded ***voltage levels*** are variations of the output voltages for the logic levels approaching its boundaries, while ***glitches*** are voltage pulses of short duration resulting from interferences from outside. These four signal and bit faults of digital signals (fig. 2) are used as quality measurement of a transmission. These are then further divided for the characterization of a complete frame, composing the actual signal characteristics included in the payload extension: Both high and low bits voltage level; rise and fall time; delay; glitch time, level and count.

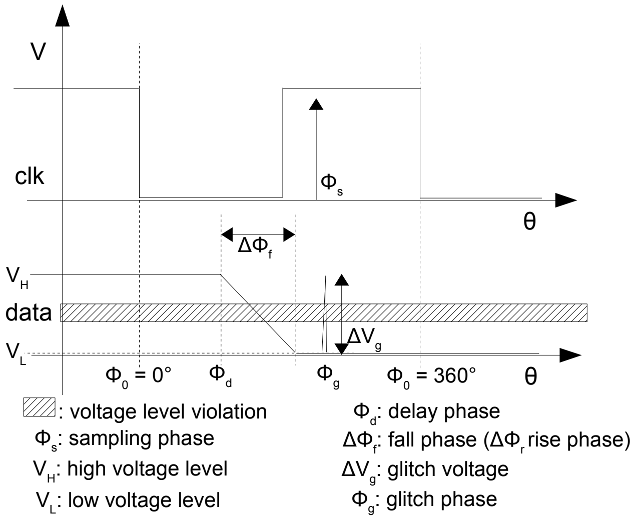

Fig. 2. Time normalized signal characteristics (time multiplied by operating frequency resulting in phase values)

Signal Conditions	Signal Failure Detection	Processing on Data
$V_H < 2.0V$	High bit Voltage Level	All 1s to 0s
$V_L > 0.8V$	Low bit Voltage level	All 0s to 1s
$\phi_d > \phi_s$	Delay	Rotate data to the right
$\phi_d + \phi_r > \phi_s$	Rise time	Assuming $x[n]$ the series of the data bits $y[n] = \begin{cases} 0, x[n-1] = 0 \\ x[n], \text{otherwise} \end{cases}$
$\phi_d + \phi_f > \phi_s$	Fall time	Assuming $x[n]$ the series of the data bits $y[n] = \begin{cases} 1, x[n-1] = 1 \\ x[n], \text{otherwise} \end{cases}$
Glitch count > 0 $\phi_s - 18° < \phi_g < \phi_s + 18°$	Glitch time	Nothing
$V_H - \Delta V_g < 2.0$	Glitch high level	If glitch time All 1s to 0s
$\Delta V_g + V_L > 0.8$	Glitch low level	If glitch time All 0s to 1s

Table 1. Signal conditions for signal failure detection (limit for bus operation) & Processing of the data according to detected signal failure (for the series, index 0 is bit 7 for a byte)

2.2 Fault Analysis and Processing

This module analyzes the signal characteristics of data being transmitted through the bus. The data sender sets the signal characteristics for the transmission. These are then compared to the conditions on table 1 to detect signal failures. The listed conditions are based on the limits imposed by the operation of the bus. For comparison, the timing signal characteristics are normalized to phase signal characteristics depending on the operating frequency of the bus. The bus operation conditions, sample time and clock phase are merged to the ϕ_s sampling phase. Logic levels and sample time are implementation dependent and thus constant, not influencing the relationship between operation mode and violation limits. Signal failures lead to data failure. In order to model that, the processes described in table 1 are carried out for each detected violation.

3 Acquisition Architecture

The architecture modules, acquisition CPU, bus master and sensors are modeled in SystemC using the Loosely-Timed coding style of the Transaction-Level Model, calling thus blocking transport only. The architecture connects the acquisition CPU to the bus master, which is connected to the sensors through the previously modeled TLM bus, fig. 3.

In the model of the acquisition CPU, only the acquisition pooling function is modeled. The bus master contains a thread safe buffer implementation, which is accessed by the CPU. To the other side it interfaces with the bus, executing two tasks. First, it request the data of every sensor. Then, if the bus detected a signal failure the bus master may change the operation mode of the bus and retry transmission. Furthermore, the operation mode of the bus can be periodically reset to raise bus performance, this also reconnects previously isolated nodes, which might have been faulty for a short period of time only.

Each sensor continuously reads data from a different input file, which can be accessed by calls to the blocking transport method. Upon each sensor access, the signal characteristics of the TLM extended payload are set. Despite of glitch count, these signal characteristics follow a Gauss distribution. The values for the mean and the standard deviation of the distributions can be set on sensor instantiation. The initialization value of the geometric distribution for the glitch count is equal the chance of no glitch occurrences in a bit. The statistical variable glitch count is then calculated by $framebits/x_k$.

3.1 Fallback Modes

In the bus master the fallback mode selection algorithm (fig. 4) can be activated. The bus master gets the information about signal failure occurrences from the bus instance. If the algorithm is activated and any failure occurs, a selected fallback mode is assigned to the bus by the bus master. Directly after mode change, a single transmission retry is carried out, for which neither fallback mode nor further retries are activated. After this transmission is completed, fallback modes can continue to be assigned.

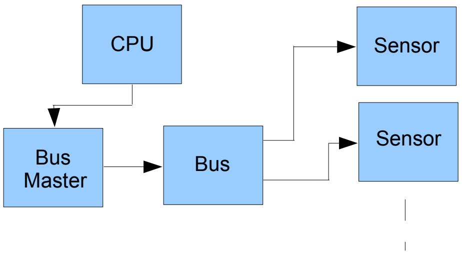

Fig. 3. Acquisition System Architecture

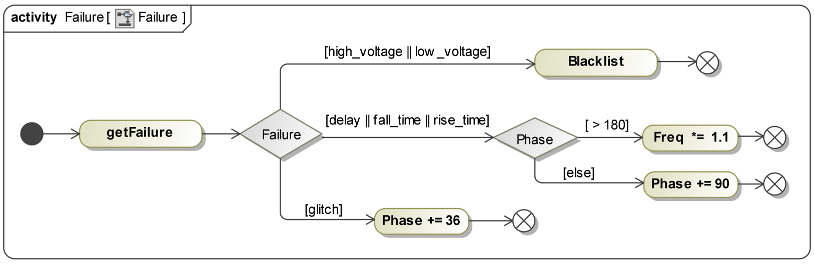
activity Failure [Failure]
getFailure
Failure
[high_voltage || low_voltage]
Blacklist
[delay || fall_time || rise_time]
Phase
[> 180]
Freq *= 1.1
[else]
Phase += 90
[glitch]
Phase += 36

Fig. 4. Select algorithm for fallback mode

3.2 Results

During the simulation of the model all data is accepted by the acquisition CPU. Faulty data is marked on simulation and counted, if faults are detected, information about isolation or correction is logged, otherwise failure occurrence is asserted. With this data, fault propagation analysis can be made, producing statistics about the robustness of the model against the environment modelled by the probability distributions.

An environment is defined in the table 2. Bus works with a 100kHz frequency clock, sample phase of implementation 216°, and TTL logic levels (bit 0: 0.8 V/bit 1: 2.0 V). For a simulation on this configuration the values of total system faults (signal failures), fault isolation, fault recovery and failure occurrence are compared for 2 modes: Fallback reset on/off. Its results are presented in table 3. Mode fallback off does not isolate neither recover any fault, the same test for a fallback off bus produces the same amount of failures as arisen faults. The signal outputs of the data received by the CPU for a simulation with fallback turned off and on can be seen on fig. 5.

Applying signal fault detection and adapting the bus operation mode accordenly, 97% of the faults generated by the faulty behavior described in table 2 could be recovered, the remaining 3% have occurred on the retry transmission after fallback mode set. In this situation no further retry is activated.

Signal Characteristic	Mean	Standard Deviation
High bit Voltage Level	3	0.35
Low bit Voltage level	0	0.3
Delay	$4\mu s$	$1.2\ \mu s$
Rise time	$2\mu s$	$0.1\mu s$
Fall time	$2\mu s$	$0.1\mu s$
Glitch time	$4\mu s$	$0.5\mu s$
Glitch level	0.5	0.1

Table 2. Signal characteristic of sensor bus connection. Glitch count initialization value is 0.8, that is 80% chance of glitch free bit

Number of	Fallback reset ON	Fallback reset ON
Transmissions	40000	40000
Transmission Retries	2833	6
Blocked Transmissions	457	39198
Faults	5516	8
Isolated Faults	239	4
Recovered Faults	5143	8
Failures	134	0

Table 3. Test results for fallback reset ON and OFF tests

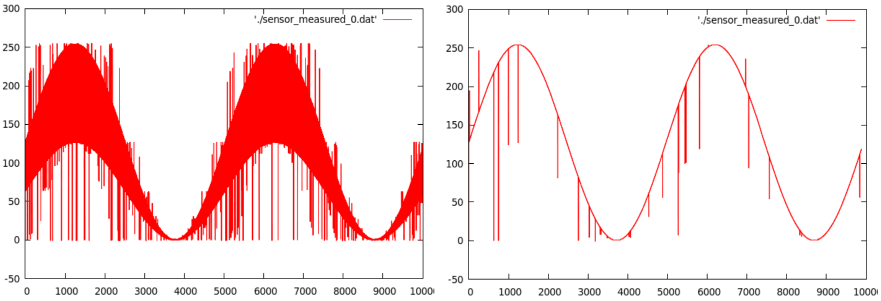

Fig. 5. Testing signal faults on bus: left fallback modes off, right on

4 Conclusion

The verification using classic hardware description languages evolves towards applying mixed signal verification to reduce uncertainty about the interoperability between analog and digital systems. Faults in the different abstraction levels of TLM have not been yet completely modelled. In this paper we have introduced a mixed signal verification strategy for TLM models, which profits from early verification of system design.

In order to process and analyze signal faults created in the system, we first developed a signal fault model, based on standard signal quality characteristics. Afterwards, an algorithm for detecting these faults based on operating properties of the same bus was created. Similarly, the same bus processes the transmitting data generating data failures according to the detected signal faults.

Then we inserted the developed bus in a TLM model of an acquisition system to reason about fault propagation through a bus with fallback modes. Here a bus master is implemented, which controls the bus, providing the bus with different operation modes. Faults have not been directly injected in the system. Instead, probability distributions have been assigned to the different signal characteristics of the sensors, building the environment of the system, which statistically generates faults.

The description of the signal characteristics of the sensors is realistic and can be easily adapted to different conditions. The online adaptation of the operation modes of the bus is able to isolate and correct almost every fault by sacrificing performance. In a future work we plan to compare this results with the fault tolerance of communication protocols with error correcting codes and error detecting codes with retries.

References

1. Grant Martin, Brian Bailey, and Andrew Piziali. *ESL design and verification a prescription for electronic system-level methodology.* Morgan Kaufmann, 2007.
2. Open SystemC Initiative. http://www.systemc.org.
3. R. Pallierer, M. Horauer, Zauner M., A. Steininger, E. Armengaud, and F. Rothensteiner. A generic tool for systematic tests in embedded automotive communication systems. In *Proc. of the Embedded World Conference*, 2005.
4. E. Armengaud, F. Rothensteiner, A. Steininger, and M. Horauer. A method for bit level test and diagnosis of communication services. In *Proc. of the IEEE Workshop on Design & Diagnostics of Electronic Systems*, 2005.
5. J. A. Crossman, Hong Guo, Y. L. Murphey, and J. Cardillo. Automotive signal fault diagnostics - part i: signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection. 52(4):1063–1075, July 2003.