Automation of model-based signal integrity analyses

Marko Moch, C&S group GmbH

Nowadays we discover a changing focus at car development. To handle the growing system- and testing complexity, simulation comes to the fore. This process is supported by innumerable tools and methodologies with the goal to speed-up and simplify the development of new automobiles and technology.

The tools are mostly highly specialized and dedicated only to a tiny area of the development process or on the other hand are such generic that they cannot be used without customization.

In the area of subjects of C&S group and its customers it occurs very often, that the validation of real and simulation-based networks and components must be integrated into the development processes of the customer. The experience of C&S group in collaboration with its customers has shown, that the validation of simulation-based network descriptions gains more and more acceptance.

Therefore C&S group has developed a new automation tool to validate network topologies, like they are used in automotive, avionics and industry areas, with the aid of a new model-based testing methodology.

This paper shall present the new automation tool and shall generally enlighten the model-based testing methodology with practical examples.

Introduction

The growing complexity of electrical and electro-mechanical systems in today's top notch technologies requires a massive amount of testing to work properly under all (defined) circumstances. To assure the correct functioning of these electrified systems, different aspects need to be investigated which would stress those systems.

As these systems are distributed networks and communicating over distance between the participating electronic control units (ECU), the possibilities are growing in disturbing the communication compared to a single circuit board. The communication on its lowest level (the physical layer) is based on driving electronic charge, controlled by different voltage levels and thresholds.

With the growing complexity more and more supplying companies such as semiconductor device developers come into play as well. It is therefore required, that their components work with each other and can be exchanged without problems.

We speak about the signal integrity analysis when validating these values to be correct in terms of a defined specification of a communication platform, such as the field busses CAN or FlexRay in our area of subjects.

Signal Integrity

The signal integrity of network systems like CAN or Flex Ray can be affected and stressed by different analogous internal (to the network) or external sources as well as failures across the network system. Worst case operating areas (e.g. high or low temperatures) and finally, silicon production spread adding the extra portion of negative impacts to the system.

These influences reshape the analogue signals in a way, which can negatively affect the signal- or generally spoken, information timings that are crucial for the functioning of the upper ISO/OSI layers and therefore the performance of the system. Also, the incorrect, up to the destructed transmission of signals may even destroy the networking function of the system permanently.

Such influences may be ESD- or EMV injections, ground shifts, short or open circuits, supply losses, etc.

To evaluate these influences, they need to be measured appropriately. Apart from that, the network system requires to be robust against these influences. This adds more protective components to the whole network, especially to the ECUs. But, adding more components refers also to add more signal deformation. That's why the impact of those protective components needs to be measured, too.

Validation Process

A distributed network, e.g. for a car, needs to be validated at certain (design decision based) time stamps within the development process before the production starts. These time stamps refer to different kinds of validation with a particular purpose within the development process.

Nowadays three different, major kinds can be seen:

n Simulations

n Specialized hardware setup measurements

n Integrated vehicle measurements

The simulations can be processed at a very early stage of development. It also allows for examining several hundreds to thousands of different variations of a network to find the right solution. It focuses on the interoperability between the used or possibly usable components and how much safety margins can be held for the upper ISO/OSI layers in terms of robustness against stress at the physical layer. With this, a manufacturer has the opportunity of reducing development costs, because the amount of prototypes, measurements and personnel costs reduces to a minimum.

Such prototypes are used in specialized hardware setups, where the investigation focuses only on the physical layer of the considered topology and is used mainly by approving the simulation results of a chosen harness design solution for the final product.

The integrated vehicle measurements are the latest step during the development process, where fully implemented ECUs are validated in an entire integrated car environment. Possibilities for observation of physical layer effects in this implementation phase are severely limited. Usually the effects can only be checked up indirectly through all of the ISO/OSI layers.

When we talk about model-based signal integrity analysis of in-vehicle, avionics or industrial field bus networks (e.g. CAN, FlexRay), we mean this by the investigation of the physical layer part of a complete distributed network with all of its wiring harness and ECUs or only parts of them using simulation.

A network description can be split up in three different physical application levels, referred to as

n bus driver / transceiver devices,

n ECU interfaces and

n topologies.

The analysis of each of them focuses on different aspects of the whole system.

The focus for bus drivers / transceivers is their corner case behaviour in terms of high or low temperature operating areas and silicon production spread resulting in different speed and strength of their output stages and also the delays / timings of the internal logic, as well as the built-in immunity against external ESD / EMV pulse injections.

The focus at the ECU circuitry in addition to the bus driver's / transceiver's capabilities is to suppress these injections as well as own emissions. This can be done by using common mode chokes and ESD protection devices.

The highest physical application level is the topology. This is where the structure of the interconnections between the ECUs plays a role in conjunction with the used cables and the network termination points.

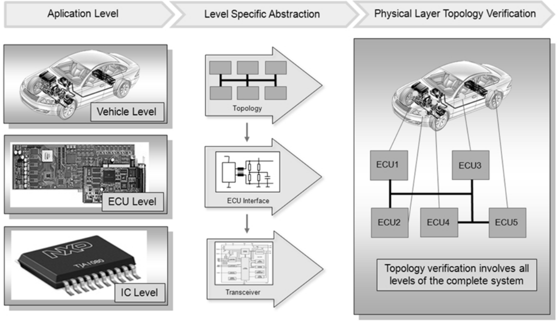
Aplication Level
Level Specific Abstraction
Physical Layer Topology Verification
Vehicle Level
ECU Level
IC Level
NXP
TJA1080
Topology
ECU Interface
Transceiver
ECU1
ECU3
ECU2
ECU4
ECU5
Topology verification involves all levels of the complete system

Figure 1: Validation process

What’s the deal with Simulation?

Simulation allows the creation of network design rules by investigating network limitations together with limits of used electronic network components through analysis of worst-cases, which is not possible to be investigated using hardware set-ups. In constrast to hardware set-ups, simulations tolerate much more observation points without increasing the expenditures for additional measurement equipment and the chained prototype costs. However, a mandatory prerequisite is the availability of suitable models and adequate simulation tools, sufficient to provide reliable and meaningful results.

One drawback of the “virtual validation” of networks was the low acceptance of simulation at this particular area in the past. Nowadays this thinking is changing towards physical layer simulations. They are used more and more frequently because real device measurements are time consuming in terms of personnel resources as mentioned before. Also they need expensive prototypes and the measurements and test management are mostly or completely hand operated.

The second drawback comes with the tools for simulation and also for real device measurements to a certain extend. These tools, on one hand, are highly specialized and dedicated to only a tiny area within the development process, but covering this area to a maximum, and of course, required extend. On the other hand there are tools designed as general purpose giving the developer a lot of functionalities but not the final solution for his needs and processes.

Additionally, simulator software is expensive. Its complex usage requires specialist knowledge and therefore distorts the cost benefit of the simulation to an unreasonable amount.

The goals to achieve

Before we can go to further details of the solution of how to overcome the complexity involving simulation into the validation process of fieldbus networks, it has to be clarified, which goals should be achieved by the simulation solution in terms of what are the general requirements of the automated and model-based signal integrity analysis:

n *Independence by the description and use of proprietary model components for all applied models,*

to have approved models by the device developer

n *Reproducibility of the results,*

to have the opportunity to repeat simulations and use different simulators leading to same results (within a defined level of uncertainty)

n *Preferably high grade of reusability of the applied model components,*

to reduce development costs of the model based test system

n *Standardized, interchangeable and therefore interoperable models,*

to have the opportunity of using one test system for all devices and therefore comparability between different devices

n *Comparability to real device measurements and their results,*

which is the main requirement to meet in a device-technical point of view in order to replace prototype measurements by simulation

n *Speeding up the topology validation,*

to reduce the expenses which are mostly personnel efforts n *As much as possible variation of the components' configuration,*

to have the opportunity of using the simulation at an early stage during the development process for the concept topology layout and circuitry verifications

Finally, it has to be clarified which specific requirements need to be checked to assure the signal integrity of a topology network concerning which measurements should be done at which pin point within the topology, just to name a few possible measurements around the transceiver device within an ECU:

n Analogue voltage levels at all possible logical states

n Logical signal delay timings to serve a robust bit timing / bit synchronisation

n Analogue signal settling times at logical signal event changing up to the bit sampling point

n Logical signal ringing at bit transmission

The model-based approach

To analyse the signal integrity of a network, it requires its physical layer definition to be described in an analogous way. As the approved IEEE standard no. 1076.1, also called VHDL-AMS, is the only standardized hardware description language for digital, analogue and mixed signals and systems, this language was the only choice by OEMs in defining standards for adequate models.

This step was necessary, because the experience in the early days of simulating fieldbus networks has shown that there were a lot of different and only proprietary device models. They were not fully reproducible in terms of their results. They were not interoperable with each other as well as their capabilities were inappropriate. Addtionally, most of the models were not covering their respective datasheets and finally they may not be designed nor at least being approved by the device developer or following commonly agreed criteria.

Another important side-effect is the VHDL-based encryption capability of the VHDL-AMS supporting simulators, because one of the future revisions of the VHDL-AMS standard will supporting it on its own based on the VHDL standard. This gives the device developer certain safety to protect their semiconductor intellectual property.

Because the OEMs don't need to know the basic physics and effects behind each single transistor like the semiconductor needs at the development of the device (if he uses model-based development), a reduced complexity can be assigned to the required models. This complexity was graded into different levels covering different investigation purposes. This also serves some benefits regarding stability and performance of the simulation, where we return to later on in this paper.

In addition to the models describing the topology of a fieldbus network, additional components are necessary for

n the creation of stimulation signals or waveforms,

n the acquisition of data, respectively the digital or analogue waveforms and finally

n waveform analysis and requirement checking.

The first item refers to power supply sources for the (energy-) conservative system descriptions as well as communication stimulation which is, in most of the cases, a round robin communication. Besides those, there are also some analogue or digital failure injections and operational mode switches.

The last two items are possible due to the capabilities of the used hardware description language. Also, they are necessary to gain a certain independence from specific simulator tool chains as they are in use by different customers.
Furthermore all of these components are configurable to specific test cases.

The test methodology

Hence, the full model description (referred to as a test bench) consists of a topology description and the environmental test system components. These environmental components are designed as general purpose to cover a defined test specification. Therefore they are needed to be configurable to specific test cases of a given test specification. This design choice leads to a fixed model framework with a parameter interface and fully qualifying test case configuration parameter sets. Finally, this defined package implements the precise requirement which should be checked with its specific test setup.

All previously described model components forming the test bench as a framework are structured in accordance with the local testing methodology defined in the ISO standard no. 9646. This method was the starting basis for the model descriptions for the automation of model-based signal integrity analyses.

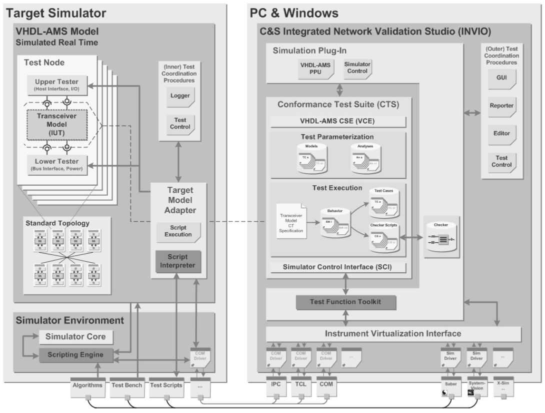

Figure 2: Implementation of local test method

The previously described features of the parameterizable, model-based test framework cover the requirements regarding reusability and, in conjunction with the semiconductor device models, reproducibility of the results – for the model part.

To gain coverage regarding the comparability of the real device measurements and results, the semiconductor device models need to be approved at first. Second, the measurement procedures, parasitic effects, and the points of observation are implemented as such, that they were the same as for the real device measurements from the technical point of view with a certain level of abstraction.

By the given parameter interface of the test bench framework the customer is allowed to vary topology component values and layout. With the requirement of independently described, approved, and interoperable semiconductor device models, the test bench framework only allows for the usage of transceiver models which are conformant to their respective transceiver model specification. Otherwise these models wouldn't be interchangeable and configurable via the parameter set of the test bench framework. This was an important rule to be defined right from the start of the development of an automation for the model-based signal integrity analysis.

The real automation –validation process

Up to now, we have only described the prerequisites for the automation regarding the model-related point of view. But the automation of model-based signal integrity analyses doesn't stop at the outer boundaries of the model descriptions. The automation has also to do with the execution of the simulator tool, the computation of the test bench framework model description and the most important thing, the integration into the customers' development process, beginning with the definition of the topology and ending with the result report. The following sections will give an overview about these items to finalise the idea and realisation of this automation.

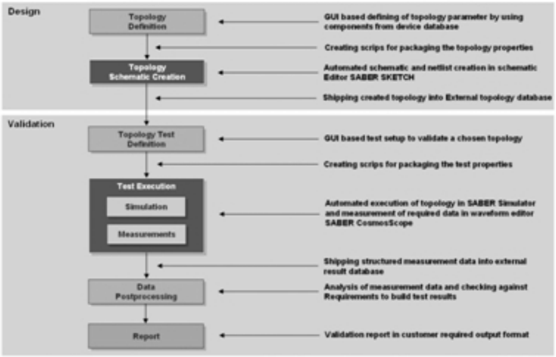
Design
Topology
Definition
GUI based defining of topology parameter by using components from device database
Creating scrips for packaging the topology properties
Topology
Schematic Creation
Automated schematic and netlist creation in schematic Editor SABER SKETCH
Shipping created topology into External topology database
Validation
Topology Test
Definition
GUI based test setup to validate a chosen topology
Creating scrips for packaging the test properties
Test Execution
Simulation
Measurements
Automated execution of topology in SABER Simulator and measurement of required data in waveform editor SABER CosmosScope
Shipping structured measurement data into external result database
Data
Postprocessing
Analysis of measurement data and checking against Requirements to build test results
Report
Validation report in customer required output format

Figure 3: Automation workflow

Building blocks – the topology definition and framework

The definition of the topology and therefore the transfer of information through its layout and ECU circuitries, as well as devices and network structure into an executable test bench (framework) stood for enormous personnel efforts and carried a huge risk of errors and massively reduced reproducibility as well as reusability. At this particular point our solution comes into play providing capabilities for the integration / transformation of topology definitions described in different OEM-specific ways based on their development process into executable model descriptions.

Based on the present simulator tool chains on the market there are several possible approaches to transfer any number of topologies into executable model descriptions within the limits of their respective field bus specifications.

The first approach uses the, somehow proprietary schematic editor of the tool chain. This provides the opportunity of creating a visual shape of the electrical circuits of the topology following a graphical routing algorithm developed by us. The approach has to

n cover all possible topology layouts allowed by the respective field bus specification, e.g. linear buses, number of passive or active stars, max. number of ECUs, etc.,

n assure semantically correct code generation, and

n offer a reasonable overview of the topology layout for the user.

It's advantage is to use the built-in model-code generator of the given tool chain. Also, another gain is the graphical image of the visual shapes as a side-product to transfer decisions within the development process to all participating persons in further topology design.

The disadvantages are, as mentioned before, the use of proprietary model code generation which may result into code descriptions not covered by other simulator tool chains as well as the complex graphical routing algorithm to support all possible topology layouts allowed by the respective field bus specifications. The schematic view is therefore modifiable only within strong limits given by the simulator tool chain.

The second approach embraces some features of the used hardware description language VHDL-AMS by using iterative and conditional generate statements for variable components and processes of the virtual test environment.

One benefit of this variant is the creation of schematic views of given topologies independent of the used simulator tool chain and thus, can be fully modified to customer's needs. Another advantage is the independent construction of the model test bench. This provides full control about the model description and its performance level and reduces also possible artifacts produced by other code generator tools.

Hence, the test bench framework can be used completely and without adoptions to use with the established simulator tool chains on the market.

The drawbacks of this approach are slightly more complex test bench framework and configuration interfaces as well as, at the current point in time, not all simulator tool chains support the full feature set and latest standard revisions of the hardware description language VHDL-AMS resulting in a subset to be used with the test bench framework which was identified in analyses carried out in the past. Following this, there were no real bottle necks in using the language over different simulator tool chains. Like with the first approach a graphical routing algorithm is needed to create the visual shapes of the circuit design of a given topology, but with more customisation more complexity is added to the algorithm.

After the topology and the surrounding test environment is completely defined in terms of a model description, this test bench framework can subsequently be executed with the simulator to obtain the required signal waveforms for the post-processing.

Home straight – from post-processing to the report

Besides the automated topology definition and framework creation the post-processing up to the report is the second source of time-consumption within a hand operated topology validation process and thus, needs to be automated.

The post-processing begins with the analysis of the acquired signal waveforms. This analysis involves the graphical displaying of the signals at first, which is required at a minimum to make hand operated measurements. The previously mentioned comparability between measurements acquired in real hardware setups and simulation results is assured using the same points of observation as well as being combined mathematically as for the real device measurements, e.g. using oscilloscope functions. By these facts it can be said that the manual effort grows dramatically with the increasing amount of topologies, ECUs and measurements that are being executed. And therefore an automation of these process steps has become a necessary requirement to the OEMs.

After gathering the measured values, it needs to be checked if they are within the defined limits in order to find a verdict either for a

n single measurement (referred to as an atomic result),

n group of measurements,

n group of ECUs,

n group of topologies or

n any other component or parameter variation or variation package

Finally, the gathered verdicts are stored into a report, which is important for further decisions within the development process of a topology and therefore has to share these relevant information and outcomes of the simulation in a way, which every involved developer and decision-makers can understand.

Such a report has usually a customer specific common look and feel has to be integrated into the customer's development processes in terms of output formats and style. This solution supports different output formats like

n database-based,

n spreadsheet-based,

n as formatted text files (with or without using markup languages),

n and others.

Conclusion

The model-based testing methodology is, as described in the previous sections, powered by a virtual and configurable test environment as a framework. It is made round by an automating and project based application to conduct the validation process of the signal integrity in different fieldbus networks.

Apart from that, this model-based testing methodology is only one part of the full solution developed so far. Completing the whole package of validating topologies, this solution doesn't stop at basic comparability in relation to requirements for real device measurements. It rather combines both the simulation and the real device measurements into one powerful solution, but with the main background of reducing required measurement efforts while having better testing coverage of topology layout and circuitry possibilities.

This circumstance serves a lot of benefits and, of course, covers the stated requirements to automate the topology validation tasks of different customers, which are:

n It extends the use of simulation by fully automated creation of test benches and test suites.

n It accompanies the development from the topology definition up to the validation report.

n It supports measurements either in hardware setups or in real vehicle prototypes with fully automated creation of test cases and test suites.

n It allows for comparison between simulation and real device measurements.

n It can be integrated into OEM-specific tool chains used for complex verification procedures at higher abstraction levels.

- It supports different field bus systems, like CAN and FlexRay. Others like LIN and Ethernet are planned to be covered in the future.
- It allows for configurable statistics over simulation and real measurements.

This finally gives the customer a:

- massive reduction of topology network validation running time and
- it assures test personnel- and equipment independent validation.

References

[1] *Requirement Specification for Transceiver Simulation Models V1.1* ICT/GIFT, 2010

[2] *Model Conformance Test Specification V1.1* ICT/GIFT, 2010

[3] *Road vehicles – Controller area network (CAN) Part 2, 5, 6* ISO 11898-2/-5/-6, 2003 - 2011

[4] *IEEE Standard VHDL Language Reference Manual* IEEE 1076, 2008

[5] *IEEE Standard VHDL Analog and Mixed-Signal Extensions* IEEE 1076.1, 1999

[6] *Information technology – Open Systems Interconnection – Conformance testing methodology and framework* ISO 9646, 1994

Marko Moch
C&S group GmbH
Am Exer 19B
D-38302 Wolfenbüttel
Phone: +49 5331/90555 0
Fax: +49 5331/90555 110
m.moch@cs-group.de
www.cs-group.de