*SAE TECHNICAL PAPER SERIES*

**2007-01-1636**

# Development of the Physical Layer and Signal Integrity Analysis of FlexRay™ Design Systems

**Thorsten Gerke**
Synopsys GmbH

**David Bollati**
C&S Group/University of Applied Science Wolfenbüttel

**Reprinted From: Simulation & Modeling Mechatronics (SP-2111)**

ISBN 0-7680-1633-9

9 780768 016338

**2007 World Congress**
**Detroit, Michigan**
**April 16-19, 2007**

SAE International™

400 Commonwealth Drive, Warrendale, PA 15096-0001 U.S.A. Tel: (724) 776-4841 Fax: (724) 776-0790 Web: www.sae.org


*All SAE papers, standards, and selected books are abstracted and indexed in the Global Mobility Database.*



**ISSN 0148-7191**


**Printed in USA**

2007-01-1636

# Development of the Physical Layer and Signal Integrity Analysis of FlexRay™ Design Systems

**Thorsten Gerke**
Synopsys GmbH

**David Bollati**
C&S Group/University of Applied Science Wolfenbüttel



## ABSTRACT

Future automotive applications, like high-speed control in power train or drive-by-wire systems, demand large bandwidth, deterministic communication behavior, and fault tolerance. FlexRay, a new standard communication system, is ideally suited to safety applications as well as applicable to the role of a central backbone in future ECU network architectures. The FlexRay physical layer specification is kept very generic to provide the network designer with a wide range of possibilities for optimization of the network implementation. Due to the highly transient behavior of the system, the developer of the network physical layer cannot manually predict the behavior of an entire FlexRay topology. To analyze design concepts like topologies, terminations, and ECU architectures much earlier in development phase, simulation is the only choice. Simulation can be used to predict physical behavior and to verify the physical layer implementation of a FlexRay network while accounting for component and environmental variations. Accordingly, the developer can use simulation in the design of a robust network to investigate the influence and interoperability of new components and ECU interfaces with the goal of improving quality in automotive networks. Using virtual prototypes or production networks in conjunction with Robust Design methods, the developer can analyze network extensions (e.g. through automated wire length variations) and verify the impact of device tolerances. This paper describes the required elements for simulating the FlexRay physical layer, including simulation models, simulation scenarios, and post-processing mechanisms needed to sufficiently evaluate system behavior.

## AUTOMOTIVE MARKET AND IN-VEHICLE NETWORKS

Automotive networks, also known as in-vehicle networks, are used to connect sensors, actuators, and ECUs (Electronic Control Units) together while minimizing the amount of wiring required. This allows for easy sharing of available information among ECUs without integration of redundant hardware like additional wiring harnesses and associated connectors. This also leads to more modular distributed electronics along with heterogeneous network architectures. Virtually every new vehicle platform contains multiple communication networks, where several different protocol types are applied to address the needs of individual applications. The main categories of applications and their related protocols can be divided as follows:

- Infotainment: MOST, D2B
- Powertrain: CAN, FlexRay
- Safety systems: FlexRay
- Body electronics: LIN, CAN

The costs of networked electronics in today's mid-size

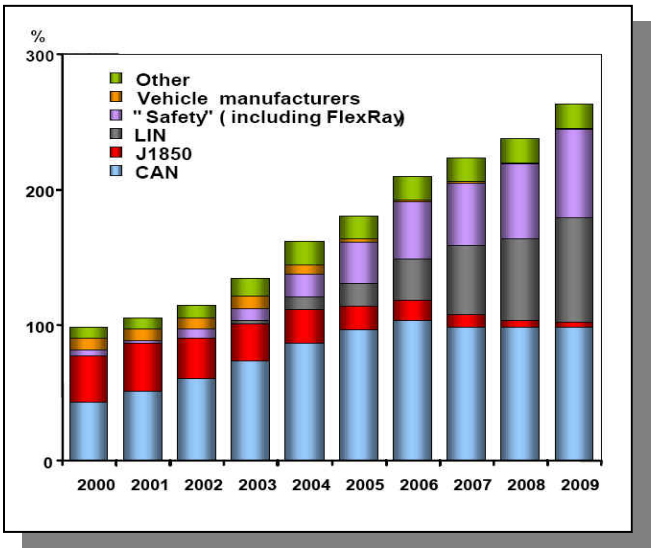


**Figure 1: Automotive bus systems on the world market (Source: Strategy Analysis)**

cars are already in the range of 40% of the total vehicle development costs. Figure 1 forecasts the increase of vehicle networking protocols used in automotive systems. The complexity of today's vehicle networks has been increasing over the past years and will continue to increase in the future as more content for safety and comfort is integrated into the vehicle. This requires a well-defined and robust in-vehicle network that must guarantee safe and correct data communication while being robust to internal and external influences. This is the challenge of the network engineer responsible for the embedded and physical layer of the network. Unlike the embedded world, the physical layer of in-vehicle networks does not have an ideal logical behavior. There is a significant analog behavior of the physical layer that must be taken into consideration. The dependencies between analog components creates a significant challenge in the design in a system with an infinite number of variants, as no manual computation or any analytical solution exists that describes the complete behavior of the analog network. The following section describes the challenges network engineers face when they deal with the development of the physical layer implementation of FlexRay designs and how this challenge can be addressed using system simulation.

## FUNDAMENTALS OF FLEXRAY

FlexRay was created at the end of the 1990's by Automotive OEMs and suppliers. It is intended for applications that need high speed data transfer and time-triggered communication, like closed-loop control systems requiring hard real-time performance. FlexRay allows a variety of topology types while taking into account the physical constraints that are present in a system. Figure 2 shows a possible topology consisting of eight nodes and two active stars. The local timing

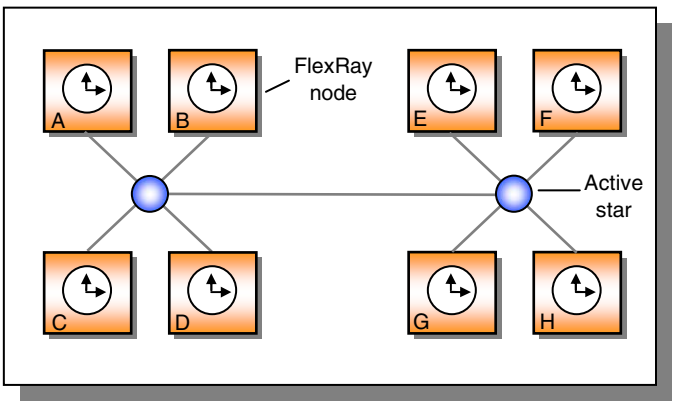


**Figure 2: Possible topology with two active stars according to FlexRay specification V2.1**

information of each node is synchronized through the FlexRay protocol. The communication cycle in FlexRay consists of several segments, as shown in Figure 3. All message frames in the static segment have the same length while the lengths of frames in the dynamic segment can be adjusted as needed. Every communication starts with a Transmission

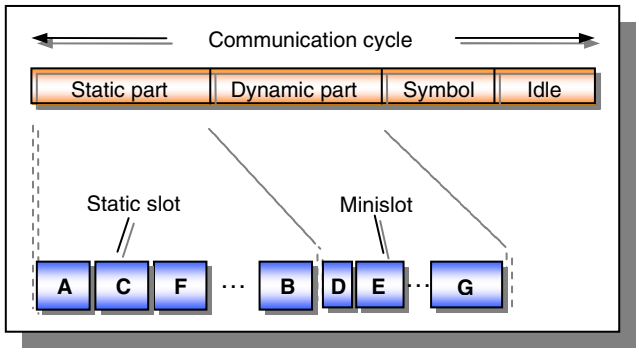


**Figure 3: FlexRay communication cycle**

Start Sequence (TSS) as shown in Figure 4. The TSS is a continuous low for a period defined for the network cluster and is used to open the gates of active stars and indicate the start of a FlexRay communication. Detailed information about the communication process is given in [1]. Each byte of the payload starts with the Byte Start Sequence (BSS), where the falling edge of the BSS is used to synchronize the local timing with the global time. The frames are terminated through the Frame End Sequence (FES). For frames inside the dynamic segment of the communication cycle, the FES is followed

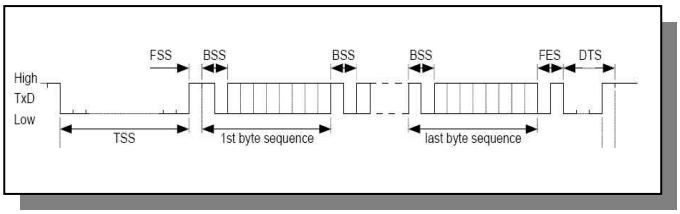


**Figure 4: FlexRay frame**

by the Dynamic Trailing Sequence (DTS), which is used to define the action point in the dynamic segment. In order to transmit signals across the network, a physical layer is needed to define how the logical states are represented by electrical voltages. The FlexRay Electrical Physical Layer (EPL) specification [2] describes the basic requirements for hardware used for this application, like bus drivers and active stars. In addition, the EPL specification gives rough recommendations on additional elements like common mode chokes and ESD protection components. The specification neither dictates nor prescribes the system implementation of the physical layer. In addition to the flexibility related to the choice of hardware components, FlexRay allows also a wide range of topology types. Possible topologies are:

- Point-to-Point
- Passive bus
- Passive star
- Active star

- Cascaded active stars
- Hybrid topologies
- Dual channel topologies (including redundant communication channel)

This flexibility provides the network developer the ability to optimize the entire network according to the needs of the application and the vehicle implementation. The selection of hardware components as well as topology type has a significant impact on the signal integrity of the entire system. As shown in Figure 5, bits transmitted by ECU A may arrive in a totally different shape at the receiving ECU F due to the impact of several elements in between each ECU. Items that impact the transmission of signals across the network and are necessary to take

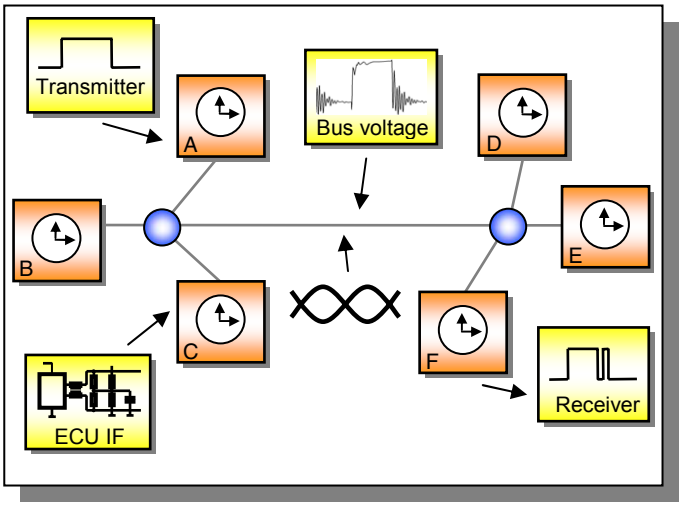


**Figure 5: Deformation of transmitted bits**

into consideration while developing the physical layer implementation include:

- Signal filters (e.g. chokes or ferrites)
- Active stars
- Transceiver
- Transmission line
- ESD protection elements
- Topology type
- Termination

While they allow great flexibility in the design, the developer is faced with the problem that the interaction of all these elements creates a system with analog behavior, and it cannot be determined a priori whether the implementation ensures well-defined signal integrity. Depending on the implementation, the system will exhibit different behavior related to circuit ringing and reflections. In addition, the system developer is also required to ensure sufficient immunity to RF injections.

## SIMULATION AS BASIS FOR ROBUST DESIGN

How does a network developer achieve and verify the requirements above and build up a running system that is sufficiently robust to environmental impacts? Development of prototypes takes too much time and is a very inflexible method when evaluating different network options. Simulation is the only choice when it comes to the development of high speed in-vehicle networks like FlexRay or CAN [4]. Simulation allows the creation of network design rules by investigating network limitations through analysis of worst cases and can be used to improve network quality without creating unneeded hardware prototypes. The result is higher quality and significantly reduced development time. Simulation also supports the education process of network designers. Past projects have indicated that network engineers who applied system simulation to the development process know and understand their implementation much better than those who do not, since system simulation allows them to study the electrical behavior to a greater depth than is possible with hardware prototypes alone.

There are several critical aspects of the signal integrity of a FlexRay network that drive requirements for system simulation models for the physical layer implementation. These items are:

- Signal propagation delay
- Asymmetric delay
- Bit deformation due to ringing and reflections
- Truncation of Transmission Start Sequence (transmission idle to busy)
- Frame stretching due to ringing after last frame bit (transition active to idle)

The signal propagation delay is the time lapse between falling edges of the transmitting and receiving node, as shown in Figure 6. This value has a significant impact on the performance of the FlexRay system related to the clock synchronization mechanism. The EPL dictates rules for the maximum propagation delay. Unfortunately, simply summing the propagation delays along the signal path between nodes does not work since effects like reflections impact this behavior significantly. The signal propagation delay depends on:

- Bus load
- Supply voltage
- Temperature

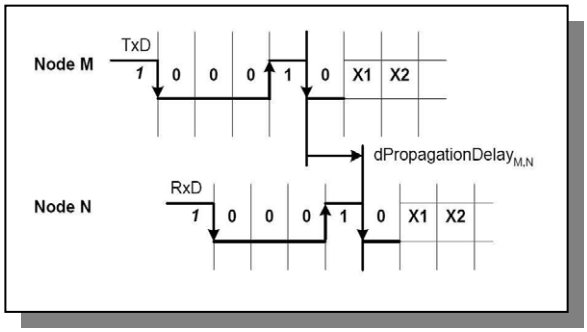


**Figure 6: Signal propagation delay**

The asymmetric delay describes how much the bit length of the original transmitted bit has changed when it arrives at the receiving ECU. This effect depends on a variety of items, as shown in Figure 7. Some of them have a static dependency, meaning that they are fixed during the communication cycle:

- Mismatches between propagation delays of negative and positive edges
- Hysteresis of common mode chokes
- Parasitic effects (e.g. capacitance coupling due to PCB)

Other effects show a stochastic behavior during the communication cycle:

- Edge Jitter
- Unbalanced behavior above ground

The asymmetric delays are important to be taken into consideration since the higher the asymmetry the smaller the robustness against injection of RF fields, thereby decreasing the reserved EMC budget. During the transmission from idle to active, the bit sequence of the TSS can be shortened. The network developer must ensure sufficient length for the TSS, which depends on network topology. The truncation is mainly caused by two items:

- Filter time for activity detection plus evaluation of internal logic in the receiver and active star
- Set up of transmitter

The FlexRay Protocol Specification defines rules for the length of the TSS, as related to the number of network nodes, clock deviation between transmitter and receiver, and transmission rate. The specification also dictates that at least one bit of the TSS must be left. The following sections will describe the required elements and simulation techniques for developing and validating the physical layer of a FlexRay network.

## DEVELOPMENT FLOW USING SABER'S ROBUST DESIGN METHOD

Based on the critical points that have been previously identified, a robust development flow for the conceptual validation of the desired physical layer implementation can be defined. As shown in Figure 8, the developer provides the information about topology, components, and their associated parameter values in conjunction with supply voltages and corresponding bit streams or bit configurations. Note that the purpose of this validation approach is not to analyze all possible bit configurations. Only those configurations that are deemed critical with respect to the physical layer should be taken into consideration. This information is used to build up the corresponding virtual prototype of the FlexRay network. The simulation environment then computes the transient behavior of the design. As a result, the developer receives information about all signals in the defined network with respect to time. This allows the developer to validate the critical aspects of the current implementation against the FlexRay specification and specific OEM requirements. If the design fails the virtual test, the system needs to be either redesigned or optimized. This may take several iterations, but since this is performed through system simulation, it takes a comparatively short time (with respect to hardware

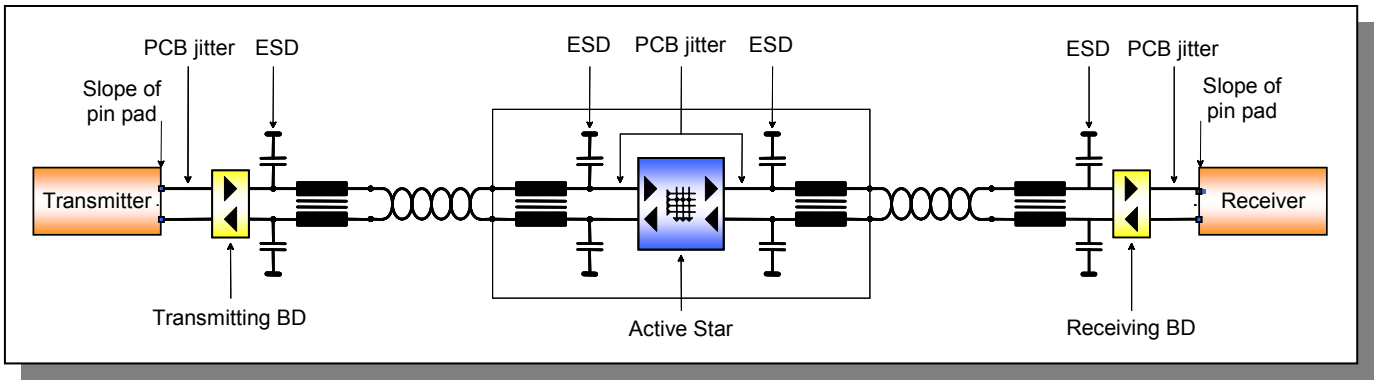


**Figure 7: Sources for the asymmetric delay**

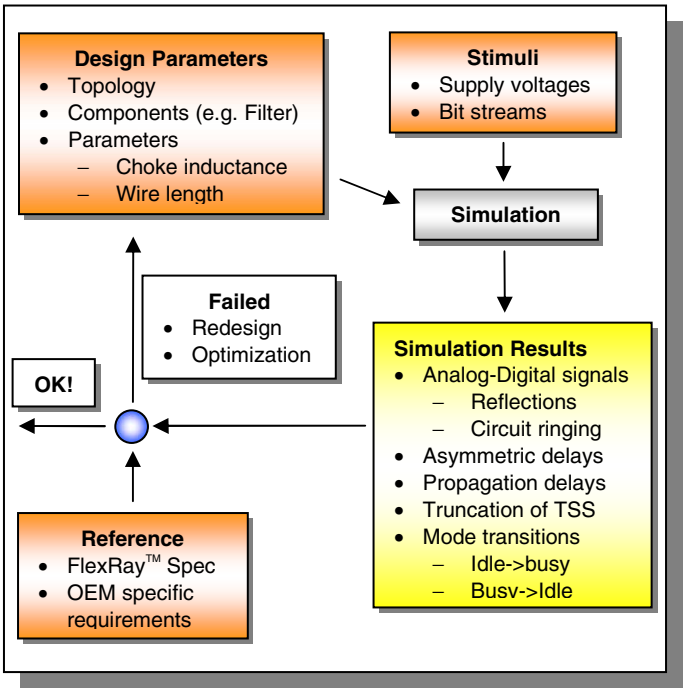


**Figure 8: Development and validation flow using a virtual FlexRay prototype network**

prototyping) to make any changes in the design, like modifying wire lengths or changing capacitances of HF filters. Even a complete redesign of the entire topology can be achieved quickly. Once the topology successfully passes the validation process, it is ensured that the design is well-defined from a signal integrity standpoint. Another important part of Saber's robust design methodology is that it takes component variation into account, e.g. tolerances associated with resistors that terminate the transmission line have a significant impact on the signal reflections. In addition, the tolerances of the bus driver thresholds impact significantly the values of asymmetric delays. An overview of how to apply advanced simulation techniques in Saber to validate the design (including component tolerances) is given in [6].

## SIMULATION MODELS

### Overall FlexRay Design

Since the validation flow and validation criteria have been defined, the corresponding simulation models can be derived from these requirements. At this stage, the purpose of simulation is to perform system-level analyses. Accordingly, the models required should address the needs of system simulation rather than the needs of component-level design. The models need to be accurate with respect to the questions that were mentioned in the previous section. At the same time, the simulation model must be optimized for simulation speed since comprehensive analysis is intended to be performed. Figure 9 shows the complete simulation model of a FlexRay physical layer implementation that includes a passive star with 6 nodes. Each ECU is connected through a transmission line to the network. In order to increase the robustness of the network against reflections, ferrites might be added as passive filter elements to the center of the star. The first topology variant example will not contain these filter elements, as it should be checked whether an implementation excluding ferrites shows sufficient quality with respect to signal integrity.

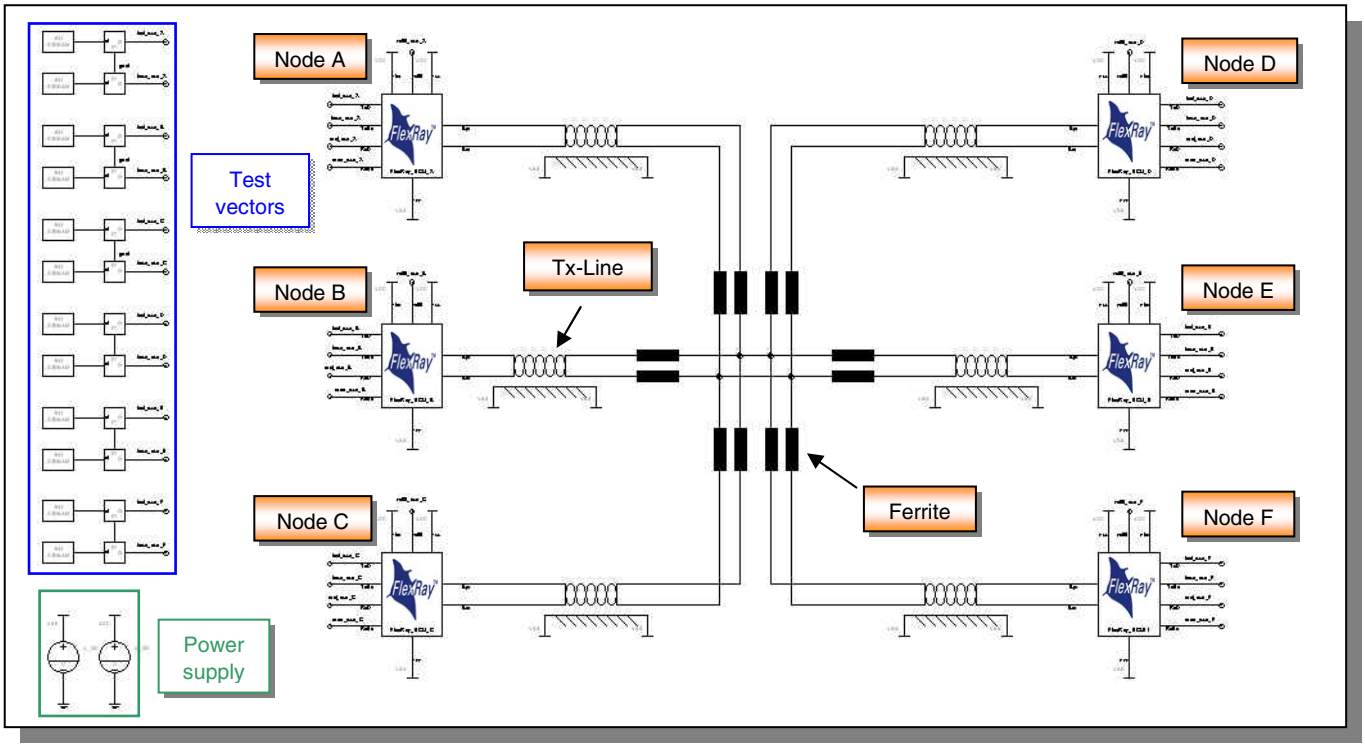


**Figure 9: Overall simulation model of a FlexRay passive star topology**

Modeling The FlexRay Node

The model of each FlexRay node, as shown in Figure 10, is a hierarchy containing:

- Transceiver (bus driver)
- Split Termination
- Common mode stabilization circuit
- Common mode choke
- ESD protection (capacitive behavior only)

The transceiver is a model that is delivered by the corresponding IC vendor. For this example, the TJA1080 from NXP Semiconductor (founded by Philips) has been chosen. NXP created a Saber MAST model for this

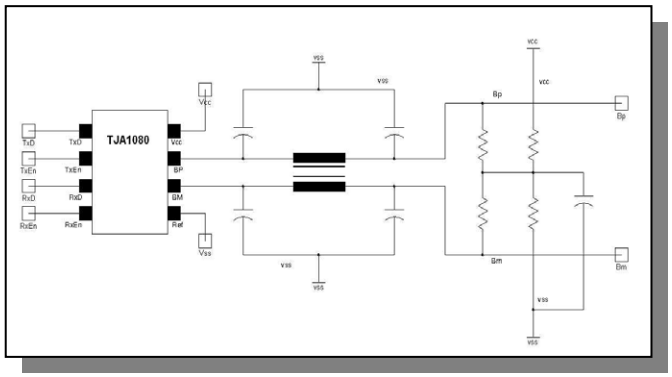


**Figure 10: Simulation model of a FlexRay node**

component consistent with the Saber transceiver modeling specification provided by DaimlerChrysler and Volkswagen [6]. Used by the Saber Simulator, MAST is a modeling language for describing Analog/Mixed-Signal components and multi-technology behaviors. The model is not the actual IC transistor level model. Instead, it has been created for the purpose of simulating complete systems, sacrificing some accuracy for speed of simulation. Philips is currently working on adding additional support to the model in order to cover other functional aspects of the transceiver like mode transitions, active star functionality and bus failure detection. The model was validated through measurements against a real system implementation [7].

Transmission Line Model

One of the most important parts of the simulation is the model of the transmission line. The requirements for the transmission line model have been put together by the FlexRay Physical Layer working group who is responsible for defining the FlexRay physical layer specification. Some of the requirements related to the model are:

- Wire length as a model argument to perform wire length variations
- Frequency dependent losses
- Support of both differential and common mode behavior

Saber's transmission line model addresses all of these requirements. The model equations are defined in the frequency domain to facilitate frequency-dependent

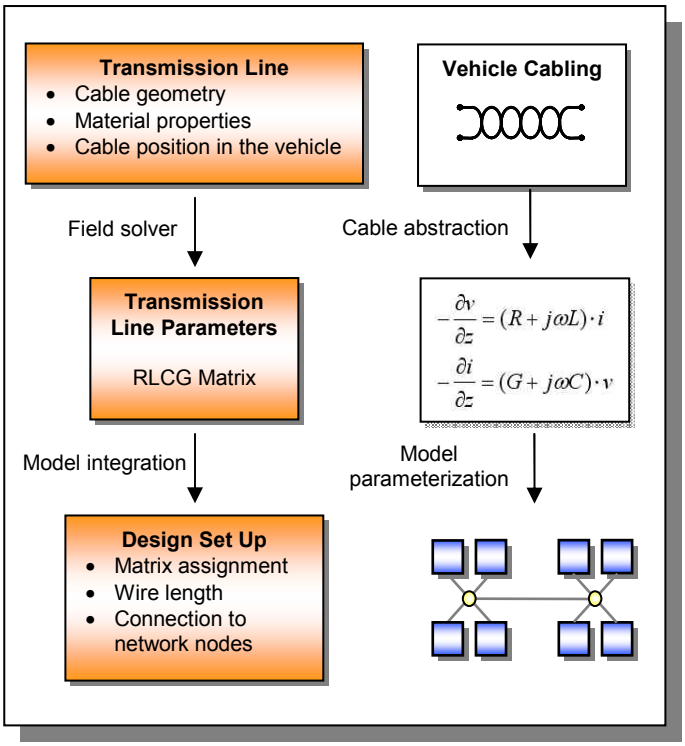


**Figure 11: Transmission line model**

effects and then applies a convolution algorithm to get back into the time domain. This method yields significantly better results compared to lumped element approaches which tend to unnecessarily oscillate and are difficult to adapt since the number of cells required for lumped wire models depends on the wire length. The characterization of the Saber model can be done through a field solver computing an RLCG matrix as shown in Figure 11. This approach enables the network engineer to analyze common mode as well as differential mode behavior.

## APPLICATION SCENARIO

In the following example, Round Robin communication is performed to analyze the signal integrity of the intended network topology shown in Figure 9. During Round Robin communication, each node in the network acts as transmitter once to discover potential communication issues relating to the actual communication procedure of the network. The setup for the network, including wire length and termination, is shown in Figure 12. According to

the FlexRay EPL application notes [3], the nodes with the largest distance between each other are terminated through a low impedance split termination. All other nodes use a high impedance termination. The wire length configuration is consistent with the defined limits since the EPL Specification states that the maximum

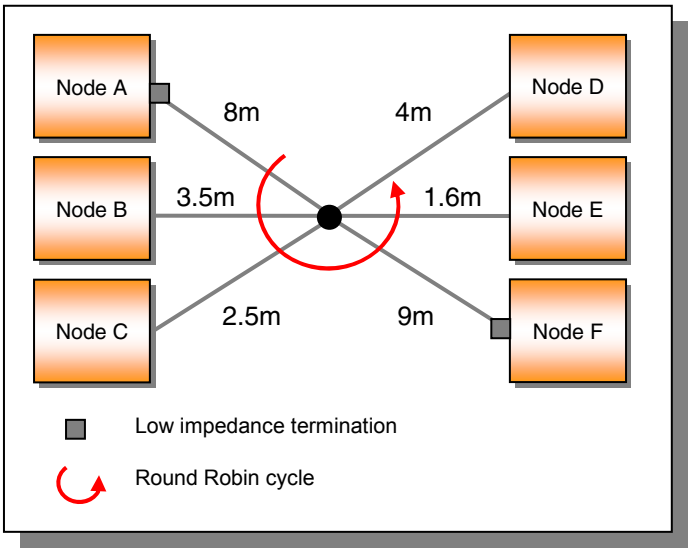


**Figure 12: Round Robin Communication**

distance between two nodes should not exceed 24m. It should be noted, however, that the specification does not guarantee that this topology will work from an electrical physical perspective. All nodes will start their communication by enabling the TxEn pin of the transceiver which changes the bus status from idle to busy and starts the transmission of a bit stream (synchronized with TxD) to analyze a specific bit configuration. A transmission rate of 10 MBaud is applied, resulting in a bit length of 100 ns. The first 10 low bits represent the TSS. This allows the developer to check, e.g., how much the TSS is truncated in this topology and how many reflections happen during the transition from idle to busy (which may cause additional

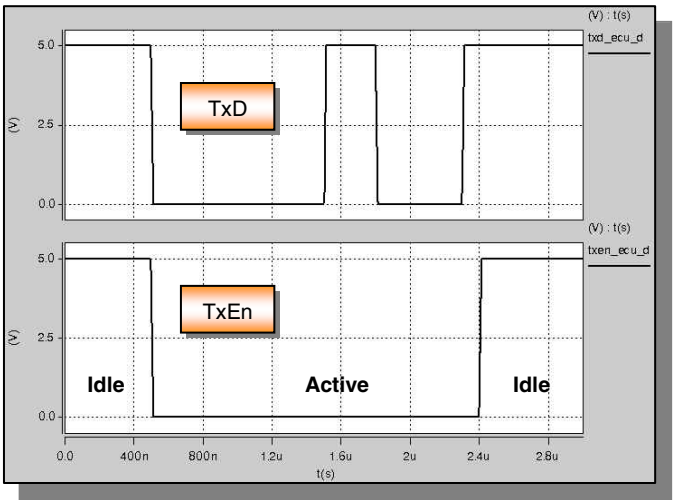


**Figure 13: TxD and TxEn signal set up**

truncation of TSS). The protocol specification allows a minimum of 6 bits for the TSS if no active stars are included in the topology. The following 3 High bits and 4 Low bits represent possible bits that may be part of a data byte of the frame payload. The bit stream is terminated with the FES, consisting of a consecutive sequence of one single Low bit and one High bit. In order to analyze the transition from active to idle, the TxEn pin is switched from Low to High immediately after the FES. This area is very important to be analyzed since it can cause significant oscillations in the analog bus signal as the bus driver goes over into a high impedance state. Neither the protocol specification nor the EPL provide guidelines on how much reserve must be added at the end of the frame to ensure that the idle state is correctly recognized. This may cause a slot boundary violation in the static segment of the FlexRay frame or issues during the DTS in the dynamic segment. Figure 14 shows the results of the first simulation scenario when ECU D is acting as transmitter. The signal RxD represents the digital receive pin of the FlexRay transceiver, and the signal uBus is the differential signal between the transceiver's BP and BM pins connected to

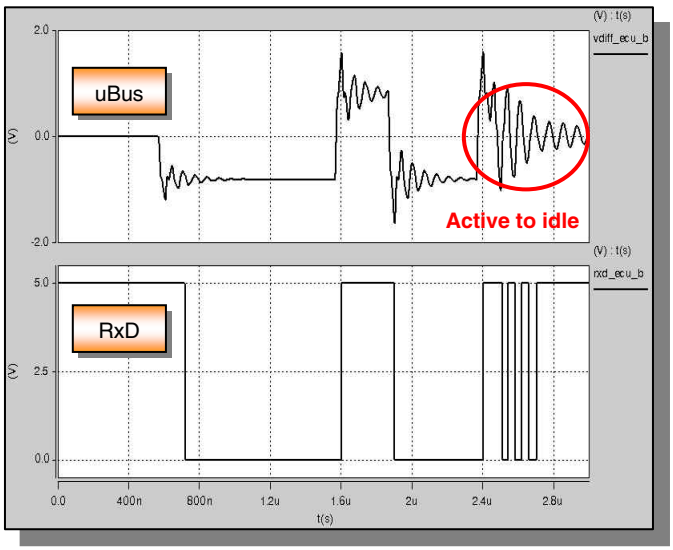


**Figure 14: RxD and differential bus signal ECU B (Transmitter ECU D)**

the bus. It can be seen that significant oscillations are associated with ECU B during the transition from active to idle even though the signal behavior during the busy state is fine. These reflections during the transition from active to idle cause repeated switching on the digital RxD pin. This ringing effect must be observed carefully since it affects the time needed to recognize the idle state of the bus. This is due to the fact that the FlexRay protocol controller waits until it has detected 11 consecutive High bits to determine that the bus is idle. This problem can be solved on the software side where the network developer must consider a sufficient safety margin when the frame length is defined. This of course causes additional frame stretching and can become an issue when the ringing exceeds a certain period of time and causes e.g. a slot

boundary violation in the static segment or arbitration issues in the dynamic segment. In the same scenario, the behavior at the low impedance terminated nodes as shown in figure 15 appears to be fine. The behavior at ECU E when ECU C is acting as transmitter should also be considered. Figure 16 shows the corresponding

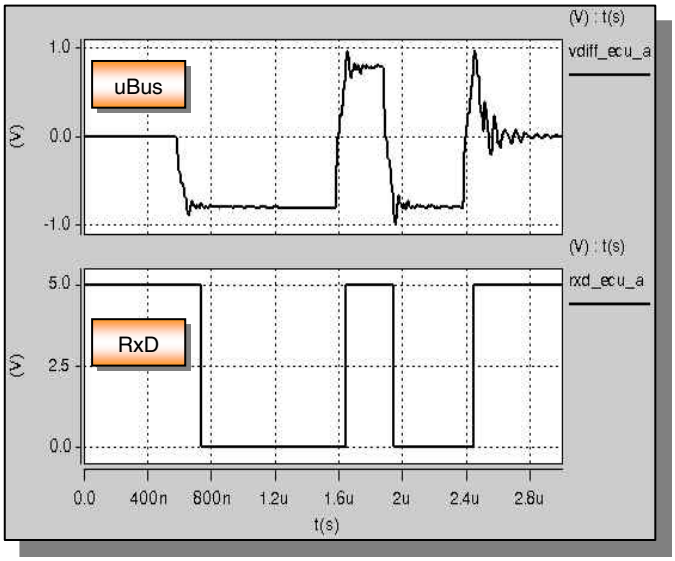


**Figure 15: RxD and differential bus signal ECU A (Transmitter ECU C)**

signals at ECU E. The signal shows reflections while the bus is busy during the transition from Low to High and vice versa. The magnitudes of these voltage peaks are 161mV and 192mV respectively. This is still below the input threshold of the transceiver for the nominal case (±225mV) but taking into account that the threshold is ±150mV in worst case due to tolerances of the receiver stage,

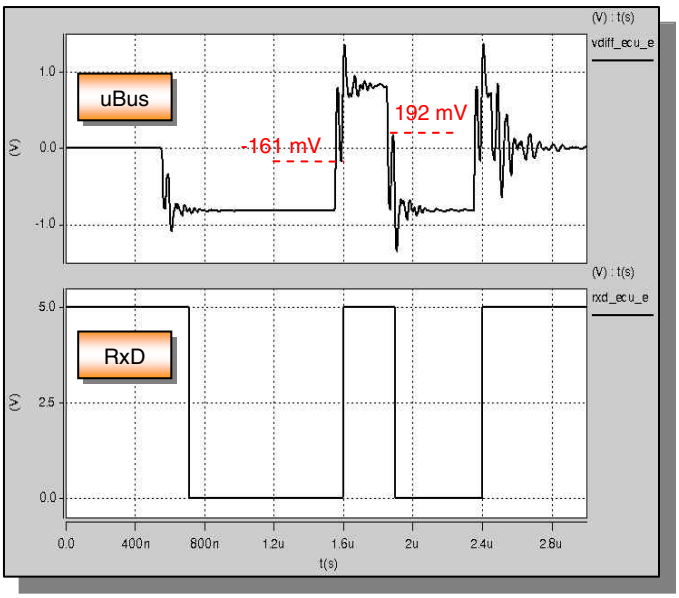


**Figure 16: RxD and differential bus signal ECU E (Transmitter ECU C)**

this may result in multiple switching of the Rxd signal of ECU E. Therefore, it is necessary to damp this signal behavior to ensure a more robust implementation. For the evaluation of the entire system behavior, this analysis must be performed for each single ECU, and the network developer has to validate the complete implementation after introducing any changes. The current implementation would be unacceptable due to the very problematic transition from active to idle. Modifications to the design are needed in order to improve the system behavior from a signal integrity point of view. In order to damp oscillations due to reflections, ferrites are often applied since they damp the reflections in the low impedance center of the passive star. The next implementation contains ferrite cores as passive damping elements (shown in Figure 17) and keeps the same wire length configuration as before.

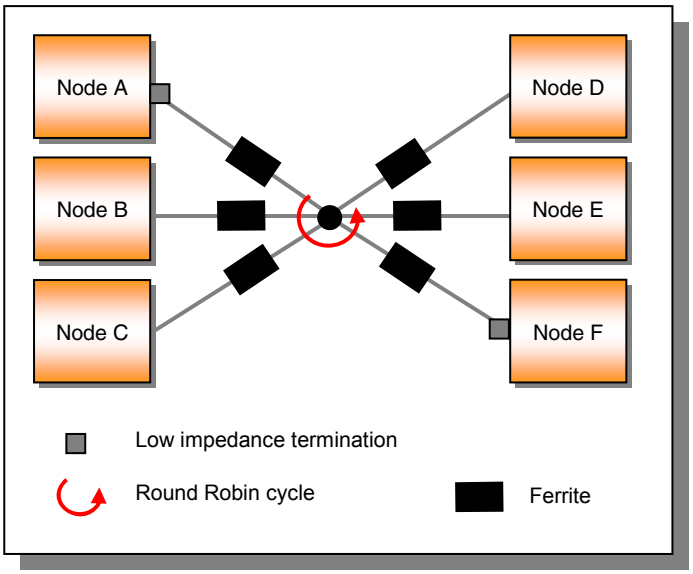


**Figure 17: Topology applying ferrite filter**

The same simulation scenario is applied to the design. Figure 18 shows results of the adapted topology. It can be seen that the behavior of the entire topology has been significantly improved by applying additional passive elements. The Rxd signal of ECU B shows much fewer oscillations during the transition from active to idle than before. Still, there is some ringing in the circuit that causes undesired switching of the Rxd signal at ECU B. Two possibilities for dealing with this problem are either to apply ferrite cores with larger signal attenuation or to analyze whether this problem can be handled on the software side. One of the advanced analysis capabilities of Saber is parametric variation. This allows the designer to vary e.g. the inductance value of the ferrites to validate whether a larger ferrite core helps to sufficiently filter the undesired reflections during the transition from active to idle. Figure 19 shows the results of the parametric analysis. It is obvious that a larger ferrite damps the magnitude of the oscillations better but even a ferrite of 340nH cannot completely remove the reflections. It is up to the network developer to decide whether a smaller

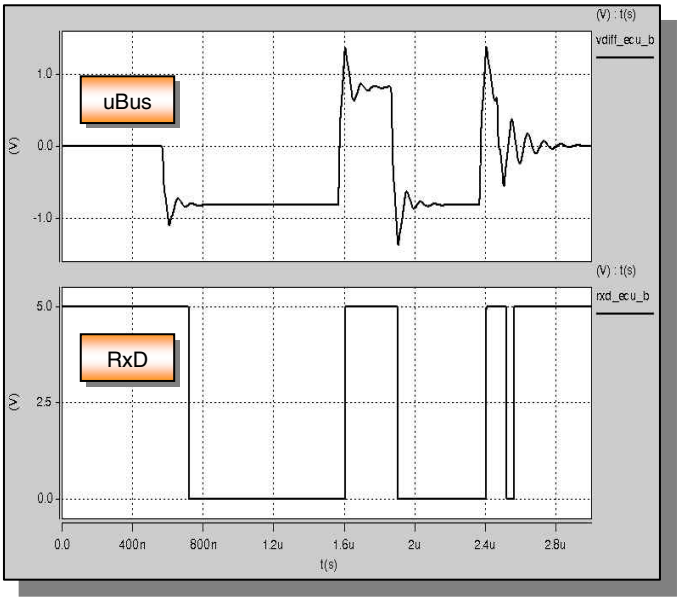


**Figure 18: RxD and differential bus signal ECU B (Transmitter ECU D) using ferrites**

ferrite can be applied, while carefully considering the ringing phase during transition from active to idle for the network and frame configuration or looking for other alternatives to optimize the system behavior. An option is for the software engineer to add a certain reserve at the end of the frame to ensure that the ECU is able to detect the network idle state without causing a slot boundary violation. This is the case for some of the other ECUs which have not been shown here. For this scenario, the

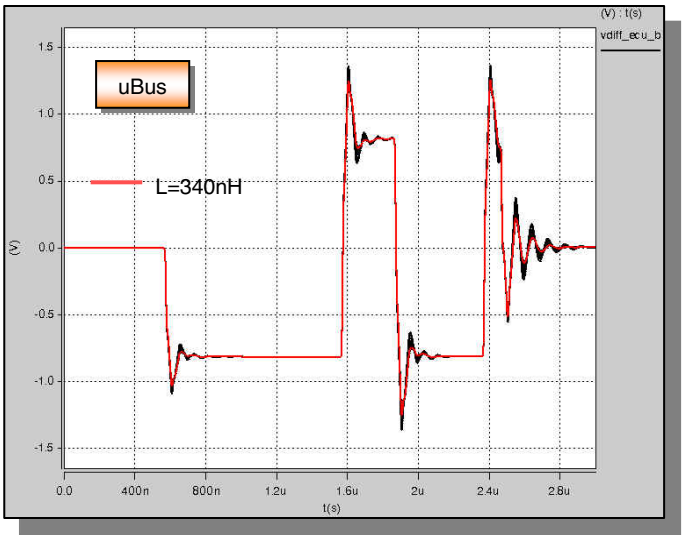


**Figure 19: Variation analysis and differential bus signal ECU B (Transmitter ECU D)**

smaller ferrite is going to be applied. Referring to the previous problem related to ECU E, the ferrites help to remove the undesirable behavior and there is now enough safety margin during the transitions of the logical bus states. Figure 20 shows the results after applying ferrites. It should be noted that this analysis

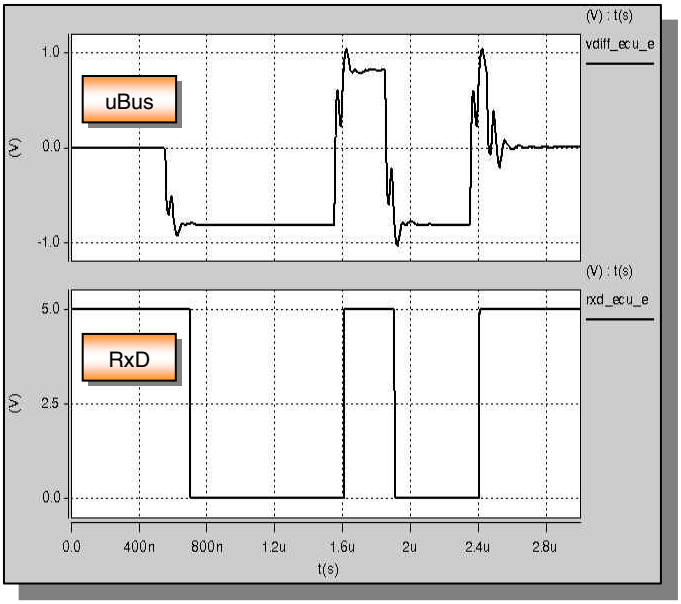


**Figure 20: RxD and differential bus signal ECU E (Transmitter ECU C) using ferrites**

was performed for all nodes in the passive star network.

Simulation can be used to further investigate the performance of the FlexRay network configuration. In order to guarantee that the right bit values are being sampled, it must be ensured that the bit length does not

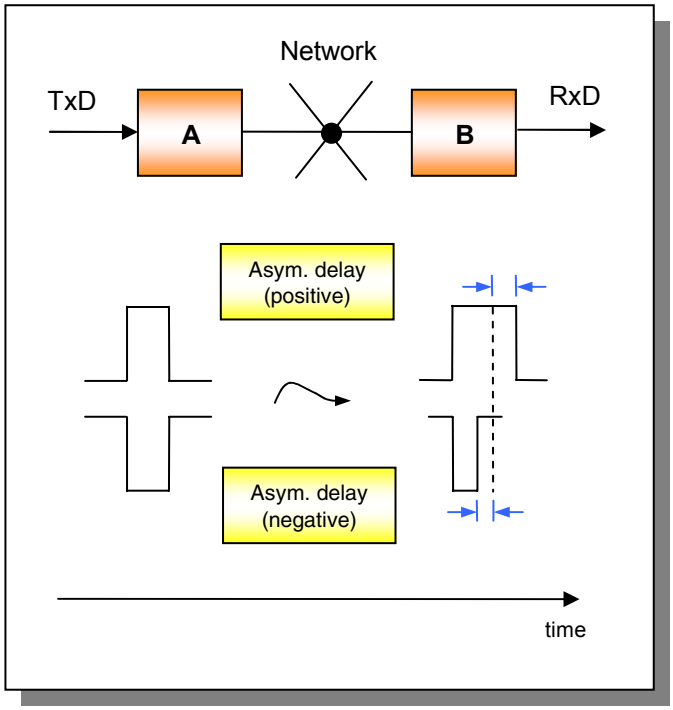


**Figure 21: Bit deformation due to asymmetric delays in the physical layer**

get too corrupted through the transmission across the network. This requires the network developer to validate the asymmetric delays encountered at each individual ECU. In order to analyze this, a single bit needs to be

transmitted across the network to check how much corruption is related to the physical layer depending on the signal path. As shown in Figure 21, the asymmetric delay is defined as the difference between actual bit time measured at the transceiver's digital receive pin and the targeted bit time represented by an ideal bit measured at the transmitter's TxD pin. The asymmetric delay will be determined for both the High bit and Low bit cases. The FlexRay specification defines limits for the asymmetric delay taking into consideration asymmetric delay due to edge jitter as well as physical layer issues. The maximum allowable asymmetric delay is ±30.75 ns for the complete signal path (e.g. from node A to node B shown in Figure 19). The developer should also take into account some reserve budget for asymmetries due to RF injection or any other uncertainty in the design, as explained in [2]. The physical layer specification does not clearly say how much asymmetry due to the physical layer is allowed for the single bit test without taking into account any RF injection issues, but a value of around ±9 ns is generally accepted as reasonable and ensures a budget of ±21.75 ns against other uncertainties in the design. Tables 1 and 2 show the simulation results

| | | Receiver (Asymmetric delay in ns) | | | | | |
|---|---|---|---|---|---|---|---|
| | | A | B | C | D | E | F |
| Transmitter (Bit length=100ns) | A | -6.1 | 2.4 | 4.9 | 2.4 | 0.5 | 0.5 |
| | B | -3.2 | -2.9 | -1.0 | -2.8 | -3.2 | -3.4 |
| | C | -2.5 | -0.1 | -2.3 | -2.2 | -2.0 | -2.5 |
| | D | -3.1 | -1.4 | -1.8 | 2.6 | -3.7 | -3.1 |
| | E | 2.2 | 0.4 | -1.7 | -2.3 | -2.1 | -2.3 |
| | F | 0.4 | 2.5 | 4.3 | 2.5 | 0.5 | -4.5 |

**Table 1: Asymmetric delays for single bit test "High Bit"**

concerning the asymmetric delays. The results show that there is no problem with respect to the asymmetric delays since all configurations are within the desired limit and there is enough budget remaining to cover eventual RF injection issues. The evaluation of the design has shown that all criteria are now met and the design is ready to be implemented. Even though the original intended implementation did not sufficiently ensure acceptable signal integrity, it was possible to improve the design by adding passive damping elements.

A complete validation of the implementation also requires the validation of the other criteria mentioned in the overview of the overall validation flow. More than that, component tolerances must be taken into consideration to check variations of the implementation

| | | Receiver (Asymmetric delay in ns) | | | | | |
|---|---|---|---|---|---|---|---|
| | | A | B | C | D | E | F |
| Transmitter (Bit length=100ns) | A | -5.2 | 3.4 | 5.9 | 3.5 | 1.5 | 1.6 |
| | B | -2.3 | -1.9 | 0.0 | -1.8 | -2.3 | -2.4 |
| | C | -1.5 | 0.8 | -1.3 | -1.2 | -1.0 | -1.6 |
| | D | -2.0 | -0.3 | -0.9 | -1.6 | -2.7 | -2.1 |
| | E | -1.2 | 1.4 | -1.7 | -1.2 | -1.1 | -1.3 |
| | F | 1.3 | 3.5 | 5.3 | 3.5 | 1.6 | -3.0 |

**Table 2: Asymmetric delays for single bit test "Low Bit"**

related to manufacturing or environmental variations, but this is out of the scope of this paper. Applying this verification methodology to FlexRay design systems through Monte Carlo analysis is shown in [7].

## CONCLUSION

FlexRay is a powerful networking protocol that allows for a wide range of possibilities in automotive design. Given the flexibility offered by FlexRay, the network developer is still required to check each individual implementation with respect to its physical layer implementation in order to predict the impact on signal integrity of the system. This is caused due to the highly transient behavior of the analog network part. Simulation is the only choice in order to very quickly obtain accurate validation results to evaluate possible vehicle communication architectures with respect to quality of the targeted topology. Simulation results can be achieved before the first hardware prototype is created, and even if the hardware is available, simulation allows much easier evaluation of the system behavior as component options and parameter variations can be easily simulated to validate different termination options and filter techniques. Vehicle manufacturers as well as component suppliers have recognized the importance of system simulation with respect to FlexRay and have decided to closely work together with tool vendors in the FlexRay simulation task force.

## REFERENCES


1. FlexRay Protocol Specification V2.1 Revision A, FlexRay Consortium 12/2005

2. FlexRay Electrical Physical Layer Specification V2.1 Revision B, FlexRay Consortium 11/2006
3. FlexRay Electrical Physical Layer Specification V2.1 Revision B, Application Notes, FlexRay Consortium 11/2006

4. Development and Verification of In-Vehicle Networks in a Virtual Environment, Thorsten Gerke (Synopsys GmbH), Carsten Schanze (Volkswagen AG), SAE World Congress, Detroit, 2005

5. Physical Layer - Basis of an Automotive Bus using the Example of FlexRay , Prof. Jürgen Minuth (University of Applied Science Esslingen), EESystems Conference, Unterhaching/Munich, 2006

6. Requirements Specification for Transceiver Simulation Models, Carsten Schanze (Volkswagen AG), Harald Bayer, Dietmar Ostowski (DaimlerChrysler AG), 2006

7. Monte Carlo Simulation for Validation of first FlexRay Series Networks, Thomas Suermann (NXP Semiconductors), FlexRay Product Day, Böblingen, 2006

8. TJA1080 Application Notes, Philips Semiconductor

## CONTACT

**Thorsten Gerke**
Application Consultant
Member of FlexRay Simulation Task Force
Synopsys GmbH
Karl-Hammer-Schmidtstr. 34
85609 Aschheim/Dornach (Germany)
E-Mail: thorsten.gerke@synopsys.com
Phone: +49 89 99320 227

**David Bollati**
Network Engineer & Project Manager
Chairman of FlexRay Simulation Task Force
C&S Group/University of Applied Science Wolfenbüttel
Salzdahlumerstr. 46/48
38302 Wolfenbüttel (Germany)
E-Mail: D.Bollati@cs-group.de
Phone: +49 5331 939 6623