ISSN 1726-5749

SENSORS & TRANSDUCERS

vol. 8 Special 2/10

Intelligent Data Acquisition and Sensors Systems

International Frequency Sensor Association Publishing

IFSA

Sensors & Transducers

Volume 8, Special Issue, February 2010 **www.sensorsportal.com** **ISSN 1726-5479**

Sensors & Transducers Journal (ISSN 1726-5479) is a peer review international journal published monthly online by International Frequency Sensor Association (IFSA).
Available in electronic and on CD.

Sensors & Transducers Journal

Contents

Volume 8
Special Issue
February 2010

www.sensorsportal.com

ISSN 1726-5479

Research Articles

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com
Please visit journal's webpage with preparation instructions: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm

SENSORDEVICES 2010:

The First International Conference on Sensor Device Technologies and Applications

July 18 - 25, 2010 - Venice, Italy

The inaugural event SENSORDEVICES 2010, The First International Conference on Sensor Device Technologies and Applications, initiates a series of events focusing on sensor devices themselves, the technology-capturing style of sensors, special technologies, signal control and interfaces, and particularly sensors-oriented applications. The evolution of the nano- and microtechnologies, nanomaterials, and the new business services make the sensor device industry and research on sensor-themselves very challenging.

Conference tracks

Sensor devices
Sensor device technologies
Sensors signal conditioning and interfacing circuits
Medical devices and sensors applications
Sensors domain-oriented devices, technologies, and applications
Sensor-based localization and tracking technologies

Important dates

Submission (full paper): February 20, 2010
Notification: March 25, 2010
Registration: April 15, 2010
Camera ready: April 20, 2010

IFSA
www.sensorsportal.com

http://www.iaria.org/conferences2010/SENSORDEVICES10.html

CENICS 2010:

The Third International Conference on Advances in Circuits, Electronics and Micro-electronics

July 18 - 25, 2010 - Venice, Italy

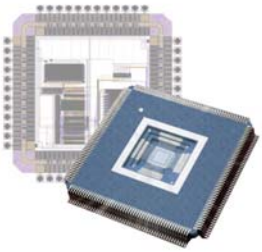
CENICS 2010 continues a series of events initiated in 2008, capturing the advances on special circuits, electronics, and micro-electronics on both theory and practice, from fabrication to applications using these special circuits and systems. The tracks cover fundamentals of design and implementation, techniques for deployment in various applications, and advances in signal processing. The topics suggested can be discussed in term of concepts, state of the art, research, standards, implementations, running experiments, applications, and industrial case studies.

Conference tracks

Semiconductors and applications
Design, models and languages
Signal processing circuits
Arithmetic computational circuits
Microelectronics
Electronics technologies
Special circuits
Consumer electronics
Application-oriented electronics

Important dates

Submission (full paper): February 20, 2010
Notification: March 25, 2010
Registration: April 15, 2010
Camera ready: April 20, 2010



http://www.iaria.org/conferences2010/CENICS10.html

Sensors & Transducers Journal, Vol. 8, Special Issue, February 2010, pp. 54-64

www.sensorsportal.com

Sensors & Transducers

ISSN 1726-5479

http://www.sensorsportal.com

Testing of CAN Based Automotive Distributed Systems Using a Flexible Set of IP Functions

Jiří NOVÁK
Czech Technical University in Prague, FEE, Department of Measurement,
Technická 2, Prague 6, 166 27 Czech Republic
Tel.: +420 224352807
E-mail: jnovak@fel.cvut.cz

Received: 28 January 2010 /Accepted: 15 February 2010 /Published: 26 February 2010

Abstract: An increasing number of Electronic Control Units in passenger cars and high communication traffic have led the car manufacturers to splitting a single CAN network into several dedicated sub-networks running concurrently. Test and evaluation tools for CAN based networks are available for years, but they are mostly focused on test scenarios with up to two CAN networks, often based on the standard CAN hardware with a limited test functionality. They usually rely on the extensive support provided by a PC running a Windows based application software, which in many cases limits accurate test timing, especially when real-time response on the bus events is required. This paper presents a set of three unique instruments (CAN generator, CAN trigger unit, CAN controller) designed in form of IP functions. Arbitrary number of the IP functions can be used in FPGA based test tools supporting real-time test scenarios over "unlimited" number of CAN sub-networks simultaneously. *Copyright © 2010 IFSA.*

Keywords: Vehicle, CAN, Communication testing, IP function

1. Introduction

Each new generation of vehicles (and especially passenger cars) uses more complex electrical and electronic equipment. To ensure the required functionality, flexibility and low price simultaneously, the distributed system concept was adopted to provide the communication path among the electronic subsystems. Controller Area Network (CAN) standard is widely adopted for in-vehicle distributed systems for years. An example provided in [1] declares 32 Electronic Control Units (ECUs) and 12 CAN networks in a single truck. Electronic systems play an important role also from the safety point

of view and therefore their correct behavior is required under the all possible working conditions. As the communication among ECUs is an important way how to receive and/or transmit information and as the communication subsystem often becomes a part of distributed control loops, testing the communication parameters and evaluation of ECU's reaction on communication issues is an extremely important task during the vehicle development.

2. State of the Art

According to the development phase in which they are applied, ECU's behavior tests can be divided into several groups. The first one is the functional testing, focused on the accurate acquisition of input data and real-time evaluation of correct outputs, used either directly for subsystem control or as the inputs for other components of the distributed system. Within this phase both the ECU hardware and software tests are applied, usually the HIL/SIL (Hardware/Software in the loop) methods are used. In case the ECU under test is a part of distributed system (nearly always), the testing of communication channel parameters and its behavior under a variety of conditions has a crucial importance, as an error in a single device can be a source of the system failure. In the second development phase the complete systems are tested in laboratory conditions, including the communication paths among ECUs. In this phase the ECUs' communication interoperability is tested as well as the correct timing and content of particular messages under the variable conditions, communication error recovery times, diagnostics functions, emergency modes and so on. The third testing phase takes place on pre-production cars and is focused on system level interoperability. The communication tests are focused mostly on the communication reliability under all possible conditions and identification of possible communication problem sources.

Several communication standards are currently used to implement distributed systems in vehicles (e.g. CAN, LIN, MOST). Each of them has its application area – LIN is used for low-speed communication especially in vehicle comfort subsystem, MOST provides the high-speed communication path for telematic, multimedia and similar services, and CAN is being used for medium-speed communication required by power-train (ECUs for engine, brakes, airbag, gear and so on) as well as for communication in comfort and infotainment subsystems. The CAN communication is therefore the most important from the point of view of vehicle functionality and safety and the behavior of ECUs at the CAN interface has to be tested properly.

There exist a lot of dedicated instruments on the market that are available for CAN communication monitoring and testing. Their range starts with elementary devices used for the simple transmission and reception of CAN frames and ends with sophisticated CAN emulators supporting more buses simultaneously with an exact timing, available scripting and so on. Nevertheless, these systems have together two important weak points. The first one comes from the fact that they are mostly based on standard CAN controllers (often SJA1000 from Infineon), that functionality for in-deep diagnostics is limited, especially when some non-standard behavior is required. The second weakness of these systems is either impossible or very difficult way of their mutual synchronization. Usually each system is based on its own time base and it doesn't provide flexible trigger-in/trigger-out subsystem.

The set of IP functions, described later at this paper, solves both these issues and can easily be used to setup the test beds for all three test-phases mentioned above.

3. Controller Area Network

The CAN standard was developed at Bosch (Germany) in Eighties and its detailed specification can be found in [2] or [3]. The following description is restricted to the attributes and features that are necessary for reader to understand the text below.

Because of the medium access control (MAC) method, the CAN standard requires a special implementation of the physical protocol layer. There are two states – dominant and recessive – that represent logical 0 and logical 1. If any node on the bus transmits logical 0, there is the dominant state on the bus, regardless of the levels transmitted by the other nodes. The recessive level is found on the bus only if all the nodes transmit logical 1. This mechanism is used especially in an arbitration field of CAN frames (which is used to send the frame identifier) in order to detect possible collisions and to resolve them. The principle is very simple. If two nodes start transmitting simultaneously, there is at least one bit in the arbitration field in which they differ (it is not allowed for two or more nodes within the system to use the same frame identifier). The first node sends this bit e.g. as the recessive bit (logical 1), the second as the dominant one (logical 0). According to the rule defined above there is a dominant level on the bus and the first node thus detects the collision. This collision is not destructive, as the first node stops transmitting and the second carries on sending the frame. This implementation of the MAC sub-layer is therefore called CSMA/CR (Carrier Sense Multiple Access with Collision Resolution).

Another important feature of the CAN protocol is a detection of errors in the received data. There are together four detection mechanisms. Each bit transmitted on the bus is received back by the transmitting node and the values are compared. If there is the difference, the bit error is indicated. Each frame is also ensured by CRC code. If the received CRC differs from the computed one, the CRC error is indicated. Each frame also contains several bits with predefined values. If any of them has the opposite value, the frame format error is indicated. Next, each transmitted frame must be acknowledged (from receiving nodes) by setting an acknowledgement bit into the dominant state (logical 0). If it is not set, the acknowledgement error is indicated.

In order to keep the receiver synchronization, the bit stuffing mechanism is implemented. When more than five bits of the same value have to be sent, one bit with the opposite value is inserted at the transmitter site and, of course, removed at the receiver site. If more than 5 consecutive bits of the same value are received, the bit stuffing error is indicated. Indication of any type of error leads the node that detected the error to generation of an error flag (six consecutive bits of the same logical level), which is encountered by all the nodes within the system and the data consistency is thus kept.

As the frames are broadcasted to the network (to all the nodes in the same time), one node, which encounters reception problems, can therefore block entire communication by permanent error flag generation. This is avoided by implementation of error level states. Only the error active node can actively generate error flags (using 6 consecutive dominant bits), error passive nodes can generate only passive error flags (using 6 consecutive recessive bits) and bus off nodes are totally disconnected from transmitting to the bus. The transition among these states depends on the detected error rate within the transmission and reception phases.

4. General System Description

The main idea of the proposed solution is to define and implement the set of the basic blocks (IP functions) for CAN based ECUs and systems testing. These blocks provide compatible triggering input/output subsystem, which allows for hardware synchronization among particular blocks as well as the event time stamping derived from the single time base. Another advantage, which is inherent for

FPGA based solutions, is the possibility to clone the number of particular blocks according to the particular test setup requirements.

To ensure an easy integration of particular blocks into the test system all the blocks are equipped with standard hardware interface, which signals can be divided into three groups. At the CAN side it consists of standard CAN_TX/CAN_RX signals and a set of three software controlled signals for physical drivers with sleep mode and wake-up features. The CAN_TX signals may be wire-anded internally in the FPGA and therefore they can share a single CAN bus driver. At the Trigger I/O side a function specific number of trigger inputs and outputs is defined, that are used to synchronize the specific IP function activity with events generated by other IP functions or to provide its local event identification. Output triggers are also used for event time stamping and/or interrupt request generation. At the third side of each IP function there is an interface to the host (microprocessor), consisting of four control signals, necessary number of address lines and 8/16/32 bit data path.

A 32-bit time base with 1 μs resolution together with a number of compare and/or capture registers provides for the precise test execution timing. The compare registers are used to start time-driven activities - their value is preloaded and continually compared with the time base value. When the time base reaches the compare register value, the trigger output is activated. The capture registers are used to capture the time stamps of asynchronous events coming from external buses.

General FPGA test system structure is shown in Fig. 1 (some interconnections are missing). One can see the support for arbitrary number of CAN networks; each network tests can additionally be supplied with more than one IP function of particular type (not shown in Fig. 1). Trigger I/O interconnection depends on the synchronization and time-stamping requirements and can be configured respectively.

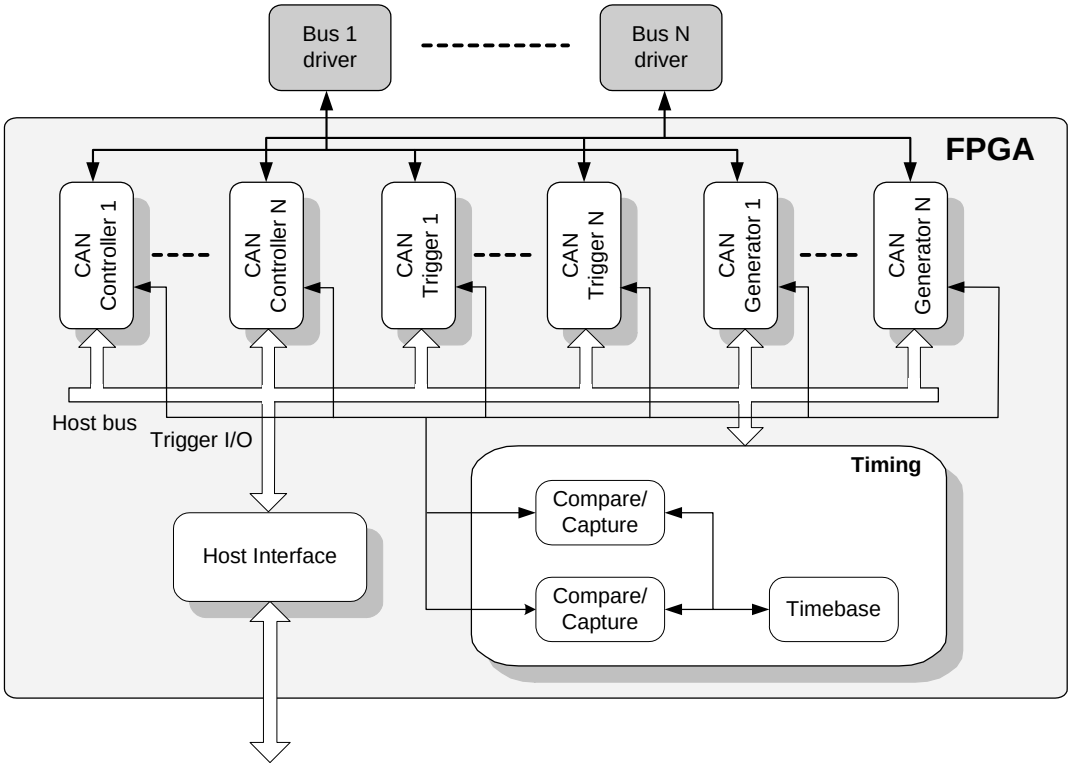


Fig. 1. General test system structure.

5. IP Functions

Three IP functions were designed and implemented for the described testing approach. Two of them (the CAN Controller and CAN Trigger) seem to be the modules with standard functionality, which is available on the market. Nevertheless, as explained below, they offer much more than the standard features.

The third IP function (CAN Generator) provides very unique features that are not commercially available at all.

5.1. CAN Controller IP Function

The implementation of the CAN Controller provides more than the standard CAN controller implementations. Here only the improvements that are important for testing of CAN based systems will be mentioned (see Fig. 2).

The controller offers two transmission queues with hardware-triggered transmission – the first supports an event-triggered transmission, where the queued messages are transmitted on an external request. The second queue supports a time-triggered transmission, where the transmission request is generated internally by the compare timer and used as a trigger source. The message data written into this queue contains the absolute transmission time in addition to the standard CAN message fields. This time value is compared with the value of the time-base oscillator and transmission is triggered when it is equal.

The event-triggered queue suits for applications that require fast response on bus events, the time-triggered queue can easily be used e.g. for the rest-bus simulation. The mutual priority of both queues can be preset and it can even be changed dynamically for particular messages.

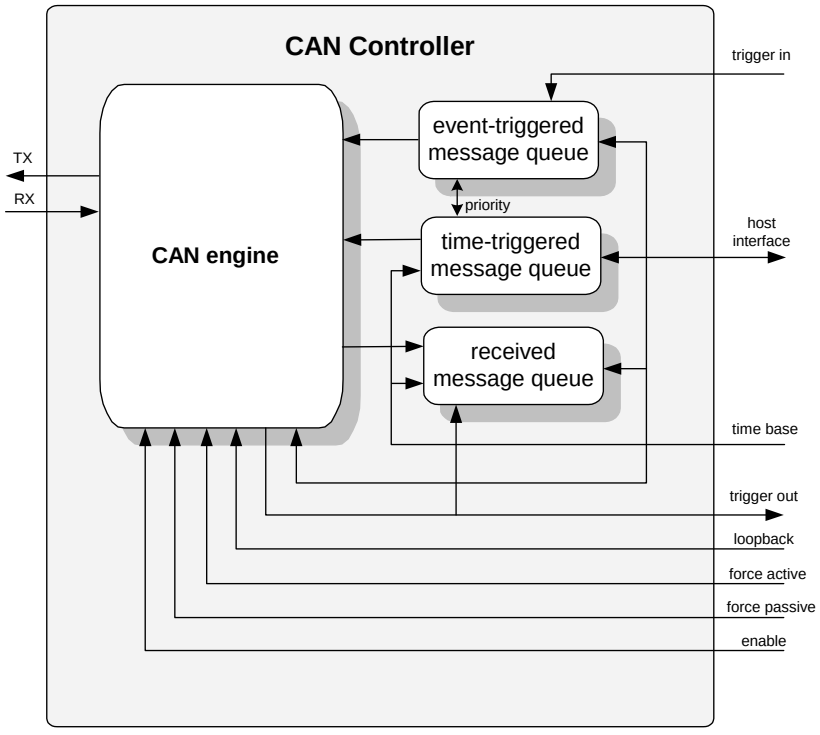


Fig. 2. CAN Controller block diagram.

The reception acceptance filtering provides much higher granularity than available in standard controllers. For standard identifier length the filtering of particular IDs can be individually switched on or off, as there is a 2048 long bit field controlling each particular ID. For extended identifiers either the same filtering like in standard controllers is available (ID, mask), or the frame reception can be qualified by the output of the Trigger IP function (-s). This is available for standard IDs as well.

The controller provides a received message queue, containing also the reception timestamps. Each message reception is signalized by the output trigger signal. Transmitted frames can be looped back into the received message queue.

Finally, the error state of the controller can be forced into the error-active or error-passive state independently on the error counter values.

5.2 CAN Trigger IP Function

For some tests the precise synchronization to the bus events is required even before they are successfully finished. This is e.g. the case of frame occurrence with a specific ID (or ID range) on the bus. An example of this need is the selective jamming of chosen CAN frames during their transmission. As the ordinary CAN controller provides this information too late (when jamming has no effect), the CAN Trigger IP function was developed, which provides the early trigger functionality.

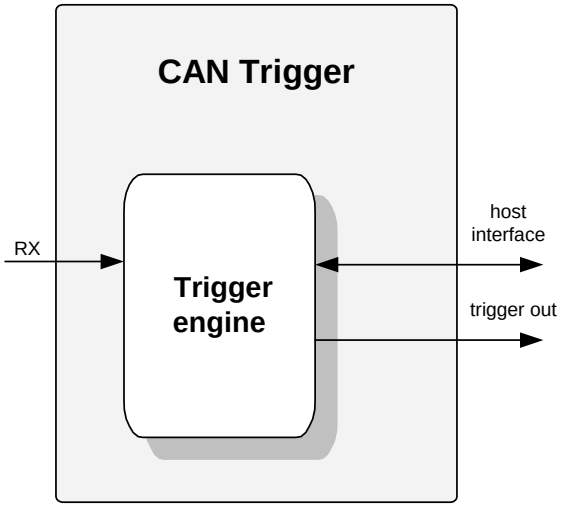


Fig. 3. CAN Trigger block diagram.

It is able to evaluate not only the message ID, but to search for a specific control information and/or data content. Generally it allows defining the bit sequence and comparing it to the incoming bit sequence, applying simultaneously the mask, which defines what bits of the frame the comparison should take into the account. In case of matching the output trigger signal is generated, either asynchronously (as fast as possible) or synchronously with the reconstructed CAN bus timing. This trigger output can either be used for time stamping or for synchronization of other test system components.

5.3. CAN Generator IP Function

Generally, the CAN Generator (Fig. 4) allows transmitting of CAN frames consisting of predefined bit sequence [4]. Several flags that modify the way it is transmitted to the bus define each bit in the sequence.

Particular flags define the bit logical value, its positioning in arbitration field (if set, in case on reception of dominant value while sending the recessive one the transmission is finished), its positioning as the last bit in frame (if set, by transmitting of this bit the frame transmission finishes), its positioning as the ACK bit (if received recessive, the acknowledge error status flag is set) and its back read checking (if set, the check of transmitted bit value by reading it back is ignored).

Additionally, each bit definition contains a pointer to the table of bit time intervals, which may contain up to 128 different bit lengths with 20 ns resolution. It offers enough flexibility to simulate e.g. transmission clock variations and similar phenomena.

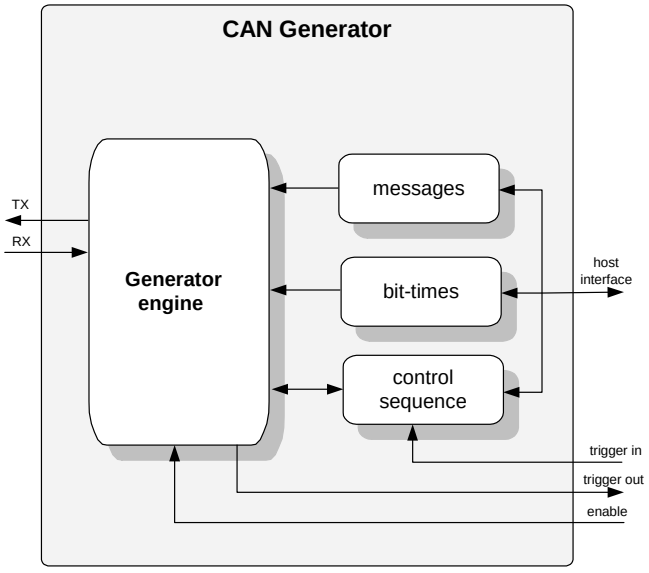


Fig. 4. CAN Generator block diagram.

Nevertheless, the CAN Generator functionality is not limited at transmission of a single frame. Users can define so called program sequence, which may include not only the SEND instruction to transmit the frame, but also the GOTO (unconditional jump at specified program position), DELAY (waiting for predefined time interval), SETVAR (setting the value of selected program variable), DECVAR (decrementing the value of selected program variable), IF (conditional jump at specified program position if value of selected variable equals to instruction parameter), TEST (conditional jump at specified program position if result of <value of selected variable & instruction parameter> is not zero), ARM (transmission of the frame on the occurrence of the hardware trigger) and END (stops the program sequence execution).

CAN Generator offers transmission of up to 256 different frames in a single sequence and up to 128 different lengths of bit period. The frame definitions are stored in external RAM (if the internal FPGA memory is not large enough), the bit timetable as well as the program sequence is stored in the internal FPGA RAM (short access time is required).

6. Test Hardware Implementations

The first generation of the test site implementation using above described IP functions was designed in form of the PCI card (see Fig. 5).

An advantage of this approach is especially a fast and straightforward access to the IP function registers, the disadvantage is a limited portability of the test site. The board supports simultaneous testing on up to two independent CAN networks (the number of networks is limited only by the number of available I/O signals, as the bus drivers are on the external modules). The external trigger input and output signals are also available allowing the seamless synchronization with other (if necessary) test system components.

Second generation of the test HW is now under the development. It is based on a standalone module equipped with a large FPGA and local microprocessor. PC is used to provide only a GUI, all test procedures running in real-time are processed by a local processor. Local microprocessor communicates with PC by means of USB and allows seamless programming of the FPGA from the PC. Simple graphical programming tool is planned to support this feature.

Fig. 5. First Generation of the test hardware implementation

7. Case Studies

Two case studies are introduced below. The first one is focused on investigation of ECU behavior during the bus-off situation (single CAN bus test), the second one on the correct indication of obsolete data in the CAN frames retransmitted from the source to the destination bus (dual CAN bus test). The third case study focused on evaluation of a sample point position within the bit interval of the tested ECU is available in [5].

7.1. Bus-Off State Behavior Evaluation

The goal of the test is to evaluate the behavior of the ECU during the bus-off situation. Bus-off is one of the CAN controller error states (remaining are error-active and error-passive). In the bus-off state

the CAN controller is logically disconnected from the bus and it doesn't take part in communication. CAN controllers move into the bus-off state when they encounter too many errors during their transmission attempts, which leads to the transmit error counter overflow. Most of car manufacturers require for the ECU's firmware to reinitialize the CAN controller and to restart the communication. Of course, there is some timeout required before the re-initialization takes place, which usually increases with each following bus-off state recovery. In some applications it is required to start receiving immediately after the re-initialization, but wait for a specific timeout before starting the transmission.

What is necessary for such an evaluation? First we need an additional CAN bus node, that will deliver the frame acknowledge for the tested ECU. The same node can be used for the rest-bus simulation. The CAN Controller IP is a natural candidate for this role. Second we need the node, which is able to jam the CAN communication issued by the tested ECU in order to force the transition into the bus-off state. The CAN Generator IP can be used for this purpose. As it is able to transmit any sequence of bits, the sequence of six dominant bits that forms an error flag can be sent as well. The last functionality we need is a synchronization of the error flag transmission start, as the CAN Generator should only jam the transmission attempts of the tested ECU and not the frames of the rest-bus simulation. CAN Trigger IP offers this functionality, providing and early trigger output immediately when the frame ID field is transmitted. The required number of CAN Trigger IP functions has to be used simultaneously to cover all the frame IDs transmitted by the tested ECU. The test system setup is shown in Fig. 6.

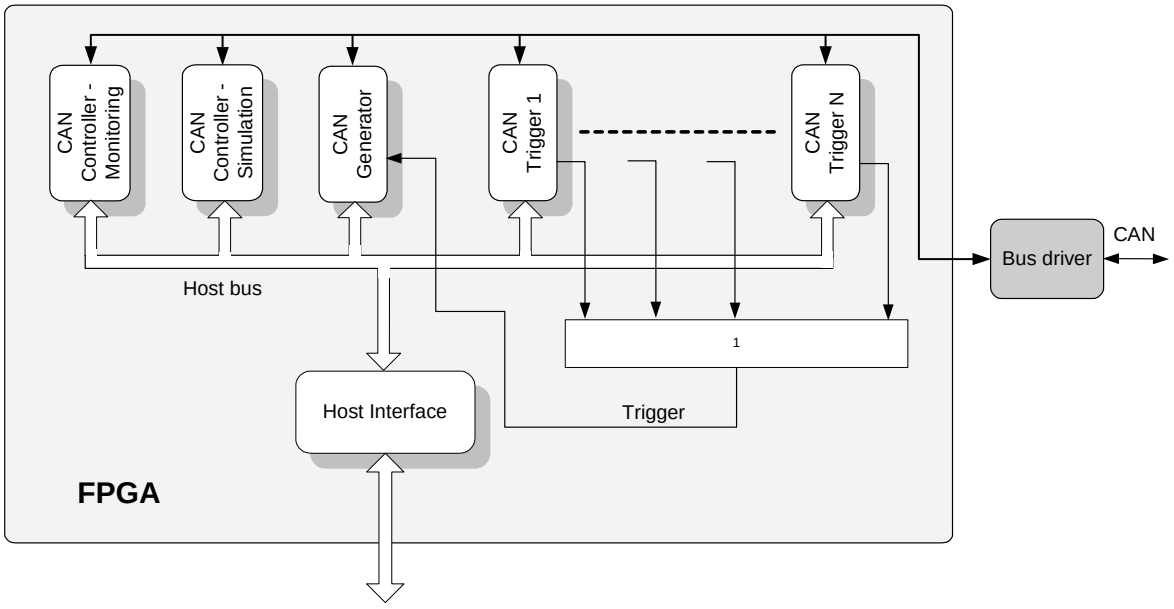


Fig. 6. Test setup for bus-off behavior investigation.

What is now the test sequence? CAN Controller IP is used to provide the rest-bus simulation and acknowledge delivery. CAN Trigger IP functions are programmed to deliver the trigger on an occurrence of particular message IDs transmitted by the tested ECU. Their outputs are or-ed providing a single trigger for the CAN Generator IP. CAN Generator IP is programmed to transmit 32 times an active error flag (six consecutive dominant bits) of nominal bit length on each trigger. Another CAN Controller IP function is used for the passive bus monitoring.

First the tested ECU is switched on and the rest-bus simulation is started. Then the CAN Generator is activated. After the 32 transmission attempts the tested ECU goes into the bus-off state, it means it is logically disconnected from the bus. CAN Generator sequence is finished and the respective timestamp

is captured. The timestamp of the next transmission attempt of the tested unit can be either captured by the CAN Trigger functions or evaluated from the timestamps of CAN messages received by the monitoring CAN Controller IP.

7.2. Obsolete Data Retransmission Indication Test

The goal of the test is to evaluate the status of data that is retransmitted by EUT (ECU under test) from the CAN2 to the CAN1. On the CAN1 the frame containing the evaluated data is transmitted periodically regardless on the presence of the source frame on the CAN2, as it may contain not only retransmitted data, but also the data that are issued by the EUT. Therefore there is a mechanism implemented allowing marking the retransmitted data as obsolete, if the source bus data are not available for a predefined period of time.

Principal test setup is shown in Fig. 7. CAN Controllers 12 and 22 are used for simulation of remaining ECUs on CAN1 resp. CAN2. Controllers 11 and 21 are used for bus traffic monitoring. It would be possible to run this test with only one CAN Controller per bus, but complete traffic monitoring (including the frames transmitted by rest bus simulation) can be easily serviced by the software in the current setup.

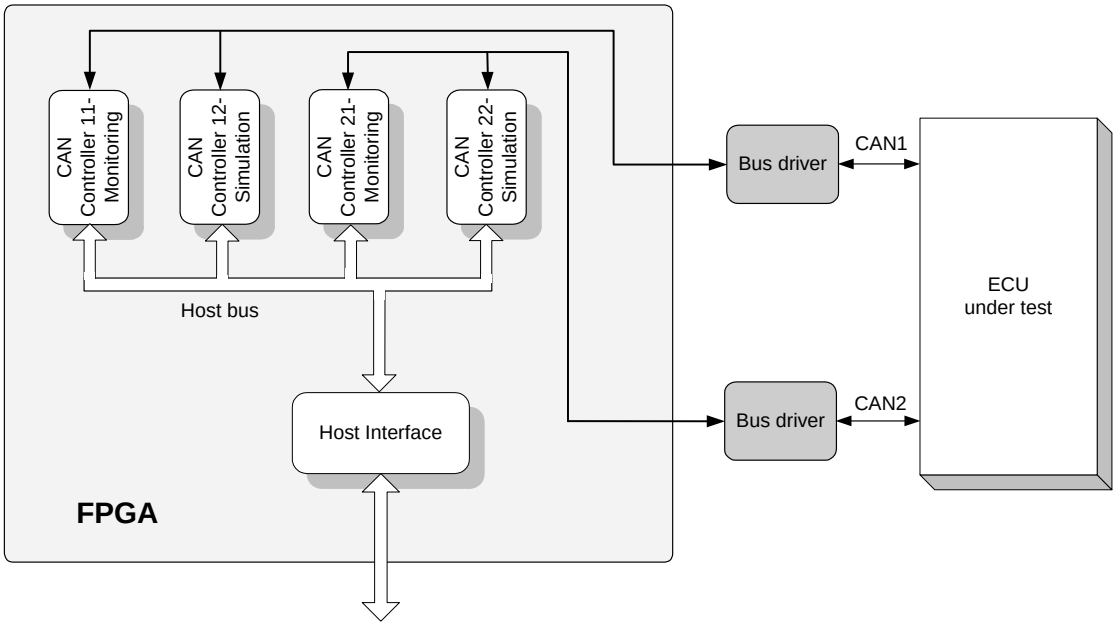


Fig. 7. Test setup for obsolete data retransmission indication test.

The test itself is quite simple. The transmission of the source data on CAN2 is interrupted (by simulation) for the required time period and the flag indicating the destination bus (CAN1) data obsolescence is checked. As all received frames on both buses are time-stamped, evaluation of the minimum period of the source data dropout is easy.

8. Conclusions

The flexible platform for CAN based ECUs and systems testing was designed and developed and examples of its application were presented. It consists of three IP functions with unique features, allowing easy and fast development of multi-network test sites. As there is a strong hardware support

for inter-function synchronization and event time stamping, the real-time requirements for the host service are quite low. In-addition, as the concrete configurations are implemented in the FPGA, it is easy to add a hardware support for other features or for the other bus types (e.g. the LIN or FlexRay).

Acknowledgements

This work was supported by the Czech Ministry of Education project No. 1M0568 – Josef Bozek Research Center of Engine and Automotive Engineering.

References

[1]. T. Bardelang, CAN im Hardware in the Loop (HiL) Simulator - Erfahrungen mit vernetzten Fahrzeugfunktionen und deren Prüfung im Rahmen der Anwendung eines HiL-Simulators für schwere Lkw, *Vector Symposium CAN im Nutzfahrzeug*, Stuttgart, May 12, 2004.
[2]. R. Bosch, GmbH, CAN Specification 2. 0, *Robert Bosch GmbH*, 1991.
[3]. K. Etschberger, Controller Area Network, *IXXAT Press*, Weingarten, Germany, 2001.
[4]. J. Novák, A. Fried, and M. Vacek, CAN Generator and Error Injector, *Proceedings* of *IEEE International Conference on Electronics, Circuits and Systems ICECS2002*, CD ROM, September 2002, Dubrovnik, Croatia, pp. 967–970.
[5]. J. Novák, New Measurement Method of Sample Point Position in Controller Area Network Nodes, *Measurement*, Vol. 2008, No. 41, p. 300-306.

Guide for Contributors

Aims and Scope

Sensors & Transducers Journal (ISSN 1726-5479) provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes state-of-the-art reviews, regular research and application specific papers, short notes, letters to Editor and sensors related books reviews as well as academic, practical and commercial information of interest to its readership. Because it is an open access, peer review international journal, papers rapidly published in *Sensors & Transducers Journal* will receive a very high publicity. The journal is published monthly as twelve issues per annual by International Frequency Association (IFSA). In additional, some special sponsored and conference issues published annually. *Sensors & Transducers Journal* is indexed and abstracted very quickly by Chemical Abstracts, IndexCopernicus Journals Master List, Open J-Gate, Google Scholar, etc.

Topics Covered

Contributions are invited on all aspects of research, development and application of the science and technology of sensors, transducers and sensor instrumentations. Topics include, but are not restricted to:

- Physical, chemical and biosensors;
- Digital, frequency, period, duty-cycle, time interval, PWM, pulse number output sensors and transducers;
- Theory, principles, effects, design, standardization and modeling;
- Smart sensors and systems;
- Sensor instrumentation;
- Virtual instruments;
- Sensors interfaces, buses and networks;
- Signal processing;
- Frequency (period, duty-cycle)-to-digital converters, ADC;
- Technologies and materials;
- Nanosensors;
- Microsystems;
- Applications.

Submission of papers

Articles should be written in English. Authors are invited to submit by e-mail editor@sensorsportal.com 8-14 pages article (including abstract, illustrations (color or grayscale), photos and references) in both: MS Word (doc) and Acrobat (pdf) formats. Detailed preparation instructions, paper example and template of manuscript are available from the journal's webpage: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm Authors must follow the instructions strictly when submitting their manuscripts.

Advertising Information

Advertising orders and enquires may be sent to sales@sensorsportal.com Please download also our media kit: http://www.sensorsportal.com/DOWNLOADS/Media_Kit_2009.pdf

Subscription 2010 is open!
SUBSCRIBE TODAY!

Sensors & Transducers Journal provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes reviews, regular research and application specific papers and short notes. This peer reviewed international journal is indexed and abstracted very quickly by Chemical Abstracts, IndexCopernicus Journals Master List (ICV=6.31 at the end of 2008), Open J-Gate, Google Scholar, Scirus, etc.

Topics of Interest

Include, but are not restricted to:
Physical, chemical and biosensors
Digital, frequency, period, duty-cycle, pulse number output sensors and transducers
Theory, principles, effects, design, standardization and modelling
Smart sensors and systems
Sensor instrumentation
Virtual instruments
Sensors interfaces, buses and networks
Signal processing
Technologies and materials
Nanosensors
Microsystems
Applications

2008
e-Impact Factor
205.767

IFSA
www.sensorsportal.com

editor@sensorsportal.com
http://www.sensorsportal.com/HTML/DIGEST/Journal_Subscription_2010.htm

www.sensorsportal.com