CAN Network with Time Triggered Communication

Florian Hartwich
Bernd Müller
Thomas Führer
Robert Hugel
Robert Bosch GmbH

The communication in the classic CAN network is event triggered; peak loads may occur when the transmission of several messages is requested at the same time. CAN's non-destructive arbitration mechanism guarantees the sequential transmission of all messages according to their identifier priority. For hard real-time systems, a scheduling analysis of the whole system has to be done to ensure that all transmission deadlines are met even at peak bus loads.

For a RTOS that is based on static cyclic scheduling of all tasks, system integration and composability are served when the communication on the CAN network also follows a synchronised schedule.

The time triggered communication option of the forthcoming new edition of ISO11898-1 describes the prerequisites needed for the synchronisation of all nodes in the CAN network. Based on the synchronisation of the nodes, the time triggered communication facilitates also the establishment of a global time in higher-layer protocols. A higher-layer protocol above the unchanged standard CAN protocol is in the process of standardization by ISO TC22/SC3/WG1/TF6, as ISO11898-4.

In parallel to the standardization process, Bosch has implemented the time triggered communication function into a CAN IP module that maintains the cyclic transmission schedule autonomously, not depending on software control.

This paper describes a CAN network with time triggered communication consisting of CAN controllers with autonomous message scheduling.

1 Introduction

TTCAN is a higher-layer protocol above the unchanged standard CAN protocol that synchronises the communication schedules of all CAN nodes in a network and that provides a global system time. When the nodes are synchronised, any message can be transmitted at a specific time slot, without competing with other messages for the bus. Thus the loss of arbitration is avoided, the latency time becomes predictable. Apart from the synchronised communication schedule, the TTCAN nodes operate according to the standard CAN protocol as defined by ISO 11898-1. The TTCAN protocol is in the process of standardization [4] by ISO TC22/SC3/WG1/TF6. [1] describes TTCAN on system level. This paper describes the implementation of TTCAN features into a CAN module and the evaluation of a TTCAN network.

2 Time Reference

In order to synchronise the activities of all CAN nodes within a network, a common time reference is needed. Each node has its own local time, which is a counter that is incremented each network time unit (NTU). The system wide NTU is derived from the node's local clock and local time unit ratio (TUR).

In TTCAN, the cyclic transmission schedule is synchronised by the repeated transmission of a particular CAN message, the reference message. This reference message (transmitted by a time master) restarts the cycle time in each node. The cycle time is derived from the node's local time.

Figure 1 describes the synchronisation of the cycle time, performed in the same manner by all TTCAN nodes, including the time master. Any message received or transmitted

invokes a capture of the local time taken at the message's frame synchronisation. This frame synchronisation event occurs at the sample point of each start of frame (SOF) bit and causes the local time to be loaded into the Sync_Mark register.

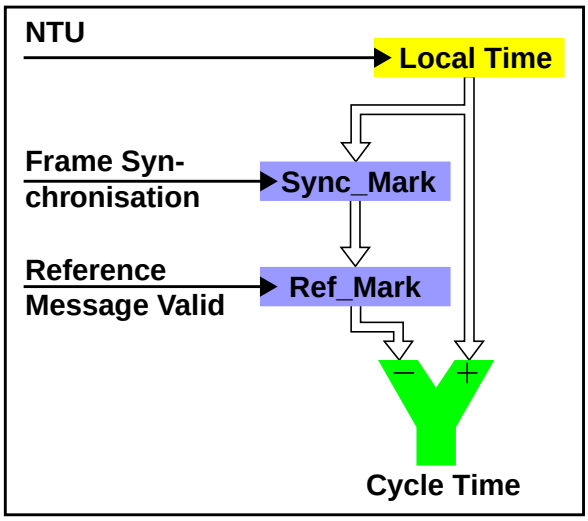

Figure 1: Cycle Time Synchronisation

Whenever a valid reference message is transmitted or received, the contents of the Sync_Mark register is loaded into the Ref_Mark register. The difference between the actual value of the Ref_Mark and the local time is the cycle time (Cycle Time = Local Time – Ref_Mark).

3 Basic Cycle

Each reference message starts a new basic cycle; the cycle time counts the time since the start of the basic cycle. The basic cycle consists of several non-overlapping time windows of different size. The sequence of the time windows is described by time marks that define when a time window begins.

The structure of the basic cycle is defined once for the whole TTCAN network. Several basic cycles may be combined to build a matrix cycle or system matrix, the sequence of basic cycles in the matrix cycle is controlled by the reference messages. All possible messages in the TTCAN system are assigned to particular elements (time windows) of the system matrix. The system designer decides whether a time window is an exclusive window (only one particular message may be transmitted in this window), an arbitrating window (messages may arbitrate for bus access), or a free window (may be reserved for further extensions of the network). The automatic retransmission of messages that could not be transmitted successfully is disabled, guaranteeing that messages in exclusive time windows are not delayed by other bus traffic.

A TTCAN node is not required to know the whole system matrix. It is sufficient to implement only that time marks that are necessary to define the time slots for the node's own transmit messages and to check whether received messages have arrived on time. The time marks are compared to the cycle time (see Figure 1). When the cycle time reaches a time mark, a particular action is triggered.

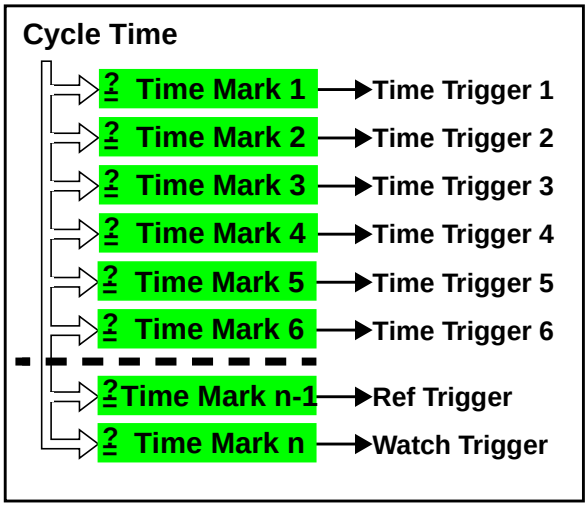

Figure 2: Scheduling based on Cycle Time

With the exception of the Watch Triggers, all Time Triggers are linked to messages of the CAN node. They either cause the start of the message's transmission or they define when the reception of the message is checked.

The Ref Trigger is linked to the reference message. When the time master transmits a reference message, a new basic cycle starts and the cycle time is reset. A Watch Trigger can only be reached when the reference message is not completed on time; an error handling procedure is initiated.

4 Global System Time

Time triggered CAN will be implemented in two levels. Level 1 is restricted to the cyclic message transfer only, level 2 also supports a global system time. The time master estab-

lishes its own local time as global time by transmitting its own Ref_Marks in the reference message, as Master_Ref_Marks.

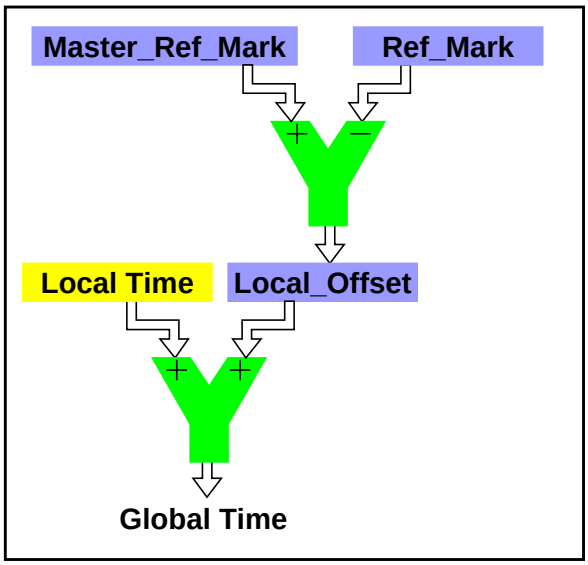

Figure 3: Generating the Global Time

The time slaves calculate their local time offset to the global time by comparing (see Figure 3) their local Ref_Mark with the received Master_Ref_Mark. The node's view of the global time is local time + local offset.

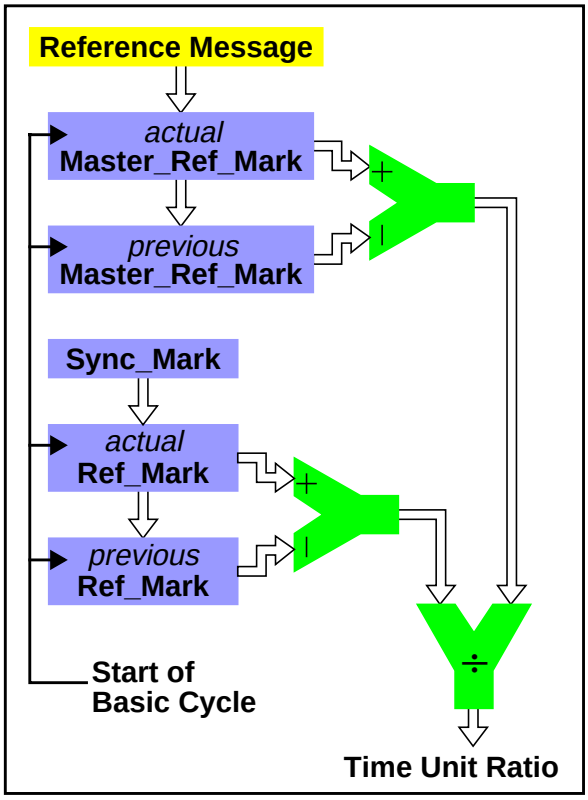

Figure 4: Drift Compensation

To compensate for slightly different clock drifts in the TTCAN nodes and to provide a consistent view of the global time, the nodes perform a drift compensation operation (see Figure 4). They compare the length of the basic cycle in local time (Ref_Mark – previous Ref_Mark) with the length of the basic cycle in global time (Master_Ref_Mark – previous Master_Ref_Mark). The quotient of the two values gives the adapted time unit ratio.

5 TTCAN Implementation

The cyclic message transfer of TTCAN level 1 can be implemented in software, based on existing CAN modules. Depending on the CAN bit rate and on the number of messages in the system matrix, the software approach may result in a high CPU load. For the evaluation of the TTCAN protocol, the hardware approach was chosen.

In parallel to the standardization process, Bosch develops an IP module that implement the TTCAN protocol. This IP module, the TT_CAN, is based on the existing C_CAN IP module [5] and will be available as VHDL code to be synthesized in FPGAs, supporting the development of CAN based time triggered communication networks.

The C_CAN consists of the components (see Figure 5) CAN Core, Message RAM, Message Handler, Control Registers, and Module Interface.

The CAN_Core performs communication according to the CAN protocol version 2.0, as defined in ISO 11898-1. The bit rate can be programmed to values up to 1MBit/s depending on the used technology. For the connection to the physical layer additional transceiver hardware is required.

For communication on a CAN network, individual Message Objects are configured. The Message Objects and Identifier Masks for acceptance filtering of received messages are stored in the Message RAM.

All functions concerning the handling of messages are implemented in the Message Handler. Those functions are the acceptance filtering, the transfer of messages between the CAN Core and the Message RAM, and the handling of transmission requests as well as the control of the module interrupt.

The register set of the C_CAN can be accessed directly by an external CPU via the module interface. These registers are used to control/configure the CAN Core and the Message Handler and to access the Message RAM.

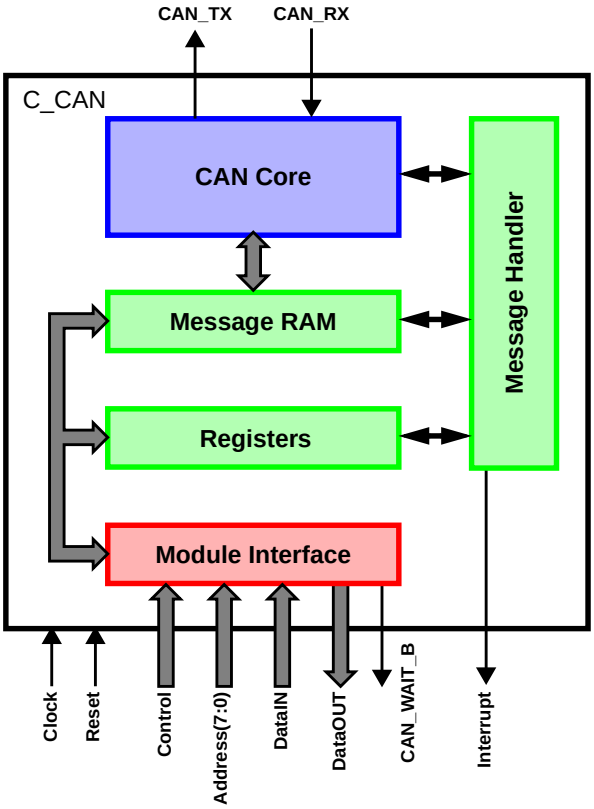

Figure 5: Block Diagram of the C_CAN

Several Module Interfaces are available, including interfaces to ARM, Motorola and Texas Instruments CPUs.

Compared to the C_CAN, the TT_CAN is expanded by two functional blocks (see Figure 6), the Trigger Memory and the Frame Synchronisation Entity FSE.

The Trigger Memory stores the time marks of the system matrix that are linked to the messages in the Message RAM; the data is provided to the Frame Synchronisation Entity.

The Frame Synchronisation Entity is the state machine that controls the time triggered communication. It synchronises itself to the reference messages on the CAN bus, controls the cycle time, and generates Time Triggers. It is divided into five blocks (see Figure 7), the Time Base Builder TBB, the Cycle Time Controller CTC, the Time Schedule Organiser TSO, the Master State Administrator MSA, the Application Operation Monitor, and the Global Time Unit GTU.

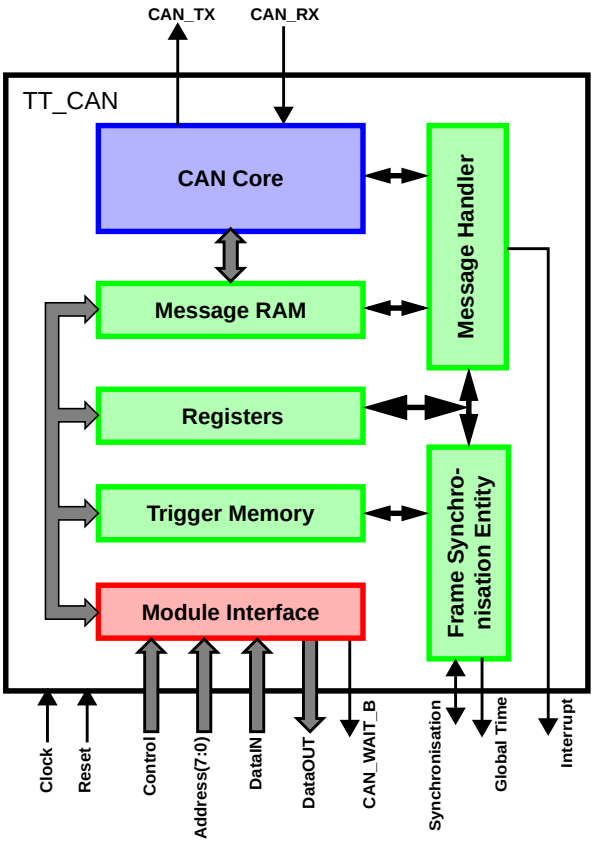

Figure 6: Block Diagram of the TT_CAN

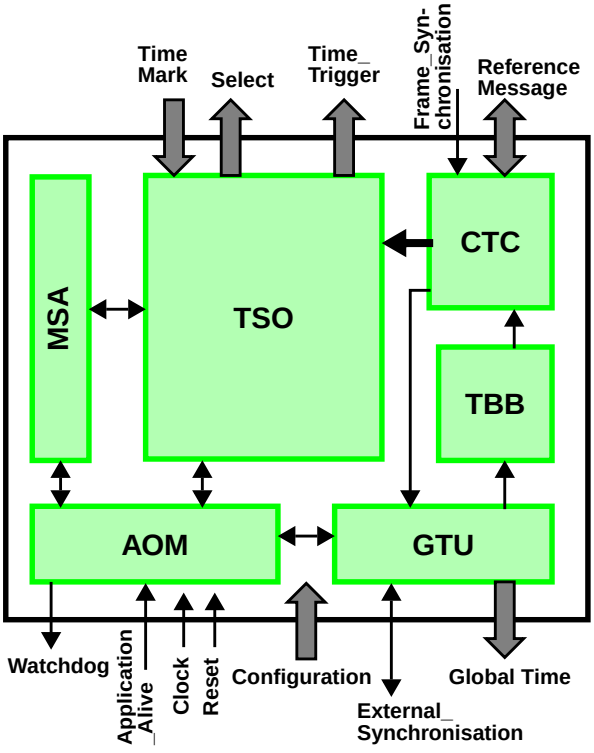

Figure 7: Frame Synchronisation Entity

Time Base Builder generates the local time from the node's system clock and the time unit ratio. In TTCAN level 1, the TUR is defined at configuration, in level 2, it is continuously adapted by the GTU.

The Cycle Time Controller gets the local time from the TBB, the Frame_Synchronisation events from the CAN_Core, and the reference messages from the Message Handler. It captures the Sync_Mark and the Ref_Mark to generate the cycle time and controls the sequence of the basic cycles in the matrix cycle. Cycle_Count (part of the reference message) identifies the actual basic cycle inside the matrix cycle. Depending on whether the node itself is time master (transmitter of reference messages), Cycle_Count is either generated from a cyclic counter or it is received in the reference message.

The Time Schedule Organiser maintains the message schedule inside a basic cycle and checks for scheduling errors. The schedule is defined by data in the Trigger Memory. The data consists of time mark (measured in cycle time), and function (trigger for transmission or check of reception), and is linked to a message in the Message RAM. The same time mark may be, selected by Cycle_Count, defined for different messages at different basic cycles of the matrix cycle. Other time marks are defined for the Ref Trigger and the Watch Trigger. The TSO compares the time marks with the cycle time and activates the Time Triggers for messages with matching time marks. The function of the TSO depends on the actual operating state; transmissions are disabled when the node is not synchronised to the system. If the node is time master, the Ref Trigger causes the reference message to be transmitted. The Watch Trigger becomes active at the end of a basic cycle, when the expected start of a new basic cycle (completion of a reference message) does not occur. This event is causes the MSA to change the operating state.

The Master State Administrator controls the FSE's operating state. The operating state depends on whether the node is synchronised to the network, whether it is (trying to become) time master or whether it is a backup time master. The synchronisation state differentiates between synchronising after the initialisation, the active mode, the loss of synchronisation, and the fault recovery. The function of the other blocks is monitored. In case of errors, transmissions are disabled and the master state is resigned

The Application Operation Monitor checks the function of the application program. The application controller has to serve the Application_Alive input regularly. If the application program fails, the application watchdog causes the MSA to disable all transmissions, preventing invalid data to disturb the system.

The Global Time Unit (TTCAN level 2 only) generates the node's view of the global time and controls the drift correction of the local time. When the node is the first time master of the network, the node's local time is the global time. When the node is not operating as time master, the difference between local Ref_Mark and Master_Ref_Mark received in the reference message is compared and defines the actual offset between the local time and the global time. The actual offset is updated at each start of a basic cycle; when the node becomes time master, the last offset is kept, avoiding a discontinuity in the global time. The Global_Ref_Mark (captured in global time) is provided as Master_Ref_Mark for reference messages to be transmitted.

The GTU compensates the drift between the local time and the global time by calibrating the local time. If the node itself is the time master, no calibration is done. Each time a reference message is completed, the actual length of the base cycle is measured in local time (Ref_Mark - previous Ref_Mark) and in global time (Master_Ref_Mark - previous Master_Ref_Mark). The difference between the two measured values divided by the length of the base cycle shows the factor by which the local time has to be calibrated in order to compensate the drift. The compensation is performed by adapting the TUR the TBB uses to generate the local time from the node's system clock. The calibration process is on hold when the node is not synchronised to the system and is (re-)started when it (re-)gains synchronisation.

Frequent significant changes in the measured drift indicate an unreliable local time

base. Time base errors are signalled to the MSA, causing it to stop all TTCAN operations.

In order to synchronise different TTCAN networks, or to provide a physical time base, the global time may be synchronised (via the time master) to an external clock, e.g. GPS.

The TTCAN implementation is done in two steps. In the first step, only level 1 is implemented, without the global system time and drift compensation of level2.

In the second step, after the evaluation of the TT_CAN IP module in a TTCAN network, a global time unit will be added to the module.

6 TTCAN Network

The first TTCAN network is build using Bosch's CAN IP evaluation boards (see Figure 8). The boards have been developed for the analysis of the CAN IP modules A_CAN, C_CAN and CAN_Gateway. The IP modules are synthesized into an FPGA (Field Programmable Gate Array); the boards provide storage for an application program to be run on a Motorola HC08AZ0 CPU and different CAN transceivers (high-speed and low-speed CAN buses). Up to three CAN transceivers are available for Gateway implementations. An emulator may be connected to P4/P5 for software development.

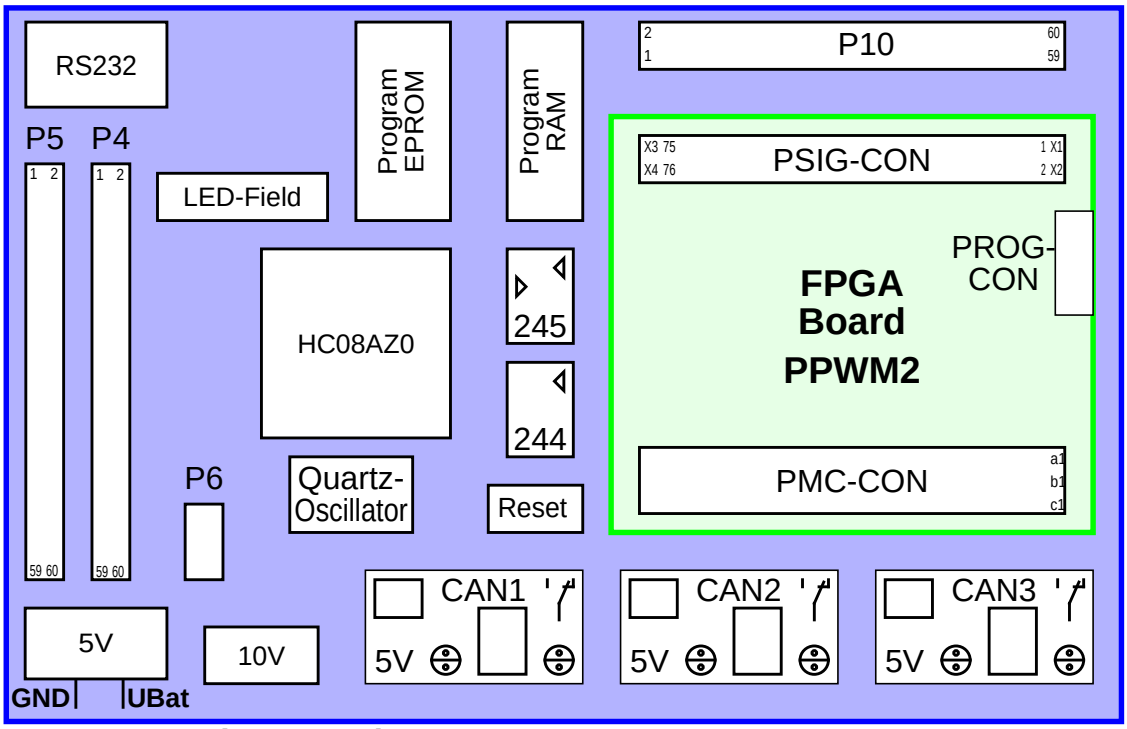

Figure 8: CAN IP Evaluation Board

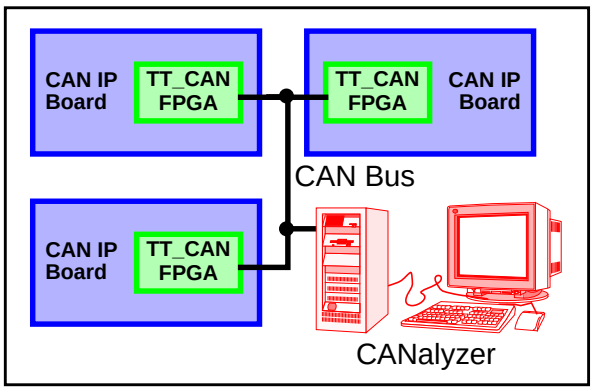

Figure 9: TTCAN Network

The TTCAN network for the evaluation of the first TT_CAN IP module implementation consists of (see Figure 9) three CAN IP evaluation boards acting as TTCAN nodes and one CANalyzer tool used to monitor the CAN bus traffic or to disrupt the time triggered message schedule.

The CANalyzer tool is a standard CAN network analysis tool, not modified for TTCAN.

The different application programs on the CAN evaluation boards control the TT_CAN IP modules and provide initialisation data.

All three TTCAN nodes may become time master, their time master priority is defined by their reference message identifiers.

7 Summary

The TTCAN protocol level 1 is implemented in a CAN IP module and synthesized into an FPGA. The FPGA modules are used to evaluate the TTCAN protocol in different operation modes.

The first evaluated mode is the undisturbed operation, where the sequential time schedule of the cyclic message transfer is strictly maintained.

Clock drift between the TTCAN nodes causes a phase shift of the messages inside their time windows. Without the adaptive drift compensation of TTCAN level 2, the drift is compensated by the lengthening of the time windows. The amount by which the time windows are lengthened increases from the first to the last time window of a basic cycle.

Failure of the actual time master is emulated by resetting the time master or by disconnecting it from the CAN bus. The node with the next highest priority becomes time master. It releases the mastership when the first time master is reinitialised and restarts TTCAN operation.

Fault recovery is evaluated after disturbed reference messages and when the message schedule is disrupted by additional messages. The CAN bus disturbances and additional messages are provided by the CANalyzer tool.

The next step of the TTCAN protocol evaluation will be the implementation of the global time unit into the TT_CAN IP module.

8 Literature

1. Time Triggered Communication on CAN; Th. Führer, B. Müller, W. Dieterle, F. Hartwich, R. Hugel, M. Walther, Robert Bosch GmbH; Proceedings 7th International CAN Conference; 2000.
2. Road vehicles - Controller area network (CAN) - Part 1: Controller area network data link layer and medium access control; ISO 11898-1.
3. Road vehicles - Controller area network (CAN) - Part 2: High-speed medium access unit and medium dependent interface; ISO 11898-2.
4. Road vehicles - Controller area network (CAN) - Part 4: Time triggered communication; Working Draft ISO 11898-4.
5. C_CAN User's Manual; Robert Bosch GmbH; 2000; http://www.bosch.de/de_e/productworld/k/products/prod/can/docu/Users_Manual_C_CAN.pdf.
6. Guaranteeing Message Latencies on Controller Area Network (CAN); K. Tindell, A. Burns; Proceedings 1st International CAN Conference; 1994; pp 2-11.
7. Real-Time Systems: Design Principles for Distributed Embedded Applications; H. Kopetz; Kluwer Academic Publishers; 1997; ISBN 0-7923-9894-7.

Florian Hartwich
Robert Bosch GmbH
P.O.Box 1342
72762 Reutlingen
Germany
Phone: +49 7121 35-2594
Fax: +49 7121 35-1746
Email: Florian.Hartwich@de.bosch.com
http://www.CAN.Bosch.com