**IP 07**

**Session : IP Design 1**

# A FPGA-BASED SOLUTION FOR ENFORCING DEPENDABILITY AND TIMELINESS IN CAN

| José Rufino | Ricardo Pinto | Carlos Almeida |
|---|---|---|
| ruf@di.fc.ul.pt | ricardo.pinto@ist.utl.pt | cra@comp.ist.utl.pt |
| FCUL† | IST/UTL‡ | IST/UTL |

**Lisboa, Portugal**

## Abstract :

The Controller Area Network (CAN) plays a very important role in the design and implementation of distributed embedded systems, in areas such diverse as industrial automation, automotive, avionics and aerospace.
However, the native CAN protocol exhibits a set of availability, reliability and timeliness limitations. This paper identifies a fundamental set of shortcomings of the standard CAN protocol and shows how the problem has been tacked in the implementation of the CANELy architecture, a CAN-based infrastructure able of extremely reliable hard real-time communication.
In particular, the paper discusses how the use of a FPGA-based design is helpful in enforcing the desired CANELy dependability and timeliness properties.

## INTRODUCTION

The design and implementation of distributed embedded computer control systems have increasingly been based on standard fieldbuses, such as CAN, the Controller Area Network [2, 1]. CAN is traditionally considered a robust fieldbus. However, the native CAN protocol exhibits a set of severe limitations with regard to the provision of strict availability, reliability and timeliness guarantees. Given the large practical base of off-the-shelf microcontrollers integrating standard CAN interfaces, a fundamental question is whether (and how) these components can be used for highly dependable applications of CAN?

It has been shown that what is missing in the standard CAN fieldbus to attain high levels of dependability is indeed a set of fault tolerance and timeliness-related services [13, 8, 10, 11, 12]. Moreover, these can be provided off-the-shelf (i.e. without modifications to the CAN standard or to commercial CAN controllers), through the use of properly encapsulated additional software/hardware components. This concept is called CANELy, the **CAN Enhanced Layer** [9]. This paper addresses how the use of a FPGA[1]-oriented design can be helpful to the materialization of CANELy.

## BASIC DEPENDABILITY OF CAN

CAN is a multi-master fieldbus where bus signaling takes one out of two values: *recessive*, also the state of an idle bus; *dominant*, which always overwrites a recessive value. A *carrier sense multi-access with deterministic collision resolution* policy is used. When several nodes compete for bus access, the node transmitting the frame with the lowest identifier always goes through and gets the bus. A *frame* is a piece of encapsulated information disseminated on the network. It may contain a *message*, a user-level piece of information.

The CAN physical layer allows a few cabling faults (one wire open/short failures) to be tolerated [2, 5]. However, no standardized mechanism exists to provide resilience against network partitioning if both wires of the network cable get simultaneously interrupted.

In addition, the occurrence of certain incidents in CAN operation (e.g. bit errors) produces a subtle form of (virtual) partitioning, called *inaccessibility* [15]. Though the CAN protocol has means of recovering from these situations, that takes time,

† Faculdade de Ciências da Universidade de Lisboa, Campo Grande - Bloco C8, 1749-016 Lisboa, Portugal. Tel: +351-217500254 - Fax: +351-217500084. Navigators Home Page: http://www.navigators.di.fc.ul.pt. This work was partially supported by FCT through Project POSC/EIA/56041/2004 (DARIO)

‡ Instituto Superior Técnico – Universidade Técnica de Lisboa, Avenida Rovisco Pais, 1049-001 Lisboa, Portugal. Tel: +351-218418397 - Fax: +351-218417499. NavIST Group CAN WWW Page - http://pandora.ist.utl.pt/CAN.

[1] Field Programmable Gate Array.

leading to increase the network access delay. This may induce a violation of the expected timeliness properties [15, 3, 6, 7, 12].

Furthermore, the occurrence of some subtle errors may lead to inconsistency in error detection mechanisms and induce the failure of dependable communication protocols based on CAN operation alone [13, 9].

## ENHANCING NETWORK AVAILABILITY

In CANELy, resilience to network physical partitioning is achieved through replication of the physical path (bus medium and transceivers) used by MAC entities to communicate (Channel).

An ingenious strategy to handle replicated media has been defined in [10]. It is based on a Columbus' egg idea that extends the wired-AND nature of CAN to the media interface level. The specification of this strategy in VHDL[2] is drawn in Figure 1: the signals from the different m redundant media receivers, $M_{Rx}$ (m), are combined in a conventional AND function, before interfacing the standard MAC sub-layer, $Ch_{Rx}$. This simple solution ensures resilience to medium physical partitions and stuck-at-recessive failures [10].

```
LogicalProduct: process (M_Rx_Internal)
   variable AndResult : std_logic;
 begin
   AndResult := '1';
   for m in 1 to NumberMedia loop
     AndResult :=
        AndResult and M_Rx_Internal(m);
   end loop;
 Ch_Rx_Internal <= AndResult;
 end process LogicalProduct;
 Ch_Rx <= Ch_Rx_Internal;
```

**Figure 1: The Columbus´ egg strategy in VHDL**

The Columbus' egg strategy is the central component of the **AND-based Media Selection** design which needs to be complemented with the following functional elements:

- **frame monitoring**, performing detection of frame errors in a receiver-based bit-by-bit basis. This mechanism is fundamental to detect medium omissions;

- **media monitoring**, allowing the detection of medium omission and stuck-at failures. The operation of a given Medium should be disabled, until repair, if a Medium stuck-at-dominant failure is detected. Furthermore, Medium operation should also be disabled if it exceeds the allowed omission degree bound. The media monitoring mechanisms also includes the detection of medium partitions and stuck-at-recessive failures. Their signaling to high-level (user) management entities allows repair actions essential to the preservation of dependability coverage [9];

- **Channel monitoring**, performing early detection of Channel stuck-at dominant failures. Channel monitoring is also related with the implementation of advanced CAN-oriented *media quarantine* schemes, having the ambitious goal of providing a Channel with an omission degree bound k=1, if at least one channel media replica is unaffected by errors (permanent or transient);

- **management interface**, for interfacing with the CANELy dependability engine processor [9].

[2] Very High-Speed Integrated Circuits (VHSIC) Hardware Description Language.

The CANELy architecture allows the combination of simple media and full network redundancy in the same infrastructure, as shown in Figure 2. This can be achieved with small overhead costs: each network cable may include an additional differential pair, to support a dual-MAC interface; many low-cost controllers already include a second CAN interface. For example, the state-of-the-art Maxim/Dallas Semiconductors High-Speed Microcontroller DS80C390 integrates in a single chip a 80C52 processor core and two advanced CAN 2.0B controllers [4].

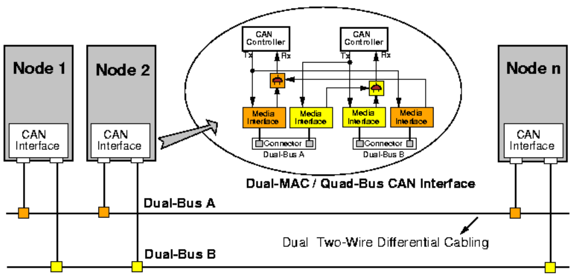
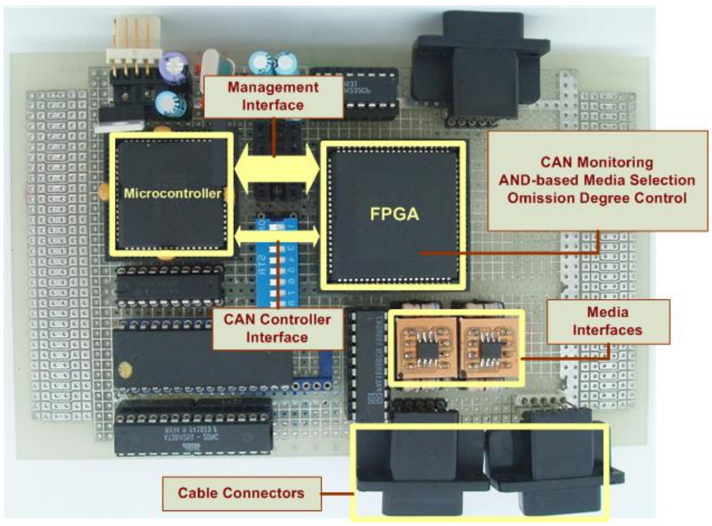


**Figure 2 : Highly-available dual-bus media redundant infrastructure**

## ENHANCING NETWORK TIMELINESS

Surprisingly the **frame monitoring** mechanisms used for improving network availability can be extended with a small-added complexity with respect to: the control of CAN inaccessibility [12]; the support to the execution of reliable group communication protocols. In addition, mechanisms have also been added to:



| CANELy Low-Level Component | FPGA Utilization Metrics | |
|---|---|---|
| | Number Xilinx Slices | Equivalent Gate Count |
| AND-based Media Selection | 1 | 6 |
| Frame Monitoring | 2 | 37 |
| Media Monitoring | 23 | 395 |
| Channel Monitoring | 20 | 344 |
| Management Interface | 22 | 384 |

**Figure 3 : CANELy prototype board and FPGA utilization metrics**

- the evaluation of the real durations of inaccessibility incidents and their upper bounds with respect to number of incidents, total duration and overall effect on the system [12];

- the signaling of inconsistent frame omissions [13, 9]. The occurrence of such kind of failures can then be taken into account in the operation of a CAN-oriented reliable protocol suite, which includes fault-tolerant reliable broadcast and group communication, fault-tolerant clock synchronization, node failure detection and membership [13, 11, 8].

## IMPLEMENTATION ISSUES

In the prototype of the CANELy architecture currently being implemented (cf. Figure 3), the CANELy processing infrastructure required for the execution of the low-level protocols (group communication [13], clock synchronization [8], node failure detection and site membership [11]) is materialized using the state-of-the-art Dallas Maxim DS80C390 High-Speed Microprocessor [4].

The support for the low-level special-purpose functions is implemented by a single FPGA [16]. The FPGA utilization for each CANELy component is given in Figure 3 with regard to *slices* and *gate count* metrics. The support for CAN enhanced timeliness and other reliability-related functions are included in the CANELy *Channel monitoring* component.

These results show that the CANEly design can be easily integrated: in a single, low complexity, programmable logic device; in existing CAN IP-cores, such as the ESA HurriCANe [14].

As an example, Table 1 provides a comparison between the FPGA areas occupied by the mechanisms relative to a complete CAN core [17] and the dependability and timeliness enforcement mechanisms, currently included in the CANELy architecture. Both designs were synthesized using the Xilinx Webpack and Xilinx Synthesis Technology (XST) tools.

| Functional Unit | FPGA Utilization Metrics | | |
|---|---|---|---|
| | Number Xilinx Slices | Equivalent Gate Count | Area (%) |
| CAN Core | 1059 | 21993 | 94.97 |
| CANELy | 68 | 1166 | 5.03 |

**Table 1 – FPGA utilization comparison**

From an engineering perspective, the practical implementation of the CANELy dependability and timeliness enforcement mechanisms in the FPGA-infrastructure obliges to address a set of particular but extremely relevant issues, such as the provision of a synchronization stage between the local network clock and the receiver incoming bit stream, as specified in the standard CAN specification [2].

In general, special attention has been given to the engineering of interfaces inherently using multiple clock domains, such as the network interface and the microprocessor management interface, in order to avoid metastable incidents.

## CONCLUSIONS

This paper has presented a practical solution for a severe problem concerning the utilization of standard CAN controllers in highly dependable real-time applications.

The approach taken aims to complement the CAN standard layer with enhanced network dependability and timeliness attributes. A FPGA-based design opens room both for: the utilization of a FPGA or other (simpler) programmable logic device in companion of existing commercial CAN controllers, e.g. the Dallas/Maxim DS80C390 High-Speed Microprocessor [4]; the integration of CANELy enhanced dependability and timeliness functions into existing IP-cores, such as the ESA HurriCANe [14].

## REFERENCES

[1] CiA - CAN in Automation. CAN Physical Layer for Industrial Applications - CiA Draft Standard 102 Version 2.0, April 1994.

[2] ISO. International Standard 11898 - Road vehicles - Interchange of digital information - Controller Area Network (CAN) for high-speed communication, November 1993.

[3] M.A. Livani, J. Kaiser, and W.J. Jia. Scheduling hard and soft real-time communication in CAN. In Proc. of the 23rd Workshop on Real-Time Programming, Shantou, China, June 1998. IFAC/IFIP.

[4] Maxim/Dallas Semiconductors. DS80C390 Dual-CAN High-Speed Microprocessor, Feb. 2005.

[5] Philips Semiconductors. TJA1053 - Fault-tolerant CAN transceiver, October 1997.

[6] L. Pinho, F. Vasques, and E. Tovar. Integrating inaccessibility in response time analysis of CAN networks. In Proceedings of the 3rd Int. Workshop on Factory Communication Systems, Porto, Portugal, September 2000. IEEE.

[7] S. Punnekkat, H. Hansson, and C. Norstrom. Response time analysis under errors for CAN. In Proceedings of the Real-Time Technology and Applications Symposium, pages 258-265, Washington, USA, May 2000. IEEE.

[8] L. Rodrigues, M. Guimarães, and J. Rufino. Fault-tolerant clock syncronization in CAN. In Proceedings of the 19th Real-Time Systems Symposium, pages 420-429, Madrid, Spain, December 1998. IEEE.

[9] J. Rufino. Computational System for Real-Time Distributed Control. PhD thesis, Technical University of Lisbon - Instituto Superior Técnico, Lisboa, Portugal, July 2002.

[10] J. Rufino, P. Veríssimo, and G. Arroz. A Columbus' egg idea for CAN media redundancy. In Digest of Papers, The 29th Int. Symposium on Fault-Tolerant Computing Systems, Madison, Wisconsin - USA, June 1999. IEEE.

[11] J. Rufino, P. Veríssimo, and G. Arroz. Node failure detection and membership in CANELy. In Proceedings of the Int. Conference on Dependable Systems and Networks, San Francisco, California, USA, June 2003. IEEE.

[12] J. Rufino, P. Veríssimo, G. Arroz, and C. Almeida. Control of inaccessibility in CANELy. In Proceedings of the 6th. Int. Workshop on Factory Communication Systems, pages 35-44, Torino, Italy, June 2006. IEEE.

[13] J. Rufino, P. Veríssimo, G. Arroz, C. Almeida, and L. Rodrigues. Fault-tolerant broadcasts in CAN. In Digest of Papers, The 28th Int. Symposium on Fault-Tolerant Computing Systems, Munich, Germany, June 1998. IEEE.

[14] L. Stagnaro. HurriCANe - Free VHDL CAN Controller core. Eupean Space Agency (ESA), Noordwijk, The Netherlands, March 2000.

[15] P. Veríssimo, J. Rufino, and L. Ming. How hard is hard real-time communication on field buses? In Digest of Papers, The 27th Int. Symposium on Fault-Tolerant Computing Systems, Washington - USA, June 1997. IEEE.

[16] Xilinx Inc. Spartan-3E FPGA Family: Complete Data Sheet, May 2007.

[17] I. Mohor, CAN Core. www.opencores.org