International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:10, No:11, 2016

# Design of Local Interconnect Network Controller for Automotive Applications

Jong-Bae Lee, Seongsoo Lee

***Abstract*—**Local interconnect network (LIN) is a communication protocol that combines sensors, actuators, and processors to a functional module in automotive applications. In this paper, a LIN ver. 2.2A controller was designed in Verilog hardware description language (Verilog HDL) and implemented in field-programmable gate array (FPGA). Its operation was verified by making full-scale LIN network with the presented FPGA-implemented LIN controller, commercial LIN transceivers, and commercial processors. When described in Verilog HDL and synthesized in 0.18 μm technology, its gate size was about 2,300 gates.

***Keywords*—**Local interconnect network, controller, transceiver, processor.

## I. Introduction

MOST electronic and electrical components in the car are connected to an electronic control unit (ECU) with wirelines. In the modern automotive applications, the number of electronic and electrical components in the car significantly increases, so the wire harness weights too much [1]. Furthermore, complicated interconnections between huge number of electronic and electrical components severely degrade communication speed. One possible solution is hierarchical interconnections. Related components are connected with local interconnections, and these clusters are further connected with gateways.

Recently, electronic and electrical components become more and more modular, and their interconnections are divided into two categories, i.e. local interconnections inside a module and global interconnections between modules. As shown in Fig. 1, LIN [2] and controller area network (CAN) [3] were presented for intra-module and inter-module communications, respectively. These two serial buses are effective for compatibility and maintenance in automotive applications, and they are standardized by International Standard Organization (ISO).

LIN bus is a low-cost sub-bus for CAN bus. It is a single-wire serial communication protocol with byte-word transmission. It is relatively low-speed (< 20 kbps), and it has typically small number of nodes (<12 nodes). Master node controls the medium access, so it needs no arbitration or collision management, while guaranteeing latency times. It has a clock synchronization mechanism, so it needs no clock generation circuit in the slave nodes. Its application covers various modules such as mirror control, window lift, door switch, door lock, control panel, wiper control, light control, light switch, seat switch, and sun roof. LIN ver. 1.3 and LIN ver. 2.0 were announced in 2002 and 2003, respectively. LIN ver. 2.2A, the newest version of LIN, was announced in 2010 [4], [5]. In this paper, a LIN controller for automotive applications was designed, implemented, and verified.

J.-B. Lee and S. Lee are with School of Electronic Engineering, Soongsil University, Seoul 06978 Korea (phone: +82-2-820-0692; e-mail: sslee@ssu.ac.kr).

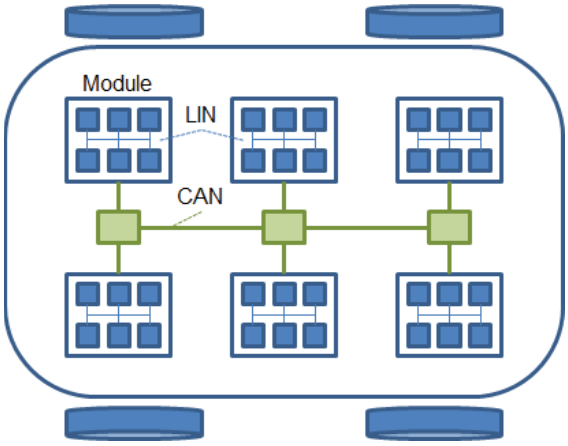


Fig. 1 LIN and CAN in the automotive applications

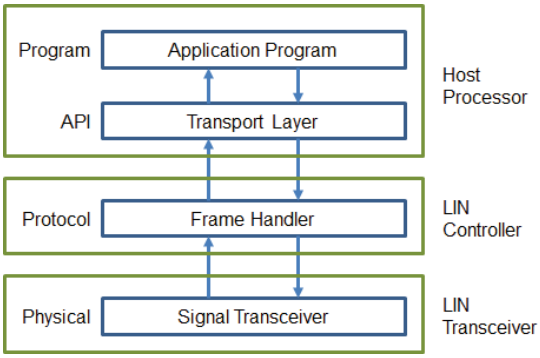


Fig. 2 LIN protocol stack

## II. LIN Controller Architecture

Fig. 2 shows the protocol stack of LIN bus. A node is connected to the physical bus wire and transmit data frame using LIN transceiver. The data frame is handled using LIN controller. There is a transport layer between program and data frame. LIN transceiver is an analog circuit controlling bus voltage and waveform. LIN controller is a digital circuit that controls the transmission of digital bits.

Fig. 3 shows the LIN network configuration. It consists of single master node and multiple slave nodes. Master task

determines which node transmits the data, and it generates frame header as shown in Fig. 4. It is implemented only in the master node. Slave task responds the frame header from master task, and it sends or receives data. It is implemented both in the master node and the slave nodes.

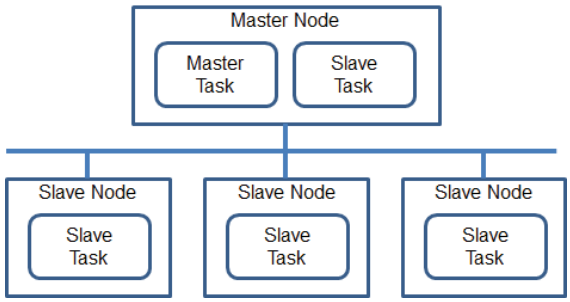


Fig. 3 LIN network configuration

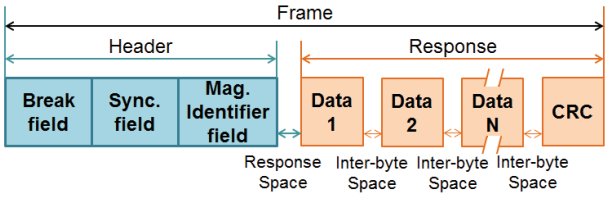


Fig. 4 LIN frame structure

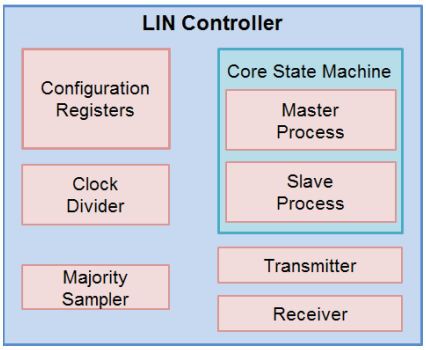


Fig. 5 The presented LIN controller

Fig. 5 shows the block diagram of the presented LIN controller. Configuration registers contain control parameters of bus operations. These registers are set by the host processor. They determine master/slave mode, transmission speed, and operation status. Clock divider generates the bus clock for data transmission from main clock. Majority sampler samples the received data from LIN transceiver and synchronizes the bus clock. It also helps accurate data detection.

Core state machine is a global controller. In master task, it generates frame header as shown in Fig. 6. In slave task, it checks the Message ID in the frame header, and determines either transmission mode (Tx mode) or reception mode (Rx mode) as shown in Fig. 7. Transmitter and receiver sends and receives the data serially with LIN transceiver.

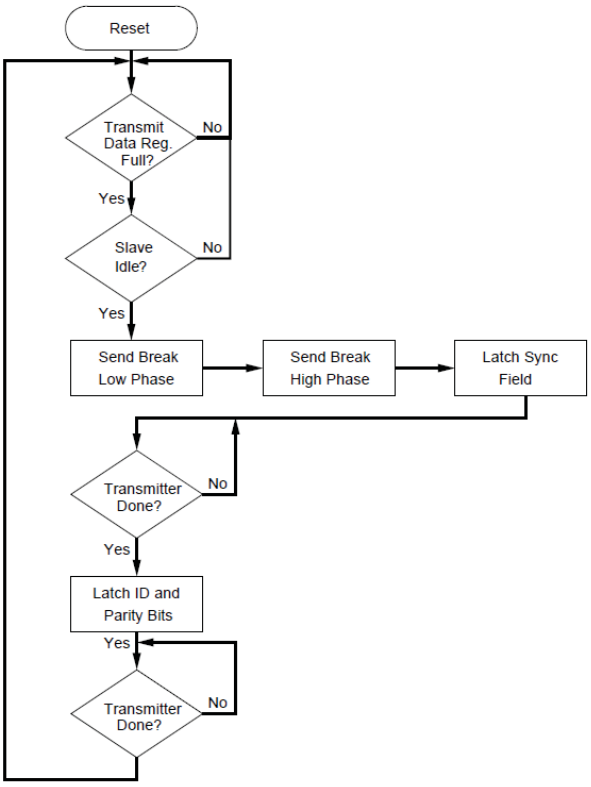


Fig. 6 LIN master task [6]

## III. LIN Controller Implementation

The presented LIN controller was described in Verilog HDL and its operation was verified by ModelSim. After it has passed the functional simulation, it was implemented in FPGA and verified in test board as shown in Fig. 8. In Fig. 8, 1 master node and 1 slave node were implemented and connected. In the master node, both master and slave tasks were programmed. In the slave node, only slave task was programmed.

In the test operation, Message ID was set to 0x20, and master node transmits 4 data bytes (0x4A, 0x55, 0x93, 0xE5) and 1 cyclic redundancy check (CRC) byte (0xE6) to slave node.

The waveform of the data frame on the bus was shown in Fig. 9. It was measured by digital oscilloscope, and it was confirmed that LIN data frame was correctly transmitted. Fig. 10 shows the received data in the slave node. Since the slave node is a small processor board without display panel, its received data was sent to and checked in the PC. As shown in Fig. 10, the received data were Message ID (0x20), data bytes (0x4A, 0x55, 0x93, 0xE5), and CRC byte (0xE6).

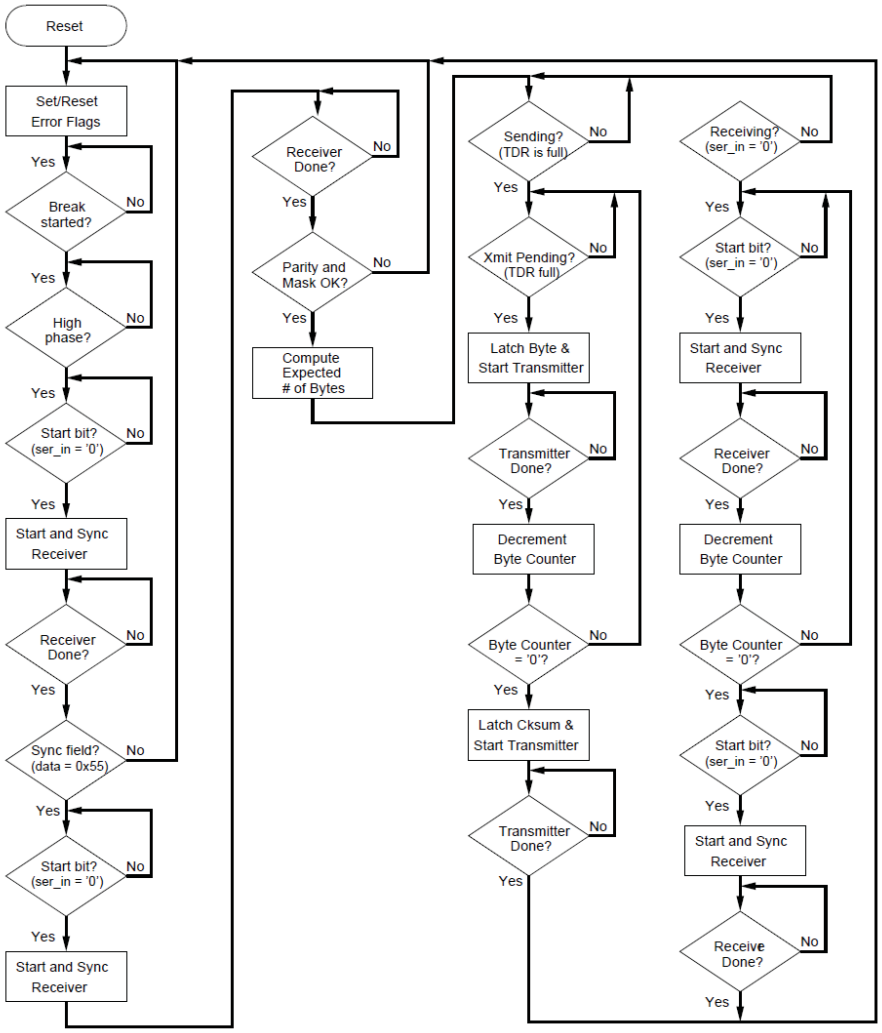


Fig. 7 LIN slave task [6]

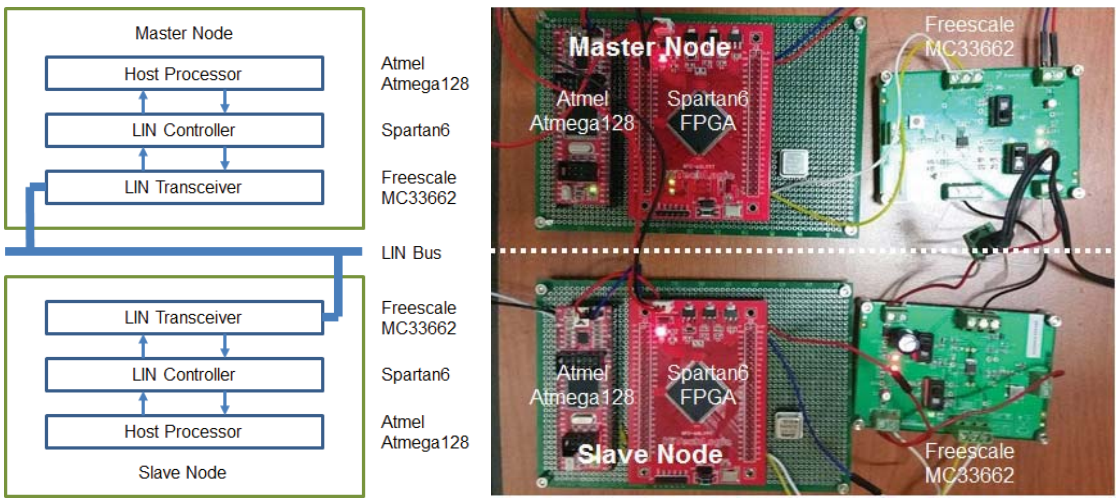


Fig. 8 LIN test board of master and slave nodes

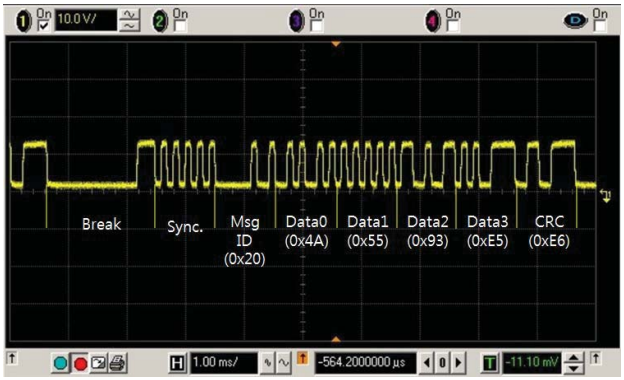


Fig. 9 LIN data frame waveform on the bus

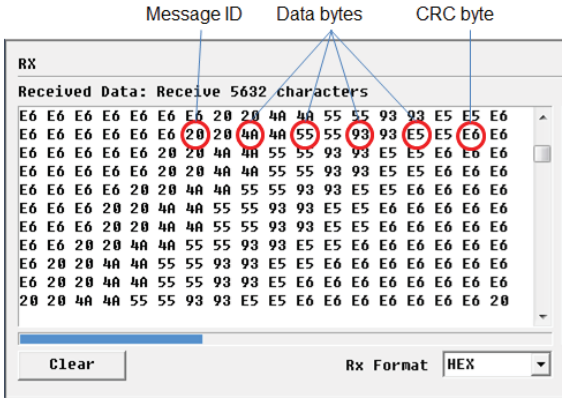


Fig. 10 LIN data received in the slave node

The designed LIN controller was synthesized in 0.18 um technology. Its gate size was about 2,300 gates, which is quite small. The designed LIN controller is a soft intellectual property (IP), so it can be applied to many automotive system-on-chips (SoC) with negligible modification.

## ACKNOWLEDGMENT

This research was supported by the Ministry of Science, ICT and Future Planning, supervised by the Institute for Information & communications Technology Promotion (R7117-16-0165). This research was also supported by Industrial Core Technology Development Program (10049095) funded by the Ministry of Trade, industry & Energy, Korea."

## REFERENCES

[1] M. Choi, S. Kim, and S. Seo, "Status and Perspective of Automotive Communication Technology", Magazine of IEIE, vol.37, no.5, pp. 23-37, May 2010.
[2] ISO SC31, ISO 11898-1:2015, "Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signaling", http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63648, (Accessed on 30 Sep. 2016).
[3] ISO SC31, ISO 17987-1:2016, "Road vehicles - Local Interconnect Network (LIN) - Part 1: General information and use case definition", http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61222, (Accessed on 30 Sep. 2016).
[4] LIN Steering Group, "LIN Specification, Revision 2.2A", www.lin-subbus.org, (Accessed on 30 Sep. 2016).
[5] J. Lee and S. Lee, "Implementation of the LIN Controller Using Verilog HDL", in Proc. of IKEEE Summer Conference, 2016, pp. 1-2
[6] XILINX, "Implementing a LIN Controller on a CoolRunner-II CPLD, XAPP432 (v1.1)", http://www.xilinx.com/support/documentation/application_notes/xapp432.pdf, (Accessed on 30 Sep. 2016).

**Jong-Bae Lee** received B.S. degree in E.E from Hanseo University, Korea, in 2009. He received M.S. degree in E.E from Soongsil University, Korea, in 2011. He is currently pursuing Ph.D. degree in E.E. Soongsil University, Korea. His research interests include high efficiency video coding, low-power SoC, multimedia SoC, power management SoC, battery management SoC, sensor SoC, and automotive SoC.

**Seongsoo Lee** received B.S, M.S, and Ph.D. degrees in E.E. from Seoul National University, Korea in 1991, 1993, and 1998, respectively. In 1998-2000, he was a research associate in Institute of Industrial Science, University of Tokyo, Japan. In 2000-2002, he was a research professor in Department of Electronic Engineering, Ewha Womans University, Korea. He joined School of Electronic Engineering at Soongsil University, Korea in 2002, where he is currently a professor. His research interests include high efficiency video coding, low-power SoC, multimedia SoC, power management SoC, battery management SoC, sensor SoC, and automotive SoC.