

차량용 LIN 제어기의 설계 및 검증

Design and Verification of Automotive LIN Controller

이종배*, 이성수*
Jong-Bae Lee*, Seongsoo Lee*

Abstract

LIN (local interconnect network) is a standard low-speed serial communication protocol, and it was developed as an efficient sub-bus for automotive electronic modules. In this paper, a LIN controller was implemented in Verilog HDL, based on LIN ver. 2.2A. The implemented LIN controller was verified in FPGA, and it can be supplied as an IP to be integrated into SoC system. Its size is about 2,300 gates when synthesized in 0.18um technology.

요약

차량 내 전자장치에서 효과적인 서브 버스로서 표준화된 저속 직렬 통신 프로토콜인 LIN(local interconnect network)이 개발되었다. 본 논문에서는 LIN 버전 2.2A를 기반으로 LIN 제어기를 Verilog HDL을 이용하여 구현하였다. 구현된 LIN 제어기는 FPGA에서 동작을 확인하였으며 IP 형태로 제공되어 SoC 시스템에 통합이 가능하다. 0.18um 공정에서 합성하였을 때의 게이트 수는 약 2,300 게이트이다.

Key words: LIN, CAN, Controller, Bus, Automotive

1. 서론

전자 기술의 발달에 힘입어 자동차에도 다양한 편의장치 및 안전장치의 장착이 급증하고 있으며, 이에 따라 차량 내부의 전자장치 사이에서 데이터 통신이 복잡해지고 통신선 길이도 길어지는 문제점이 발생하였다[1]. 또한 차량 내 전자장치가 모듈화되면서[2] 호환성 및 유지보수를 위해서도 효과적이고 표준화된 통신 프로토콜이 필요하게 되었다. 이에 따라 자동차 내의 여러 모듈을 연결하는 통신 프로토콜은 CAN(controller area network)[3] 버스가, 모듈 내에서의 통신 프로토콜은 LIN(local interconnect network)[4] 버

스가 주로 사용되고 있다.

LIN 버스는 CAN 버스를 위한 저가격 서브 버스로서의 기능을 하는 최대 20Kbps의 저속 직렬 통신 프로토콜이다. 주로 모듈 내에서 센서, 모터, 마이크로프로세서 등을 연결하며, 안전과 큰 상관없는 파워 윈도우 조작, 쉘루프 제어, 내부조명 제어 등에 많이 사용되고 있다. 2002년에 LIN 버전 1.3 스펙이 발표되었고, 2003년에는 LIN 진단 요소 및 진단 인터페이스가 정의된 LIN 버전 2.0이 발표되었다. 2010년에는 LIN 버전 2.2A[5][6]이 발표되었으며, 본 논문에서는 LIN 버전 2.2A를 기반으로 Verilog HDL을 이용하여 LIN 제어기(LIN controller)를 설계하였다.

* School of Electronic Engineering, Soongsil University

★ Corresponding author (e-mail: sslee@ssu.ac.kr, tel: 02-820-0692)

※ 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)."

Manuscript received Sep. 19, 2016; accepted Sep. 27, 2016 Sep. 28.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

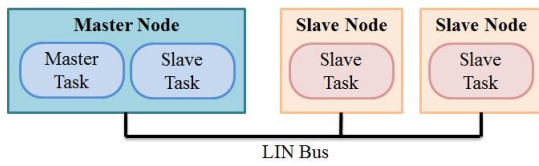


Fig. 1. LIN network configuration[5]

그림 1. LIN 네트워크 구성[5]

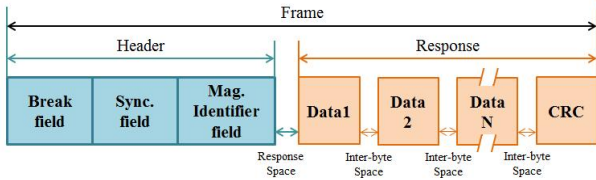


Fig. 2. LIN frame structure[5]

그림 2. LIN 프레임 구조[5]

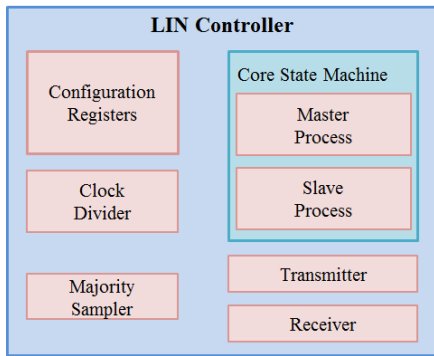


Fig. 3. LIN controller block diagram[6]

그림 3. LIN 제어기 블록도[6]

II. LIN 제어기 아키텍처

1. LIN 버스

LIN 버스는 직렬통신 버스이며 그림 1과 같이 단일 마스터(master)와 복수 슬레이브(slave)를 가진다. 마스터 태스크는 버스에서 어떤 노드가 데이터를 전송할지를 결정하여 그림 2에 나타난 프레임 헤더를 생성하며, 마스터 노드에만 탑재된다. 슬레이브 태스크는 마스터 태스크에서 전송한 프레임 헤더에 응답하여 데이터를 송수신하며, 마스터 노드와 슬레이브 노드 모두에 탑재된다.

2. LIN 제어기

본 논문에서 설계한 LIN 제어기의 구조는 그림 3과 같으며 LIN 버전 2.2A를 기반으로 Verilog HDL을 이용하여 설계하였다.

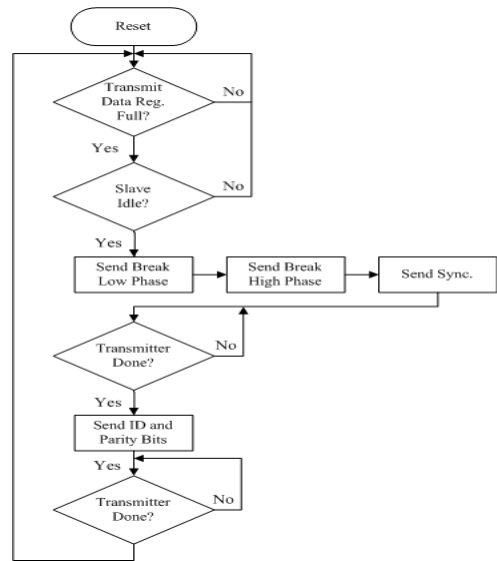


Fig. 4. LIN master process[7]

그림 4. LIN 마스터 프로세스[7]

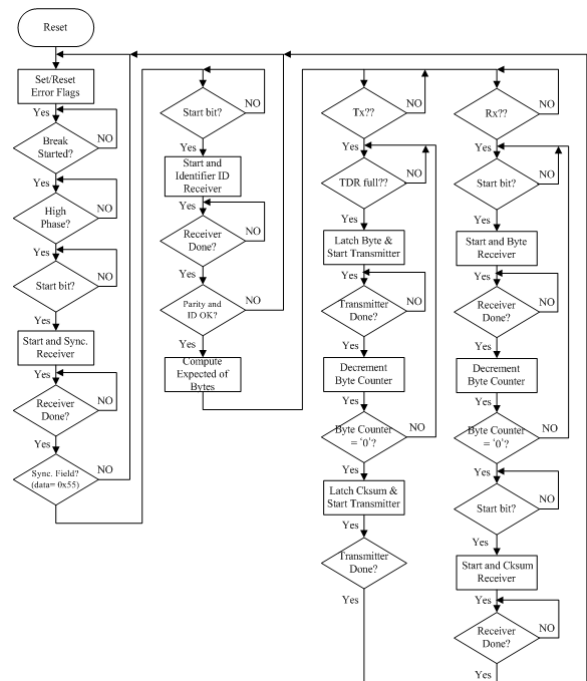


Fig. 5. LIN slave process[7]

그림 5. LIN 슬레이브 프로세스[7]

구성 레지스터(configuration register)는 LIN 버스의 제어를 위한 레지스터들이 정의되어있다. 제어기가 마스터 모드 또는 슬레이브 모드로 동작할지를 설정할 수 있으며 LIN 버스의 통신 속도를 설정하고 특정 상태를 기록할 수 있다.

클록 분주기(clock divider)는 메인 클록을 입력 받아 LIN 통신 속도에 해당하는 클록을 생성하여 필요한 블록에 공급한다.

다수 표본기(majority sampler)는 LIN 버스로부터 수신한 데이터를 샘플링하여 클록의 동기를 맞추고 데이터의 정확한 검출을 도와준다.

핵심 제어기(core state machine)는 LIN 제어기 전체의 핵심 기능을 담당하고 있으며 구성 레지스터를 마스터 모드로 설정하면 마스터 프로세스가 동작하여 프레임 헤더 부분을 생성하게 되며 그림 4의 순서도와 같이 동작한다[7]. 또한 슬레이브 모드로 설정하면 슬레이브 프로세스가 동작하여 수신한 프레임 헤더의 Identifier ID에 따라 Tx 모드 또는 Rx 모드로 진입하게 되어 그림 5와 같이 데이터를 송수신한다[7].

송신기(transmitter)는 버스로 송신하고자 하는 데이터를 직렬로 내보내며, 수신기(receiver)는 다수 표본기에서 샘플링된 직렬 수신 데이터를 필요한 각 블록에 전달한다.

III. 구현 및 검증 결과

본 논문에서 제안한 LIN 제어기는 Verilog HDL로 기술한 후 Modelsim을 이용하여 시뮬레이션하여 검증하였다. LIN 버스의 전체 동작을 확인하기 위해서는 그림 6과 같이 실험 환경을 구축하여 그림 7과 같은 테스트 보드를 제작하여 측정하였다.

그림 6에서 굵은 선으로 표시된 LIN 제어기는 본 논문에서 설계한대로 Verilog HDL로 기술하여 Xilinx사의 Spartan6-XC6SLX9 FPGA 보드에 탑재한 후 동작시켰으며, 주변 회로인 마이크로컨트롤러(MCU)와 LIN 송수신기(LIN transceiver)는 각각 상용 칩인 Atmel사의 Atmega128과 Freescale사의 MC33662를 사용하였다.

테스트 보드에서 LIN 제어기는 MCU로부터 해당 레지스터들의 설정 값을 입력받아 마스터 모드 또는 슬레이브 모드로 동작하게 된다. Identifier ID를 16진수 값 0x02으로 설정하게 되면 마스터가 슬레이브에게 4 바이트 값을 전송하게 설계되어 있다. 마스터는 검증을 위한 테스트 데이터로 0x4A, 0x55, 0x93, 0xE5 값을 차례로 전송하게 되며 마지막으로 CRC 값인 0xE6을 전송하게 된다. 그림 8은 오실로스코프를 이용하여 LIN 버스에 데이터가 정확히 전송되는지 측정할 결과이며, 정확한 데이터가 전송되고 있는 것을

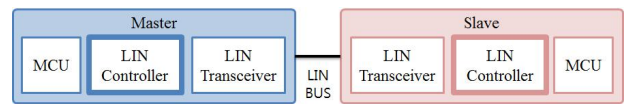


Fig. 6. Experimental environment

그림 6. 실험 환경



Fig. 7. Test board

그림 7. 테스트 보드

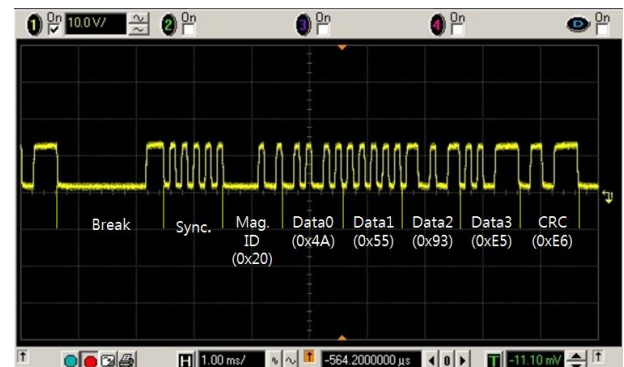


Fig. 8. LIN signal waveform measured on oscilloscope

그림 8. 오실로스코프에서 측정한 LIN 신호 파형

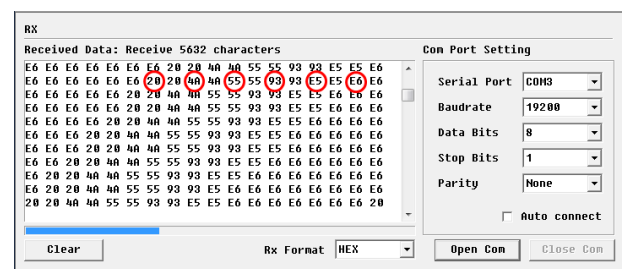


Fig. 9. Received LIN data measured on PC

그림 9. PC에서 측정한 LIN 수신 데이터

확인하였다. 이로써 LIN 제어기가 마스터 모드로의 동작을 잘 수행하는 것을 검증하였다.

또한 슬레이브 모드로 동작이 잘 되는지 확인하기 위해 LIN 제어기는 수신한 데이터를 MCU에게 전송해 MCU의 UART 통신^[8]을 이용하여

Table 1. Synthesize result

표 1. 합성 결과

Tools	Slice Logic	Used	Utilization
ISE13.1	Slice Registers	551	4%
	Slice LUTs	565	9%
	Memory	98	6%
	Occupied Slices	296	20%
	Bonded IOBs	35	34%
	RAMB16BWERs	15	46%
	BUFG/BUFGMUXs	4	25%
Tools	Gate Count		
Design Vision	about 2,300		

PC에서 수신된 데이터를 확인하였다. 그림 9와 같이 정확한 데이터가 수신되는 것을 확인할 수 있으며 LIN 제어기가 슬레이브 모드로의 동작을 잘 수행하는 것을 검증하였다.

구현한 LIN 제어기의 합성 결과는 표 1에서 확인할 수 있으며, Design Vision과 ISE를 사용하여 합성한 결과가 각각 나타나 있다. Design Vision을 사용한 결과는 0.18 μ m 공정을 사용하였으며 게이트 수는 약 2,300 게이트였다.

V. 구현 결과 및 결론

본 논문에서는 LIN 버전 2.2A를 기반으로 LIN 제어기를 Verilog HDL를 이용하여 설계 및 구현하였다. 자체 테스트 보드를 제작하여 LIN 버스에서 LIN 프레임을 통해 통신이 제대로 수행되는 것을 확인하였으며, 구현된 LIN 제어기가 마스터 및 슬레이브 모드로 정확한 동작을 수행하는 것을 확인하였다.

0.18 μ m 공정에서 합성하여 게이트 수는 약 2,300 게이트였다. 또한 IP 형태로 제공이 가능하여 LIN 제어기가 필요한 어떠한 SoC 시스템에도 적용이 가능하다.

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] H. Park, "Design Technique and Application

for Distributed Recovery Block Using the Partitioning Operating System Based on Multi-Core System," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 3, pp. 357-365, Sep. 2015.

[3] ISO SC31, ISO 11898-1:2015, "Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling," http://www.iso.org/iso/catalogue_detail.htm?csnumber=63648

[4] 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

[5] LIN Steering Group, "LIN Specification, Revision 2.2A," www.lin-subbus.org

[6] J. Lee and S. Lee, "Implementation of the LIN Controller Using Verilog HDL," in *Proc. of IKEEE Summer Conference*, 2016, pp. 1-2

[7] XILINX, "Implementing a LIN Controller on a CoolRunner-II CPLD, XAPP432 (v1.1)," http://www.xilinx.com/support/documentation/application_notes/xapp432.pdf

[8] J. Kim and S. Lee, "Design of an 8051 Microcontroller With Application-Specific Instructions and I/O Ports for Data Transmission," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 4, pp. 625-631, Dec. 2015.