ISSN:1226-7244 (Print)
ISSN:2288-243X (Online)
논문번호 17-02-11
j.inst.Korean.electr.electron.eng.Vol.21,No.2,162~165,June 2017
http://dx.doi.org/10.7471/ikeee.2017.21.2.162

차량용 CAN 제어기의 설계 및 검증

Design and Verification of Automotive CAN Controller

이 종 배*, 이 성 수*★
Jong-Bae Lee*, Seongsoo Lee*★

Abstract

CAN (controller area network) is a standard real-time serial communication protocol, and it was developed to control various in-vehicle electronic modules. In this paper, a CAN controller was designed in Verilog HDL, based on CAN ver. 2.0A and 2.0B. The designed CAN controller was implemented in FPGA, and it was verified its operation by connecting commercial chips. Its size is about 7,800 gates when synthesized in 0.18um technology.

요 약

차량 내 다양한 전자 장치를 제어하기 위해 실시간 직렬 통신 프로토콜인 CAN(controller area network)이 개발되었다. 본 논문에서는 Verilog HDL을 이용하여 CAN 버전 2.0A, 2.0B를 만족하는 CAN 제어기를 설계하였다. 설계된 CAN 제어기는 FPGA로 구현하여 상용 칩과 연결하여 동작을 확인하였다. 0.18um 공정에서 합성하였을 때의 게이트 수는 약 7,800 게이트이다.

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

Ⅰ. 서론

차량 내의 다양한 전자 제어 장치(electronic control unit: ECU)들의 증가로 배선 케이블의 숫자와 길이가 늘어남에 따라 차체의 무게 및 비용의 증가를 해결하고자 CAN (controller area network)[1] 통신이 개발되었다. 1986년 벤츠(Benz)의 요구로 보쉬(Bosch)에 의해 최초로 개발되었고 1991년 CAN 2.0 표준 규격이 발표되었으며, 현재 ISO (International Organization for Standardization))에 의해 ISO 11898 표준으로 지정되어있다[2]. CAN 버스는 다중 마스터 직렬 버스 프로토콜로서 차량 내부의 주요 전자 모듈인 파워 트레인(엔진, 추진축), 샤시 (스티어링, 브레이크 등), 안전 장치(에어백, 액티브 서스펜션) 등을 연결하는 데이터 버스로 사용되고 있다[3]. CAN 버스는 실시간 다중 마스터 동작, 다중 노드 동시 데이터 전송, 빠른 데이터 전송 속도, 뛰어난 오류 검출 및 보정 기능, 전기적 잡음 환경에서 높은 신뢰성 등의 장점을 가지고 있어서 자동차 내부 네트워크(in-vehicle network: IVN)에서 널리 사용되고 있다.

본 논문에서는 CAN 버전 2.0 Part A, Part B를 만족하는 CAN 제어기를 Verilog HDL을 이용하여 설계하고 이를 FPGA로 구현하였으며, 상용 칩을 사용하여 동작을 검증하였다.

* 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 (2016-0-00136)."
Manuscript received May. 25, 2017; received Jun. 23. 2017; accepted Jun. 27. 2017

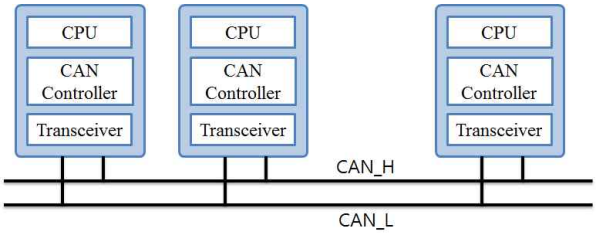

Fig. 1. CAN network configuration [1]
그림 1. CAN 네트워크 구성[1]

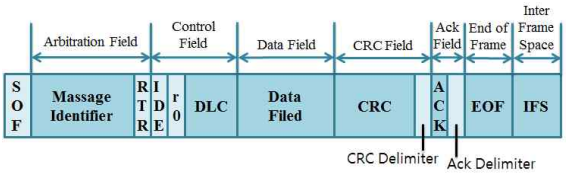

Fig. 2. CAN 2.0A frame structure [1]
그림 2. CAN 2.0A 프레임 구조[1]

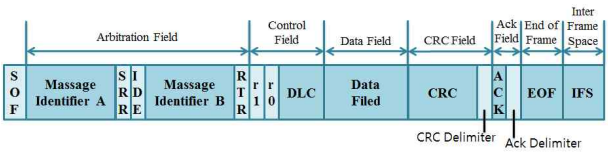

Fig. 3. CAN 2.0B frame structure [1]
그림 3. CAN 2.0B 프레임 구조[1]

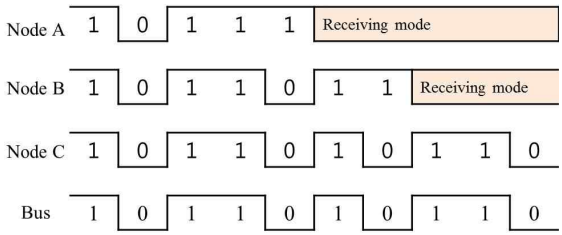

Fig. 4. Bus arbitration method based on message identifier
그림 4. 메시지 식별자 기반의 버스 중재 방법

II. CAN 제어기 아키텍쳐

1. CAN 버스

CAN 버스는 그림 1과 같은 직렬 통신 버스이며 꼬인 두 선(twisted pair), 다중 마스터(multi-master), 차동 신호(differential signal) 방식으로 동작한다. 버스 상에 연결되어 있는 모든 노드가 마스터가 되어 버스가 비어있을 때 언제든지 메시지를 전송할 수 있다. CAN 통신 프로토콜의

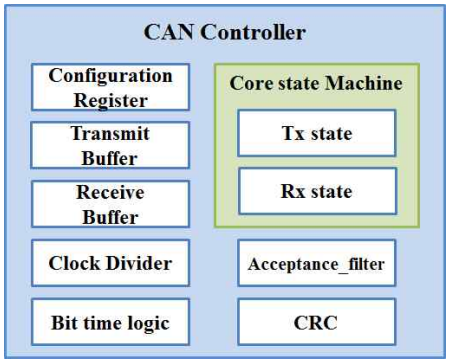

Fig. 5. CAN controller block diagram
그림 5. CAN 제어기 블록도

데이터 프레임(data frame)은 그림 2와 같이 ISO 표준으로 지정된 버전 2.0A와 그림 3과 같이 버전 2.0A를 확장한 형태인 2.0B를 가지며, 각각 11비트 식별자(identifier)와 29비트 식별자를 사용한다. 식별자는 메시지의 형태를 식별시켜주는 역할과 메시지에 우선순위를 부여하는 역할을 수행한다. 만일 두 개의 노드에서 메시지를 동시에 전송하려고 시도하더라도 그림 4와 같이 우선순위가 높은 식별자를 가진 노드의 전송이 먼저 완료되므로 버스의 충돌을 회피하게 된다. 이외에도 CAN 통신 프로토콜은 동일한 식별자를 갖는 데이터 프레임의 전송을 상대 노드에 요청하는 리모트 프레임(remote frame), 오류가 검출되었을 때 전송되는 오류 프레임(error frame), 송신 측에게 다음 프레임의 송신을 잠시 지연하도록하는 오버로드 프레임(overload frame)등 4종류의 프레임이 정의되어 있으며 본 논문에서 설계한 CAN 제어기에서는 4종류의 프레임을 모두 지원하도록 설계하였다.

2. CAN 제어기

본 논문에서 설계한 CAN 제어기의 구조는 그림 5와 같으며 CAN 버전 2.0A, 2.0B를 모두 만족하도록 Verilog HDL을 사용하여 설계하였다.

구성 레지스터(configuration register)는 CAN 제어기의 설정을 위한 레지스터를 정의한다. 제어기가 발신 모드(Tx) 또는 수신 모드(Rx)로 동작할지를 설정할 수 있으며 CAN 버스의 통신 속도를 설정하고 특정 상태를 기록할 수 있다.

클록 분주기(clock divider)는 메인 클록을 입력 받아 CAN 통신에서 가장 작은 시간 단위인

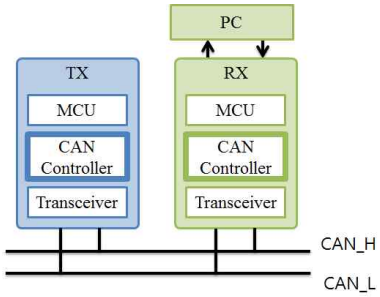

Fig. 6. Experimental environment
그림 6. 실험 환경

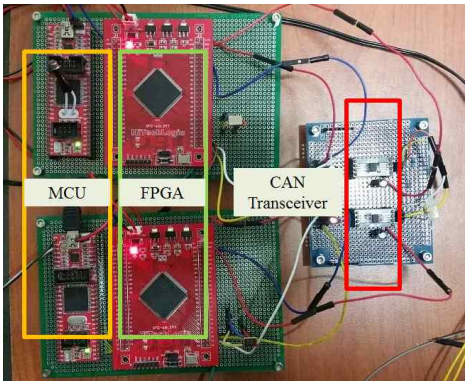

Fig. 7. Test board
그림 7. 테스트 보드

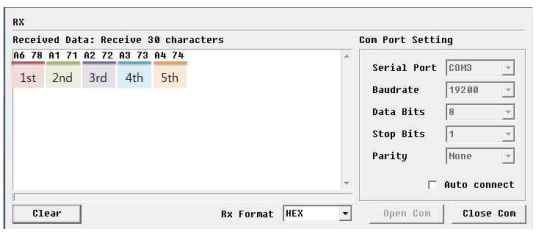

Fig. 8. Received CAN data measured on PC
그림 8. PC에서 측정한 CAN 수신 데이터

Fig. 9. CAN signal waveform measured on oscilloscope
그림 9. 오실로스코프에서 측정한 CAN 신호 파형

시간 퀀텀을 설정하고 해당하는 클록을 생성하여 필요한 블록에 공급한다.

순환 중복 검사(cyclic redundancy check: CRC)는 프레임 시작(start of frame)부터 데이터 필드(data field) 마지막까지 식 (1)의 CRC 다항식을 이용하여 체크섬(checksum)을 계산한다.

$$X^{15}+X^{14}+X^{10}+X^{8}+X^{7}+X^{4}+X^{3}+1 \quad (1)$$

수용 필터(acceptance filter)는 구성 레지스터에서 설정된 식별자와 수신 프레임의 식별자를 확인 후 비교하여 일치 여부를 가려내어 수신 데이터의 수용 여부를 결정 하게 된다.

비트 시간 로직(bit time logic)은 비트 동기화를 수행하는 블록이다. 버스에 연결된 제어기들에서 오실레이터의 클록 주파수는 오차 범위 내에서 서로 다를 수 있으며, 이 오차를 보상하여 정확하게 비트를 동기화하고 샘플 시점과 전송 시점을 계산하여 필요한 블록에 공급한다.

송신 버퍼(transmit buffer)는 버스로 송신하고자 하는 데이터를 저장하며, 수신 버퍼(receive buffer)는 수신 데이터를 저장한다.

코어 제어기(core state machine)는 CAN 제어기 핵심 기능을 담당하고 있으며 구성 레지스터를 송신 모드로 설정하면 해당 모드에 따라 데이터 프레임, 리모트 프레임, 오버로드 프레임을 생성하여 전송하게 된다. 또한 오류 상태를 체크하여 오류 프레임을 생성 및 전송한다. 수신 모드로 설정하게 되면 수신된 프레임의 식별자가 수용 필터와 일치하는지에 따라 데이터를 수신 버퍼에 저장하여 수신을 완료 하거나 수신 상태로 천이하게 된다.

III. 구현 및 검증 결과

본 논문에서는 Verilog HDL 사용하여 CAN 제어기를 설계한 후 IDEC (IC Design Education Center)에서 제공한 ModelSim을 이용하여 시뮬레이션하였다. 또한 CAN 버스의 동작 검증을 위해 그림 6과 같이 실험 환경을 구축하였고 그림 7과 같은 테스트 보드를 제작하여 확인하였으며 제작한 보드에 사용한 대표 부품은 표 1과 같다.

그림 6에서 굵은 선으로 표시된 CAN 제어기는 본 논문에서 설계한 것으로 Verilog HDL로 설계하여 FPGA 보드에 다운로드 후 동작시켰다.

그림 7의 테스트 보드에서 CAN 제어기는

Table 1. Board parts
표 1. 보드 부품

	MCU	FPGA	Transceiver
Manufacturer	Atmel	Xilinx	NXP
Part	Atmega128	Spartan6 (XC6SLX9)	MC33901

Table 2. Synthesize result
표 2. 합성 결과

Tools	Slice Logic	Used	Utilization
ISE13.1	Slice Registers	582	5%
	Slice LUTs	685	11%
	Occupied Slices	289	20%
	Bonded IOBs	31	30%
	BUFG/BUFGMUXs	1	6%
Tools	Gate Count		
Design Compiler	about 7,834		

MCU (microcontroller unit)로부터 동작에 필요한 레지스터들의 설정 값들을 입력받아 트랜스미터 모드 또는 리시버 모드로 동작하게 된다. 테스트를 위해 프레임의 컨트롤 필드의 4비트 DLC (data length code) 값을 2진수 '0010'으로 설정하여 2바이트 데이터를 송수신 하게 설정하였다. 또한 설계 되어진 CAN 제어기가 트랜스미터로 동작할 때 전송 할 데이터를 오류 없이 송신이 완료되어지면 송신 완료 인터럽트를 발생 시켜 입출력 핀을 통해 MCU에게 알려준다. 인터럽트 신호를 받은 MCU는 다음 전송할 2바이트 데이터를 CAN 제어기의 송신 버퍼에 저장하여 송신 준비를 하고 차례로 전송한다. 버스에 연결된 노드들의 송수신 동작이 잘 되는지 확인하기 위해 트랜스미터로 동작하는 CAN 제어기는 2바이트 데이터 (0xA6, 0x78), (0xA1, 0x71), (0xA2 0x72), (0xA3 0x73), (0xA4 0x74)을 차례로 다섯 번 전송 하였다. 리시버로 동작하는 CAN 제어기는 수신한 프레임의 식별자를 확인해 일치하면 데이터를 수신 버퍼에 저장하게 되고 MCU에게 전송해 MCU의 UART 통신을 이용하여 PC에서 수신된 데이터를 확인하였다.

그림 8은 PC에서 수신된 CAN 데이터를 나타낸 것으로 2바이트 데이터가 다섯 번 정확히 수신되는 것을 확인할 수 있으며 CAN 제어기의 송수신 동작이 잘 수행하는 것을 검증하였다.

그림 9는 CAN 버스를 오실로스코프를 이용하여 측정한 것이며 가장 위쪽 빨간색 신호는 CAN 버스 High 신호 이고 가장 아래쪽 노란색 신호는 CAN 버스 Low 신호이며, 가운데 초록색 신호는 CAN_H와 CAN_L 신호에 차동 신호를 측정한 것이다.

구현한 CAN 제어기의 합성 결과는 표 2에서 확인할 수 있으며, IDEC에서 제공한 Synopsys사의 Design Compiler와 Xilinx사의 ISE를 사용하여 합성한 결과가 각각 나타나 있다. Design Compiler을 사용한 결과는 0.18㎛ 공정을 사용하였으며 게이트 수는 약 7,834 게이트였다.

Ⅴ. 구현 결과 및 결론

본 논문에서는 CAN 버전 2.0A 및 2.0B를 만족하는 CAN 제어기를 Verilog HDL를 이용하여 설계 및 구현하였다. CAN 제어기의 검증을 위해 자체 테스트 보드를 제작하여 CAN 버스에서 CAN 제어기의 정확한 송수신이 되는 것을 확인하였으며, 송신 또는 수신 모드에서의 동작을 확인하였다.

설계된 CAN 제어기를 0.18㎛ 공정에서 합성하여 게이트 수는 약 7,834 게이트였다. 또한 IP 형태로 제공이 가능하여 다양한 SoC 시스템에 내장할 수 있다.

References

[1] Bosch, "CAN Literature", http://www.bosch-semiconductors.de/en/automotive_electronics/ip_modules/can_literature_2.html

[2] 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

[3] J. Lee and S. Lee, "Design and Verification of Automotive LIN Controller," *j.inst.Korean.electr.electron.eng*, vol. 20, no. 3, pp. 333-336, 2016.DOI:10.7471/ikeee.2016.20.3.333