

# 차량용 CAN-FD 제어기의 구현 및 검증

## Implementation and Verification of Automotive CAN-FD Controller

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

### Abstract

CAN (controller area network) suffers from data bottleneck since the number of in-vehicle electronic modules significantly increases. To mitigate this problem, CAN-FD (CAN with flexible data rate) has been proposed. Transmission speed is same with CAN in arbitration phase but much higher than CAN in data phase, which successfully achieves both compatibility and efficiency. In this paper, a CAN-FD controller was designed in Verilog HDL and it was implemented and verified in FPGA. The designed controller can perform CAN-FD version 1.0 and CAN version 2.0A, 2.0B. Synthesized in 0.18um technology, its size is about 46,300 gates.

### 요약

차량 내부의 전자 장치가 급증함에 따라 CAN(controller area network)에 데이터 병목 현상이 발생하기 시작했다. 이에 따라 CAN을 개량한 CAN-FD(CAN with flexible data rate) 버스가 개발되었는데, 버스 중계 단계(arbitration phase)에서는 CAN과 동일한 속도로 전송되 데이터 전송 단계(data phase)에서는 훨씬 빠른 속도로 전송함으로써 호환성과 효율성을 모두 높였다. 본 논문에서는 CAN-FD 규격 1.0과 CAN 규격 2.0A, 2.0B를 모두 만족하는 CAN-FD 제어기를 Verilog HDL를 사용하여 설계하고 FPGA로 구현한 뒤 동작을 검증하였다. 0.18um 공정을 사용하여 합성한 결과는 약 46,300 게이트이다.

*Key words: CAN-FD, CAN, Controller, Bus, Automotive*

### 1. 서론

대표적인 차량용 네트워크로 표 1의 LIN (local interconnect network)[1][2], CAN (controller area network)[3][4], FlexRay[5][6] 등이 있다. 이 중에서 CAN은 실시간 직렬 버스로서 파워트레인, 샤시 등에서 핵심적으로 사용되고 있다. 데이터 통신량이 증가하면서 CAN의 전송 속도가 모자라게 되었으나 이를 다른 프로토콜로 대체하기 쉽지 않아서 CAN을 확장한 CAN-FD (CAN with flexible data rate)[3][7]가 개발되었다. CAN-FD는 CAN과 높은 호환성을 유지하면서도 전송 효율을 크게 향상시켰으며 CAN 국제 표준인 ISO 11898에 추가되었다[3].

본 논문에서는 Verilog HDL로 CAN-FD 제어기를 설계하였으며 이를 FPGA로 구현한 후 상용 칩과 연결하여 검증하였다.

\* School of Electronic Engineering, Soongsil University

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

※ Acknowledgment

“This work was supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean government (Motie: Ministry of Trade, Industry&Energy, HRD Program for Software-SoC convergence) (No. N0001883).”

Manuscript received Sep. 05, 2017; accepted Sep. 19, 2017.

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.

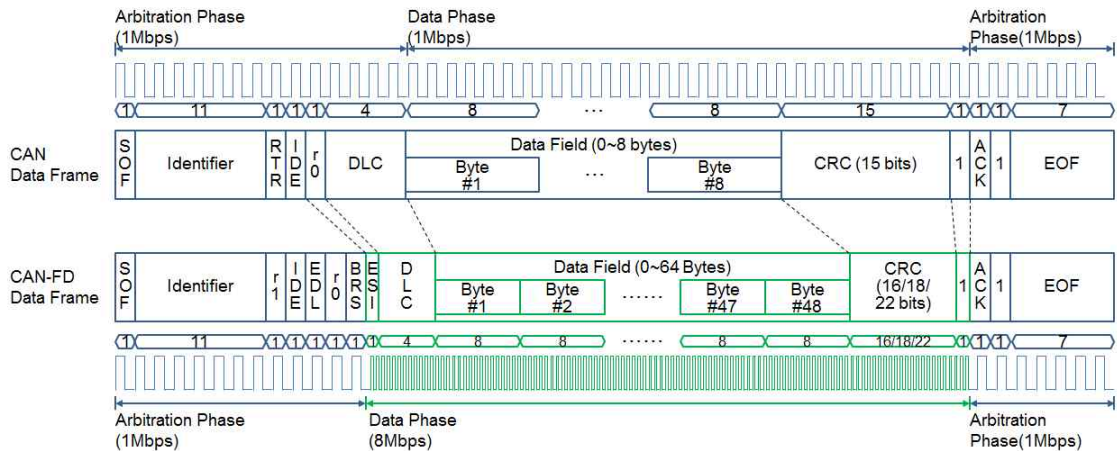


Fig. 1. Data frame structures of CAN (1 Mbps speed, 8 bytes payload) and CAN-FD (8 Mbps speed, 64 bytes payload)  
 그림 1. CAN(1 Mbps 속도, 8 바이트 페이로드)과 CAN-FD(8 Mbps 속도, 64 바이트 페이로드)의 데이터 프레임 구조

## II. CAN-FD 버스

CAN-FD는 CAN과 높은 호환성을 가지고 있어서 차량 내 네트워크 환경을 크게 변화시키지 않아도 되는 장점을 갖는다. CAN-FD는 한 프레임 내에서 전송 할 수 있는 데이터 길이, 즉 페이로드(payload)를 최대 8 바이트에서 최대 64 바이트로 확장하였고 데이터 전송 속도도 최대 1 Mbps에서 최대 8 Mbps로 증가하였다.

CAN-FD는 CAN을 기반으로 설계되었기 때문에 그림 1과 같이 버스 중재 단계(arbitration phase)까지는 기존 CAN bus와 동일한 프로토콜, 속도, 파형을 가진다. 따라서 버스 상에서 CAN 데이터 프레임과 CAN-FD 데이터 프레임을 공존하여 사용할 수 있다. 버스 중재 단계에서 데이터 프레임의 종류를 판별해서 CAN-FD이면 데이터 전송 단계(data phase)에서 버스 속도가 CAN-FD 속도로 증가했다가 다시 버스 중재 단계로 돌아오면 CAN 속도로 원위치한다.

그림 1의 CAN-FD 데이터 프레임은 CAN과

유사하지만, 전송 속도에 따라 버스 중재 단계(arbitration phase)와 데이터 전송 단계(data phase)로 나누고, 제어 필드에서 EDL, BRS, ESI 비트를 추가하고 DLC 비트를 수정하였다.

- EDL(extended data length): CAN과 CAN-FD 형식의 프레임을 구분한다. 우성(0)이면 CAN이며, 열성(1)이면 CAN-FD이다.
- BRS(bit rate switch): 데이터 전송 속도를 전환할 것인지 여부를 결정한다. 우성이면 버스 중재 속도로 그대로 유지되며, 열성이면 사전에 정의된 데이터 전송 속도로 전환된다.
- ESI(error state indicator): 현재 노드의 에러 상태를 나타낸다. 우성이면 에러 능동 상태(active error)를 의미하고 열성이면 에러 수동 상태(passive error)를 의미한다.
- DLC (data length code) : 데이터 필드의 데이터 길이를 나타낸다. CAN에서 사용하지 않는 0b1001~0b1111에 각각 12, 16, 20, 24, 32, 48, 64바이트의 데이터 길이를 할당하였다.

## III. CAN-FD 제어기 아키텍처

본 논문에서 설계한 CAN-FD 제어기의 구조는 그림 2와 같으며 CAN 규격 2.0A, 2.0B 및 CAN-FD 규격 1.0을 모두 만족한다.

설정 레지스터(configuration register)에서는 CAN-FD 제어기의 설정을 위한 레지스터들을 정의한다. CAN 또는 CAN-FD 등의 동작 모드를 설정할 수 있으며 일반 프레임, 확장형 프레임, 통신 속도, 데이터 길이, 인터럽트 등과 제어에 필요한 전반적인 것들을 설정할 수 있다.

Table 1. Comparison of automotive networks

표 1. 차량용 네트워크 방식의 비교

	LIN	CAN	FlexRay
Standard	ISO 17987	ISO 11898	ISO 17458
Application	Low-Cost Low-Speed	Soft Real-Time	Hard Real-Time
Used in	Body	Chassis, Powertrain	Chassis, Powertrain
Bandwidth	20 kbps	1 Mbps	10 Mbps
Bus Access	Polling	CSMA/CA	TDMA
Bus Wire	Single Wire	Differential	Differential

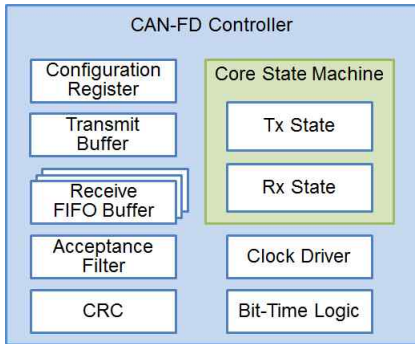


Fig. 2. CAN-FD controller block diagram  
그림 2. CAN-FD 제어기 블록도

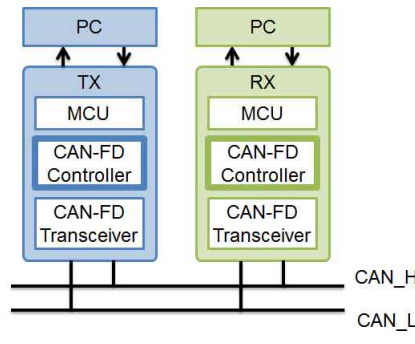


Fig. 3. Experimental environment  
그림 3. 실험 환경

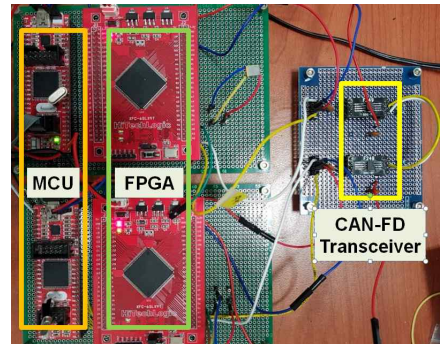


Fig. 4. Test board  
그림 4. 테스트 보드

송신 버퍼(transmit buffer)와 수신 버퍼(receive fifo buffer)는 각각 송신 및 수신 데이터를 저장하며 수신 버퍼는 FIFO 구조를 가진다. 수용 필터(acceptance filter)는 수신 모드에서 수신 프레임의 식별자 ID가 설정 레지스터와 일치하는지에 따라 데이터 수용 여부를 결정한다.

CRC(cyclic redundancy check)[8]는 CRC 다항식을 사용하여 프레임 시작부터 데이터 필드 마지막까지 체크섬을 계산한다. 페이로드가 8바이트 이하일 때는 식(1)을, 9~16바이트일 때는 식(2)를, 16바이트 보다 클 때는 식(3)을 사용한다.

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

$$CRC_{17} = X^{17} + X^{16} + X^{14} + X^{13} + X^{11} + X^6 + X^4 + X^3 + X + 1 \quad (2)$$

$$CRC_{21} = X^{21} + X^{20} + X^{13} + X^{11} + X^7 + X^4 + X^3 + 1 \quad (3)$$

클록 분주기(clock divider)에서는 설정 레지스터에서 설정되어진 CAN-FD 통신의 가장 작은 시간 단위인 시간 쿼텀(time quantum)에 해당하는 클록을 메인 클록으로부터 생성한다.

비트 시간 로직(bit time logic)에서는 비트 동기화를 수행한다. 버스에 연결된 제어기들에 사용되어지는 오실레이터의 클록 주파수의 오차를 보정하여 정확한 비트 동기를 수행한다. 또한 버스 중재 단계와 데이터 전송 단계의 통신 속도를 계산하여 필요한 블록에 공급한다. CAN-FD의 최대 전송 속도는 8 Mbps이지만 현재 상용화된 칩은 최대 2 Mbps를 지원하는 것이 주를 이루고 있다. 본 논문에서 설계된 CAN-FD는 최대 4 Mbps까지 지원이 가능하도록 설계되었다.

핵심 제어기(core state machine)는 CAN-FD 제어기의 핵심 기능을 담당한다. 설정 레지스터의 모드 설정에 따라 CAN 제어기 및 CAN-FD 제어기로서 동작 할 수 있다. 각 제어기를 다시 송신 모드로 설정하면 CAN 제어기 모드일 때는

설정되어진 모드에 따라 데이터 프레임과 리모트 프레임 및 오버로드 프레임을 생성하여 전송하고 CAN-FD 제어기 모드일 때는 데이터 프레임과 오버로드 프레임을 생성하여 전송한다. 또한 오류 상태가 발견되는 경우에는 오류 프레임을 생성하여 전송한다. 수신 모드일 때는 수신된 프레임의 식별자와 수용 필터의 일치 여부를 확인한 후 일치하면 데이터를 수신 버퍼에 저장한 후 수신을 완료하고, 일치하지 않으면 수신 상태로 돌아가 다음 프레임의 수신을 기다리게 된다.

#### IV. 구현 및 검증

본 논문에서 설계한 CAN-FD 제어기는 Verilog HDL과 ModelSim을 사용하여 설계한 후 FPGA로 구현하였다. 또한 그림 3의 실험 환경에 따라 그림 4와 같이 테스트 보드를 제작하여 CAN-FD 제어기의 동작을 검증하였다.

그림 4의 테스트 보드에서 CAN-FD 제어기는 MCU(microcontroller unit)로부터 동작에 필요한 레지스터들의 설정 값들을 입력받아 CAN-FD 모드 또는 CAN 모드로 동작하게 된다. CAN-FD에서 확장된 데이터 길이를 테스트하기 위해 4 비트 DLC 값을 2진수 '1111'으로 설정하여 64 바이트 데이터를 송수신하게 설정하였다.

송신 모드에서는 CAN-FD 제어기가 오류 없이 송신을 완료하면 송신 완료 인터럽트 신호를 발생시켜 MCU에게 알려준다. MCU가 인터럽트 신호를 받으면 다음 데이터를 CAN-FD 제어기의 송신 버퍼에 저장하여 송신을 준비하고 전송 시작 신호를 받아 그림 5와 같이 차례로 전송한다. 수신 모드에서는 CAN-FD 제어기가 그림 6에 나타난 ID와 같은 수신 프레임의 식별자 ID를

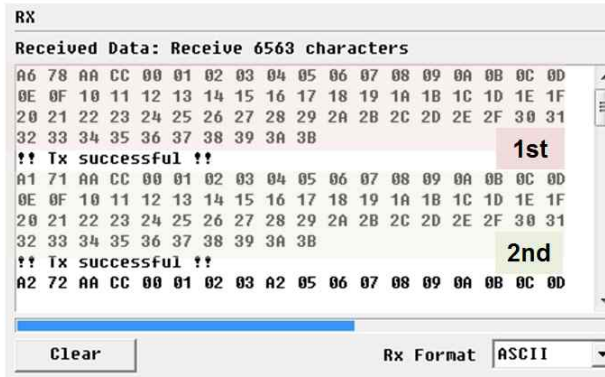


Fig. 5. Transmitted CAN-FD data measured on PC

그림 5. PC에서 측정된 CAN-FD 송신 데이터

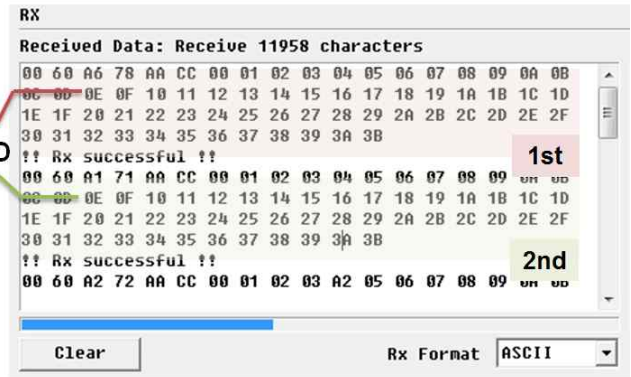


Fig. 6. Received CAN-FD data measured on PC

그림 6. PC에서 측정된 CAN-FD 수신 데이터

확인해 일치하면 데이터를 수신 버퍼인 FIFO 메모리에 저장하게 된다. MCU는 수신 데이터가 저장되어 있는 FIFO를 읽어 그림 6과 같은 데이터를 PC에 전송하였으며, 송수신이 정상적으로 수행되었음을 검증할 수 있다.

구현한 CAN-FD 제어기의 합성 결과는 표 2와 같다. Design Compiler와 0.18 $\mu$ m 공정을 사용하여 합성한 결과는 약 46,290 게이트였다.

### V. 결론

본 논문에서는 Verilog HDL를 사용하여 CAN-FD 규격 1.0와 CAN 규격 2.0A, 2.0B를 만족하는 CAN-FD 제어기를 설계 및 구현하였다. CAN-FD 제어기의 검증을 위해 상용 칩들과 연결하여 CAN-FD 버스에서 CAN-FD 제어기가 정확히 데이터를 송수신하는 것을 확인하였다.

설계된 CAN-FD 제어기를 0.18 $\mu$ m 공정에서 합성한 결과는 약 46,290 게이트로 MCU 등에 내장하기에 적절한 크기이며, 소프트 IP 형태라서 쉽게 다른 시스템에 내장할 수 있다.

Table 2. Synthesize result

표 2. 합성 결과

Tools	Slice Logic	Used	Utilization
ISE13.1	Slice Registers	1,617	14%
	Slice LUTs	1,848	32%
	Occupied Slices	725	50%
	Bonded IOBs	31	30%
	BUFG/BUFGMUXs	1	6%
Tools	Gate Count		
Design Compiler	about 46,290		

### References

- [1] ISO 17987-1:2016, "Road Vehicles-Local Interconnect Network (LIN)-Part 1: General Information and Use Case Definition", <https://www.iso.org/standard/61222.html>
- [2] J. Lee and S. Lee, "Design and Verification of Automotive LIN Controller", *j.inst.Koreanelectr.electroneng*, vol. 20, no. 3, pp. 333-336, 2016.
- [3] ISO 11898-1:2015, "Road Vehicles-Controller Area Network (CAN)-Part 1: Data Link Layer and Physical Signalling", <https://www.iso.org/standard/63648.html>
- [4] J. Lee and S. Lee, "Design and Verification of Automotive CAN Controller", *j.inst.Koreanelectr.electroneng*, vol. 21, no. 2, pp. 162-165, 2017.
- [5] ISO 17458-1:2013, "Road vehicles-FlexRay Communications System-Part 1: General Information and Use Case Definition", <https://www.iso.org/standard/59804.html>
- [6] S. Shreejith and S. Fahmy, "Extensible FlexRay Communication Controller for FPGA-Based Automotive Systems", *IEEE Trans. Veh. Technol.*, vol. 64, no. 2, pp. 453-465, 2015.  
DOI : 10.1109/TVT.2014.2324532
- [7] CAN in Automation, "CAN-FD: The basic idea", <https://www.can-cia.org/can-knowledge/can/can-fd/>
- [8] J. Lee, "Low-Cost CRC Scheme by Using DBI (Data Bus Inversion) for High Speed Semiconductor Memory", *j.inst.Koreanelectr.electroneng*, vol. 19, no. 3, pp. 288-294, 2015.  
DOI : 10.7471/ikeee.2015.19.3.288