ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 23-04-41
j.inst.Korean.electr.electron.eng.Vol.27,No.4,668~678,December 2023
https : //doi.org/10.7471/ikeee.2023.27.4.668
301

FlexRay: 프로토콜, 시간 계층, 메시지 프레임, 커뮤니케이션 컨트롤러, 적합성 시험

FlexRay: Protocol, Time Hierarchy, Message Frame, Communication Controller, and Conformance Test

한 석 준*, 신 수 아*, 박 나 은*, 박 찬*, 이 대 기*, 이 성 수*★

Seokjun Hahn*, Sua Shin*, Naeun Park*, Chan Park*, Daegi Lee*, and Seongsoo Lee*★

Abstract

FlexRay is an in-vehicle network with maximum two channels and maximum transmission speed of 10Mbps per channel. FlexRay exploits TDMA (Time Division Multiple Access) and FTDMA (Flexible Time division Multiple Access) to ensure real-time communication with efficient transmission, so it is used for real-time electronic control of safety-critical vehicular modules such as powertrain. This paper explains FlexRay protocol, time hierarchy, message frame, communication controller, and conformance test in detail based on ISO 17458 standard and FlexRay consortium documents.

요 약

FlexRay는 차량 내 네트워크(In-Vehicle Network)로 최대 2개의 채널로 각 채널당 최대 10Mbps 전송 속도를 가진다. FlexRay는 TDMA(Time Division Multiple Access)와 FTDMA(Flexible Time division Multiple Access) 방식을 사용하여 효율적으로 데이터를 전송하면서도 실시간 통신을 보장하기 때문에 파워트레인 등 안전성이 필수적인 차량 모듈의 실시간 전자 제어에 사용된다. 본 논문에서는 ISO 17458 표준과 FlexRay 컨소시엄 문서를 바탕으로 FlexRay의 프로토콜(Protocol), 시간 계층(Time Hierarchy), 메시지 프레임(Message Frame), 커뮤니케이션 컨트롤러(Communication Controller), 적합성 시험(Conformance Test)에 대해 자세히 설명한다.

Key words : FlexRay, In-Vehicle Network, Protocol, Time Hierarchy, Message Frame, Communication Controller, Conformance Test

Ⅰ. 서론

자동차 기술이 빠른 속도로 발전함에 따라 차량에 새로이 적용되는 전장 기술(Automotive electronics)이 계속해서 늘어나고 있다. 또한 차량 내의 기계적 제어를 전자적 제어(X-by-Wire)로 대체하려는 자동차 산업계의

* School of Electronic Engineering and Department of Intelligent Semiconductor, Soongsil University (Student, Student, Student, Student, Student, Professor)
★ Corresponding author
E-mail : sslee@ssu.ac.kr, Tel : +82-2-820-0692
※ Acknowledgment
This work was supported by the R&D Program of the Ministry of Trade, Industry, and Energy (MOTIE) and Korea Evaluation Institute of Industrial Technology (KEIT). (20023805, RS-2022-00154973, RS-2023-00232192)
Manuscript received Dec. 18, 2023; revised Dec. 20, 2023; accepted Dec. 22, 2023.

움직임에 따라 차량에 탑재되는 ECU(Electronic Control Unit)의 숫자와 통신 대역폭 역시 크게 증가하고 있다.

그러나 기존에 차량용 통신 프로토콜로 가장 많이 사용되고 있던 LIN(Local Interconnect Network)[1], CAN(Controller Area Network)[2]의 통신 속도는 각각 19.2kbps, 1Mbps에 불과하고 실시간 전송을 보장할 수 없기 때문에 파워트레인 등 안전성이 중요한 모듈의 전자 제어에는 적합하지 않았다. 때문에 자동차 산업계는 높은 통신 속도와 실시간 전송을 보장하는 새로운 통신 프로토콜을 개발할 필요성을 느끼게 되었다.

FlexRay[2]는 BMW, Bosch, Philips 등 다수의 자동차 산업체가 설립한 FlexRay 컨소시엄에서 개발한 새로운 통신 프로토콜이다. FlexRay는 두 개의 통신 채널을 통해 각 채널당 최대 10Mbps의 속도로 메시지를 전송 및 수신할 수 있으며, 이는 기존의 CAN 프로토콜에 비해 10배나 빠른 속도이다. 또한 메시지 전송에 TDMA (Time Division Multiple Access)와 FTDMA(Flexible Time Division Multiple Access) 방식을 사용하여 실시간 전송을 보장하면서도 효율적인 전송이 가능하다. FlexRay는 차량 통신 백본(Backbone), 파워트레인 전자 제어, 샤시 전자 제어를 비롯한 다양한 분야에 널리 사용된다. 표 1은 FlexRay와 다른 차량용 통신 프로토콜을 비교한 것이다.

Table 1. Comparison of in-vehicle network protocols.
표 1. 차량 통신 프로토콜의 비교

Protocol	LIN	CAN	FlexRay
Target	Peripherals	Soft Real-Time	Hard Real-Time
Application	Body	Chassis	Backbone Powertrain
Architecture	Single Master	Multi-Master	Multi-Master
Arbitration	Polling	CSMA/NBA	TDMA/ FTDMA
Data Rate	19.2 kbps	1 Mbps	10 Mbps
Data Bytes	0-8 Bytes	0-8 Bytes	0-254 Bytes
Wire	Single Wire	Dual Wire	Dual Wire
Channel	1 Channel	1 Channel	2 Channel

본 논문에서는 ISO(International Standard Organization)에서 제정한 ISO 17458 국제표준[3]-[7]과 FlexRay 컨소시엄에서 발행한 FlexRay 규격 문서인 Protocol Specification (PS10[8]), Electrical Physical Layer Specification (EPL10[9]), Protocol Conformance Test Specification (PCT10[10]), Bus Guardian Specification[11]의 내용을 바탕으로 FlexRay 프로토콜의 세부 내용을 설명한다. 2장에서는 FlexRay 프로토콜의 기본 사항, 3장에서는 시간 계층, 4장에서는 메시지 프레임, 5장에서는 커뮤니케이션 컨트롤러, 6장에서는 적합성 시험을 다룬다.

II. FlexRay 프로토콜

1. TDMA/FTDMA

TDMA는 클러스터를 구성하는 통신 노드들이 채널을 시간적으로 분할하여 사용하는 방식을 말한다. 통신 채널을 일정한 간격으로 나눈 시간 구간을 슬롯이라 하며, 슬롯은 클러스터를 구성할 때 클러스터의 각 노드에 할당된다. 노드는 자신에게 할당된 시간 슬롯에서만 메시지를 전송할 수 있다. FTDMA는 TDMA의 슬롯 할당 방식을 조금 더 유연하게 변형한 방식이다. FTDMA 방식에서는 슬롯의 길이를 메시지 길이에 맞춰 동적으로 확장할 수 있으며, 이를 통해 대역폭을 조금 더 효율적으로 사용할 수 있도록 한다.

FlexRay 프로토콜은 동적, 정적 세그먼트에서 각각 TDMA, FTDMA 방식을 사용하여 메시지를 전송한다. 엄격한 시간 슬롯 할당이 이루어지는 정적 세그먼트에서의 메시지 전송을 통해 프로토콜에서 메세지의 결정론적 전송을 보장하며, 보다 유연하게 시간 슬롯을 할당하는 동적 세그먼트에서의 메시지 전송을 통해 네트워크의 성능을 최적화한다.

FlexRay 프로토콜은 시분할 방식을 통해 버스상의 충돌을 효과적으로 방지하고 예측 가능한 실시간 네트워크 통신을 구현할 수 있으나, 시분할 방식을 사용하기 위해 반드시 사전에 네트워크 구조를 정의하고, 개별 노드에 슬롯을 할당해야 한다. 따라서 FlexRay 프로토콜은 트래픽 패턴의 동적인 변화에 대응하기 어려우며, 기존 클러스터에 새로운 노드를 추가하기 어렵다는 단점 또한 존재한다.

2. 동기화 방식

시분할 방식의 통신 네트워크를 구현하기 위해서는 클러스터를 구성하는 노드들 사이에 정밀한 수준의 시간 동기화가 이루어져야 한다. FlexRay 클러스터의 모든 개별 노드는 특정한 유형의 메시지 프레임을 수신하여 해당 프레임의 시간 정보로부터 자신과 네트워크 사이의 시간 오차를 계산하거나, 다른 외부 클러스터와 연결된

Time Gateway로부터 시간 정보를 수신하여 동기화 프로세스를 수행할 수 있다.

이때 단일 동기화 노드로부터 파생된 시간 정보를 통해 동기화를 수행하도록 구성한 클러스터를 TT-L(Time-Triggered Local Master) 클러스터라 하며, 2~15개의 동기화 노드를 사용해 동기화를 수행하는 클러스터를 TT-D(Time-Triggered Local Distributed)라 한다. Time Gateway를 통해 다른 외부 클러스터에 동기화하도록 구성한 클러스터를 TT-E(Time-Triggered External) 클러스터라고 한다.

3. Architecture of Flexray Node

FlexRay 통신 클러스터를 구성하는 노드는 그림 1과 같이 Host, Communication Controller (CC), Bus Driver(BD), Bus Guardian(BG)으로 구성된다. FlexRay의 Host, CC, BD는 각각 CAN의 프로세서, 콘트롤러, 트랜시버에 대응하며 BG는 FlexRay에만 존재한다.

CC는 Host로부터 받는 제어 신호와 메시지 데이터를 올바른 방식으로 처리하여 BD가 실제 전기 신호로 변환해 출력할 수 있도록 하며, 동시에 BD를 통해 수신한 프레임과 자신의 상태 정보를 Host가 읽어갈 수 있게 한다. CC는 BD라는 물리적 버스 인터페이스를 통해 실제 전송 매체에 연결된다. BD는 전송할 논리 신호를 물리적 전압으로 변환하는 역할을 하며, 수신된 물리적 전압을 CC가 사용할 수 있는 논리 신호로 변환한다.

BD는 고주파 간섭과 정전기 등으로 인한 영향을 최소화하기 위해 그림 2와 같이 BP(Bus Plus)와 BM(Bus Minus)의 두 개의 선으로 이루어진 차동 신호 전송 방식을 사용한다. BD가 저전력 모드로 동작할 경우 BP와 BM의 전압은 0다. 정상 동작 중 IDLE 상태에서 BP와 BM의 전압은 2.5V이며, 두 라인의 전위차는 없다. High 신호를 전송할 때 BP는 3.5V, BM는 1.5V가 되며 Low 신호를 전송할 때는 BP가 1.5V, BM이 3.5V가 된다. CAN과 달리 FlexRay는 High와 Low 신호 사이에 우성/열성 관계가 없으며, 대신에 각 채널에서는 한 번에 하나의 노드만 메시지를 전송할 수 있도록 하여 데이터 파괴를 방지한다.

BG는 Host와 BD의 데이터 송/수신 스케줄을 확인하고, 허용되지 않은 시간 슬롯에서 노드가 데이터 전송을 시도하거나 데이터를 수신하는 경우 이를 차단하여 네트워크 오류가 발생하는 것을 방지하는 역할을 한다.

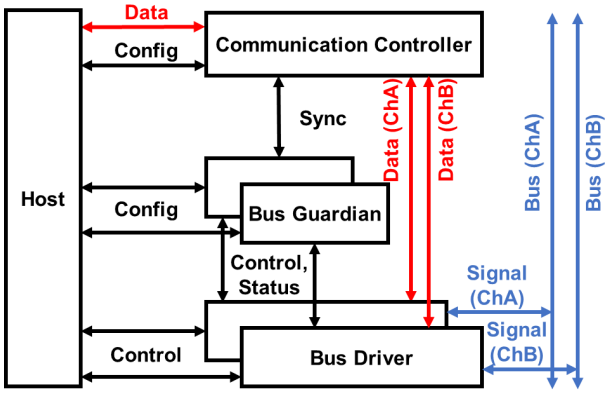

Fig. 1. FlexRay node architecture.
그림 1. FlexRay 노드 구조

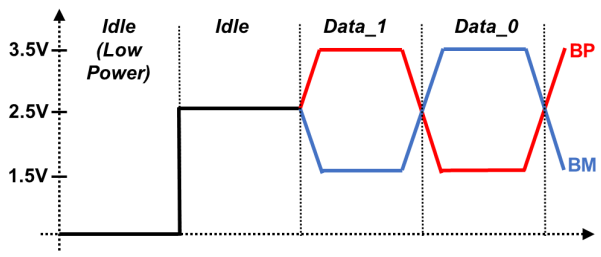

Fig. 2. FlexRay physical layer voltage level.
그림 2. FlexRay 물리 계층 전압 레벨

Ⅲ. FlexRay 시간 계층

FlexRay 프로토콜은 효율적인 통신과 정밀한 시간 동기화를 위한 계층 구조(Time Hierarchy)를 가지고 있으며 Cycle, Segment, Slot, Macrotick, Microtick 레벨로 세분화할 수 있다.

Cycle은 FlexRay 프로토콜에서의 가장 큰 시간 단위이다. FlexRay는 사이클 기반의 통신을 하며, 특정 FlexRay 통신 클러스터는 7~63 사이에서 홀수 개의 사이클을 갖도록 구성될 수 있다.

Cycle은 다시 Static, Dynamic, Symbol Window(SW), Network Idle Time(NIT)의 4가지 Segment로 세분화할 수 있다. 모든 Cycle은 Static, NIT Segment를 반드시 포함해야 하며, Dynamic, SW Segment는 Configuration을 통해 사용 여부를 자유롭게 결정할 수 있다. 한 Cycle은 그림 3과 같이 4가지 형태를 가질 수 있다.

Static Segment는 클러스터의 노드들이 TDMA 방식에 따라 버스를 점유하는 구간이다. 정적 세그먼트는 최대 1023개의 Static Slot으로 구성될 수 있으며, 각 Slot들은 클러스터를 구성할 때 각 개별 노드에 할당된다. 개별 노드는 오직 자신에 할당된 Static Slot에서만 Frame을 보낼 수 있다.

Static Segment를 구성하는 Static Slot의 길이는 일정하며, Slot에 담겨 전송되는 모든 Static Frame의 길이 역시 동일하다. 따라서 각 노드는 Static Segment에서 Frame이 언제 도착할지 예상할 수 있으며, 이를 통해 FlexRay 클러스터를 구성하는 각 노드들은 수신되는 특정한 메시지 프레임들의 시간 정보를 이용해 동기화 작업을 수행할 수 있다.

Dynamic Segment는 그림 4와 같이 클러스터의 노드들이 FTDMA 방식에 따라 버스를 점유하는 구간이다. Dynamic Segment는 최대 7988개의 Minislot으로 구성될 수 있으며, 특정 Minislot에서 Dynamic Frame이 전송될 경우, 해당 Frame을 전송할 수 있도록 해당 Minislot이 Dynamic Slot으로 확장된다. Dynamic Slot의 길이는 전송하려는 Frame의 길이에 따라 달라진다.

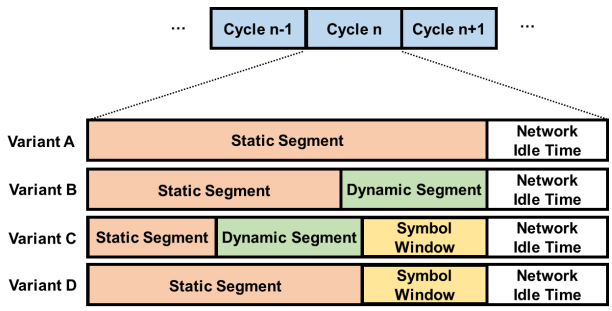

Fig. 3. 4 cycle configurations in FlexRay.
그림 3. FlexRay의 4가지 사이클 형태

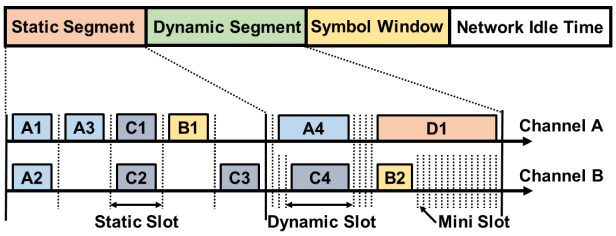

Fig. 4. Slot assignment in static and dynamic segment.
그림 4. 정적 및 동적 세그먼트에서의 슬롯 할당

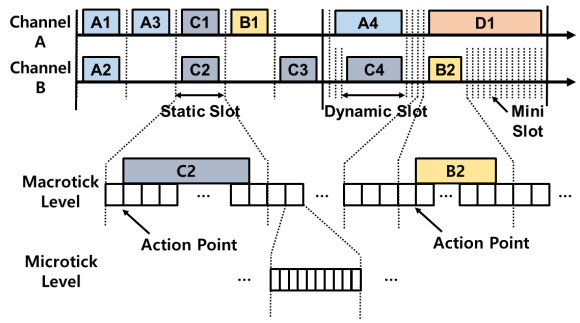

Fig. 5. Time hierarchy in FlexRay.
그림 5. FlexRay의 시간 계층

Static, Dynamic Segment의 슬롯 경계에서 각 노드는 메시지 버퍼에 현재 노드의 슬롯 카운터 변수 값인 vSlotCounter 값과 일치하는 FrameID를 가진 데이터가 있는지 확인하고 해당 데이터가 존재한다면 슬롯에서 메시지를 전송하게 된다.

SW는 프로토콜에서 특수한 목적으로 사용되는 Symbol을 전송할 수 있는 구간이다. SW에서는 CAS_MTS (Collision Avoidance Symbol_Media Access Test Symbol), WUDOP(WakeUp during Operation Pattern)과 같은 특수한 패턴 전송이 트리거될 수 있다.

NIT는 Cycle 상에서 어떠한 통신도 일어나지 않는 구간이다. NIT의 길이는 가변적이며, 클러스터의 각 노드는 NIT 구간에서 계산된 클록 오차 보정치를 적용하여 클러스터와의 동기화를 오차 범위 내(~0.3%)로 유지한다. 한 사이클에 포함되는 Macrotick의 개수는 항상 일정하므로, NIT의 길이는 Static, Dynamic, SW Segment에 할당되지 않은 길이의 Macrotick이 된다.

Macrotick은 그림 5와 같이 통신 클러스터에 속한 모든 노드가 공유하는 가장 작은 시간 단위이다. FlexRay 클러스터의 노드들은 동일한 시점에 Macrotick이 발생하도록 동기화되며, Macrotick의 길이는 1~6us 사이에서 구성할 수 있다.

CC는 통신의 안정성을 위해 Segment 또는 Slot Boundary에서 일정 Macrotick이 경과한 시점에 CE (Communication Element) 전송을 트리거하는데, 이 Macrotick 경계 지점을 Action Point라 한다.

Microtick은 CC의 오실레이터 클록으로부터 직접 파생되는 시간 단위로, 클러스터의 개별 노드가 각자 가지고 있는 가장 작은 시간 단위이다. FlexRay는 동기화를 위해 Macrotick을 보다 더 작은 단위로 세분화한 Microtick을 사용함으로서 보다 정밀한 조정과 동기화를 가능하게 한다.

Ⅳ. FlexRay 메시지 프레임

FlexRay 프로토콜의 데이터는 프레임 형태로 인코딩되어 송/수신된다. 프레임은 그림 6과 같이 Header Segment, Payload Segment, Trailer Segment 세 부분으로 나눌 수 있다.

1. Header Segment(5바이트)

Header Segment는 프레임의 핵심 정보들을 담고 있는 부분으로 그 길이는 항상 5바이트로 고정되어 있다.

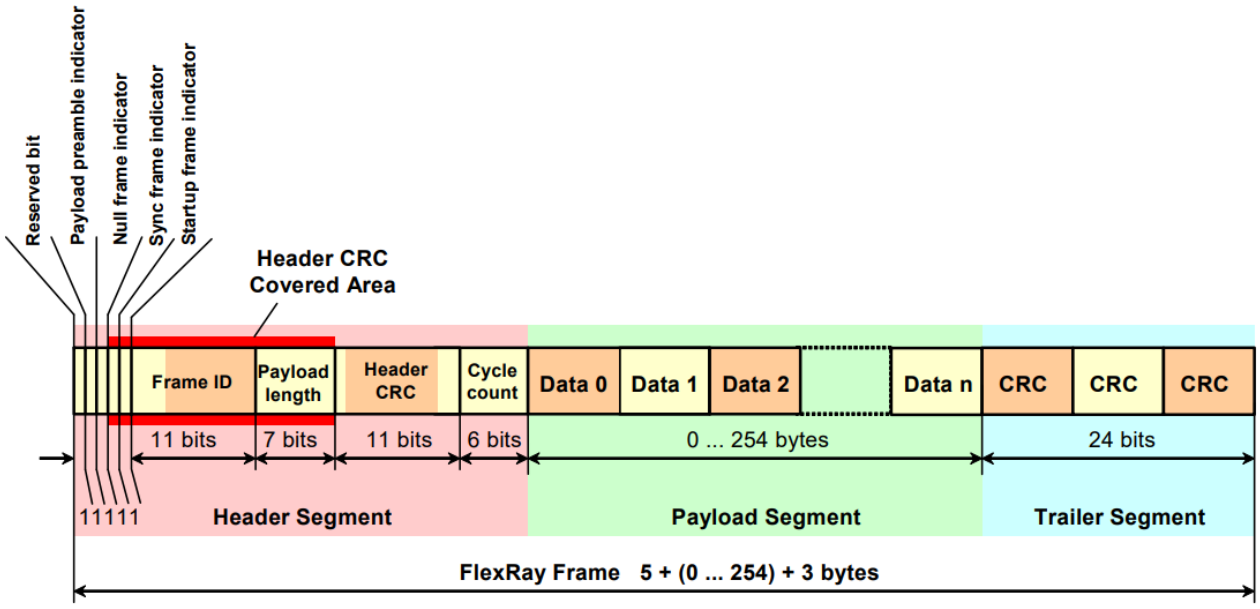

Fig. 6. Flex message frame.
그림 6. FlexRay 메시지 프레임

Header는 Reserved Bit(1비트), Payload Preamble Indicator(1비트), Null Frame Indicator(1비트), Sync Frame Indicator(1bit), Startup Frame Indicator(1비트), Frame ID(11비트), Payload Length(7비트), Header CRC(11비트), Cycle Count(6비트)로 나뉜다.

Reserved bit는 향후 프로토콜의 확장을 위해 예약되어 있는 비트로 항상 0이다. Payload Preamble Indicator (PPI)는 해당 프레임의 Payload Segment 안에 Network Management Vector(정적 세그먼트에서 전송되는 프레임의 경우) 또는 Message ID(동적 세그먼트에서 전송되는 프레임의 경우) 가 포함되어 있는지 여부를 나타낸다. PPI가 1로 설정된 프레임의 경우, Host는 Payload Segment의 데이터 일부를 응용 프로그램 데이터로써 사용할 수 있다. Null Frame Indicator (NFI)는 해당 프레임이 Null 프레임인지 아닌지를 나타내며, Frame이 Null 프레임일 경우 0, 아닐 경우 1의 값을 가진다. Sync Frame Indicator (SyFI) 는 해당 프레임이 Sync Frame인지 여부를 나타낸다. 개별 노드가 유효한 Sync Frame을 수신할 경우, 해당 프레임의 시간 정보를 CSP (Clock Synchronization Process) 모듈에서 노드의 클록 오차를 측정하고 보정값을 계산하는 데 사용할 수 있다. Startup Frame Indicator(SuFI)는 해당 프레임이 Startup Frame인지 여부를 나타낸다. 개별 노드가 유효한 Startup Frame을 수신할 경우, 해당 프레임을 CSS (Clock Startup Synchronization Process) 모듈에서 통신 시작 전 클러스터의 모든 노드의 시간을 동기화하는 STARTUP 프로세스에 사용할 수 있다.

Frame ID는 해당 프레임이 전송되어야 하는 Slot을 정의한다. TDMA/FTDMA 통신을 수행하는 Flexray 프로토콜에서 모든 노드는 현재 클러스터의 SlotCounter 값과 일치하는 FrameID를 가진 프레임만 전송할 수 있다. Payload Length는 프레임의 Payload Segment의 길이를 나타낸다. Payload Length는 7비트로 0부터 127까지의 값을 가질 수 있으며, Payload Segment의 길이는 Payload Length의 2배이다. Header CRC는 Header의 SyFI, SuFI, Frame ID, Payload Length, 총 20비트에 대해 CRC(Cyclic Redundancy Code) 연산을 수행한 결과이다. Header CRC는 CC에서 계산하지 않고 Host에서 계산되어 전달된 값을 사용하며 CRC 다항식과 초기화 벡터 값은 식 (1)과 같다. Cycle Count는 노드에서 프레임이 전송될 때, MAC(Media Access Control) 모듈이 가지고 있는 CycleCounter의 값을 나타낸다. MAC 모듈은 프레임을 Assemble할 때 Header의 Cycle Count를 현재 노드의 CycleCounter 값으로 설정한다.

$$G(x) = x^{11} + x^9 + x^8 + x^7 + x^2 + 1 \qquad (1)$$

(초기화 벡터: 0×01A)

2. Payload Segment(0~254바이트)

Payload Segment는 프레임에서 전송하려는 실제 데이터를 담고 있는 부분이다. Payload Segment의 길이는 Header의 Payload Length의 2배로, 항상 짝수 단위 바이트를 가지며, 0(Null Frame)부터 최대 254바이트까지의 데이터를 포함할 수 있다.

3. Trailer(3 Byte)

Trailer Segment는 Header와 Payload Segment의 모든 비트(5~257바이트)에 대해 CRC 연산을 수행한 결과를 담고 있는 부분이다. Frame CRC의 계산은 CC에서 이루어진다. CODEC(Coding-Decoding) 모듈은 MAC으로부터 전송하려는 Frame vTF Frame을 수신받아 CRC를 계산한다. CRC 다항식과 초기화 벡터 값은 식 (2)와 같다.

$$G(x) = x^{24}+x^{22}+x^{20}+x^{19}+x^{18}+x^{16}+x^{14}+ x^{13}+x^{11}+x^{10}+x^{8}+x^{7}+x^{6}+x^{3}+x+1 \quad (2)$$

(채널 A의 초기화 벡터 : 0×FEDCBA)
(채널 B의 초기화 벡터 : 0×ABCDEF)

V. FlexRay 커뮤니케이션 컨트롤러

Flexray CC(Communication Controller)는 그림 7과 같이 8종의 하위 모듈로 구성되며 Controller Host Interface(CHI)를 제외한 나머지 7종의 모듈, 즉 Protocol Operation Control(POC), Media Access Control (MAC_A, MAC_B), Coding and Decoding(CODEC_A, CODEC_B), Frame and Symbol Processing (FSP_A, FSP_B), Clock Synchronization Startup Process (CSS_A, CSS_B), Clock Synchronization Process(CSP)를 PE(Protocol Engine)로 구분한다.

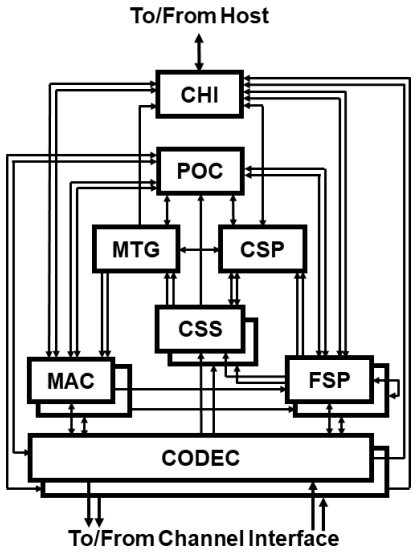

Fig. 7. FlexRay communication controller.
그림 7. FlexRay 커뮤니케이션 컨트롤러

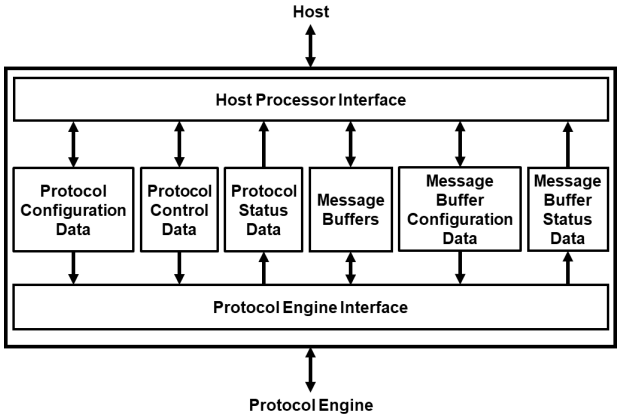

Fig. 8. FlexRay communication host interface.
그림 8. FlexRay 커뮤니케이션 호스트 인터페이스

1. Controller Host Interface(CHI)

CHI는 Host와 PE 사이의 데이터 교환을 담당하는 모듈이다. Host는 PE에게 직접 명령을 내리거나 제어할 수 없으며, 대신 Host는 CHI로 하여금 자신의 명령을 PE에 적절한 형식으로 전달하도록 한다. 마찬가지로 PE 역시 자신의 Status 정보를 직접 Host에 전달할 수 없으며, 대신 PE의 각 모듈은 자신의 Status Data를 CHI에 전송하고, CHI는 이를 다시 적절한 형식으로 변환하여 Host에 전달하게 된다.

CHI는 역할에 따라 Protocol Data Interface와 Message Data Interface로 나뉜다. Protocol Data Interface는 Host와 PE가 프로토콜에서 규정하는 방식으로 Protocol Data를 주고받을 수 있도록 한다. Host는 CHI 내부의 Protocol Configuration Data Register에 Write하여 PE가 원하는 대로 동작하도록 Configuration 하며, Host가 CHI 내부의 Protocol Control Register에 Write하면 CHI가 그에 따른 적절한 CHI Command를 POC(Protocol Operation Control)에 전송하는 방식으로 PE의 동작을 제어한다. 또한 PE의 각 모듈은 자신의 Status 정보를 CHI에 알리고, CHI는 이를 수신하여 Protocol Status Data Register에 저장한 뒤 프로토콜에 규정된 형식대로 Host에 알리게 된다.

CHI의 Message Data Interface는 Host와 PE가 프로토콜에서 규정하는 방식으로 Message Data를 주고받을 수 있도록 한다. CC가 언제 메세지를 송/수신해야 하

는지에 대한 정보를 Communication Slot Assignment 라고 하는데, Host는 각 노드 CHI 내부의 Message Buffer Configuration Register에 Write하여 채널별로 Slot Assignment를 수행한다. 또한 Host는 Message buffer Register에 송신할/수신한 Frame Data를 저장하여 Host와 PE가 주고받을 수 있도록 하며, 송신용/수신용으로 구성된 각 메시지 버퍼에 대한 Slot Status 정보를 저장한다.

2., Protocol Operation Control(POC)

POC는 PE의 전체적인 동작을 제어하는 모듈이다. Flexray 프로토콜의 동작은 크게 4가지 핵심 프로세스로 나누어 볼 수 있는데, Flexray 클러스터에서 통신이 진행될 때 개별 노드의 PE는 Coding and Decoding, Media Access Control, Frame and Symbol Processing, Clock Synchronization의 4가지 프로세스를 반복적으로 수행한다. 이때 POC는 CHI로부터 전송받는 CHI Command와 자신의 현재 상태 정보, 타 모듈들로부터 수신받는 Signal에 따라, PE의 하위 모듈인 MAC, FSP, CODEC, CSS, CSP, MTG로 적절한 SDL signal과 Data를 보내어 PE에서 앞선 4가지 매커니즘이 정상적으로 수행되도록 한다.

POC는 통신이 시작될 때 WUP(Symbol)을 전송하여 채널에 연결된 모든 노드의 BD가 저전력 모드에서 활성 모드로 전환하도록 한다. 이를 WAKEUP 프로세스라 한다. CC는 한 번에 하나의 채널에서만 WAKEUP을 수행할 수 있다.

3. Media Access Control(MAC_A, MAC_B)

MAC은 CC가 5종의 CE(Static Frame, Dynamic Frame, CAS_MTS, WUP, WUDOP)를 올바른 타이밍에 전송하도록 제어하는 역할을 한다. MAC은 MTG로부터 수신되는 Macrotick을 기반으로 동작하는데, MAC은 Macrotick 신호와 그 외 다른 모듈들로부터 수신하는 여러 신호들을 통해 통신 클러스터에서 통신이 진행 중인지 여부, 클러스터의 Global Time(Cycle Counter, Segment, Slot Counter, Macrotick Counter 값)을 알 수 있다. MAC은 이 정보들을 바탕으로 CODEC에게 CE 전송을 트리거하는 신호를 특정 시점에 전송함으로써 CC가 CE 전송을 시작하게 한다.

각 노드는 Static/Dynamic Segment에서 현재 MAC이 가지고 있는 SlotCounter 값과 일치하는 Frame ID를 가진 프레임만 전송할 수 있다. MAC은 Static/Dynamic Segment의 각 Slot Boundary에서, CHI의 Memory buffer에 현재 SlotCounter와 일치하는 Frame ID를 가진 전송 데이터가 존재하는지 확인한다. 버퍼에 조건을 만족하는 Data가 있다면 이를 vTCHI 구조체로 수신받아 vTF로 Assemble한 뒤, 이를 Slot의 Action Point에 맞춰 CODEC에 전송함으로써 프레임 전송을 Trigger한다.

SW에서는 CAS_MTS, WUDOP(Symbol)만 전송할 수 있다. MAC은 매 Cycle의 SW Segment에서 CHI로부터 직접 CAS_MTS, WUDOP 전송 여부를 결정하는 신호를 수신하고, SW의 Action Point 시점에 맞춰 CODEC에 Symbol 전송을 Trigger한다. 이에 반하여 NIT에서는 모든 CE 전송이 금지된다.

4. Coding and Decoding(CODEC_A, CODEC_B)

CODEC은 물리 계층인 BD에 직접 연결되어 TxD, TxEN 신호를 보내고 RxD 신호를 수신하는 모듈이다. CODEC은 송수신 데이터에 대한 인코딩/디코딩을 수행하는 CODEC 프로세스 외에도, Bit Strobing과 WUP Decoding 프로세스를 수행하는 2개의 하위 모듈인 BITSTRB 모듈과 WUPDEC 모듈을 가지고 있다.

CODEC은 MAC이 CE 전송을 Trigger하면 MAC으로부터 전송받은 Frame Data 또는 Symbol을 직렬 비트스트림으로 변환해 BD로 전송한다. 이 때 전송의 안정성을 위해 CODEC은 Frame Data에 TSS, FSS, BSS, FES 등의 통신 요소를 추가로 삽입해 인코딩하여 전송한다.

Flexray 프로토콜에서 통신이 이루어지고 있지 않을 때 버스는 High 상태이다. 자신이 CE를 전송하고 있지 않고 채널이 유휴 상태일 때 RxD에서 Low가 감지되면 CODEC은 통신이 시작되었음을 인지하고 비트스트림 수신과 디코딩을 시작한다.

R×D를 통해 수신된 비트스트림에는 여러 가지 요인으로 인한 지연와 노이즈가 포함되어 있다. 비트스트림으로부터 CE를 올바르게 디코딩하기 위해서는 비트스트림을 적절히 샘플링해 디지털 데이터로 복구해야 하는데 이를 Bit Strobing이라 한다. CODEC의 하위 모듈 BITSTRB는 Bit Strobing 프로세스를 담당한다. CODEC의 모든 Decoding 관련 프로세스는 수신한 데이터 입력 RxD 값을 그대로 사용하는 것이 아니라, BITSTRB 프로세스에서 샘플링 후 Voting window를 거쳐 노이즈를 제거한 Data 값을 사용한다.

CODEC의 하위 모듈 WUPDEC은 현재 버스에서 WAKEUP 프로세스에 관련된 CE가 전송되고 있는지 감

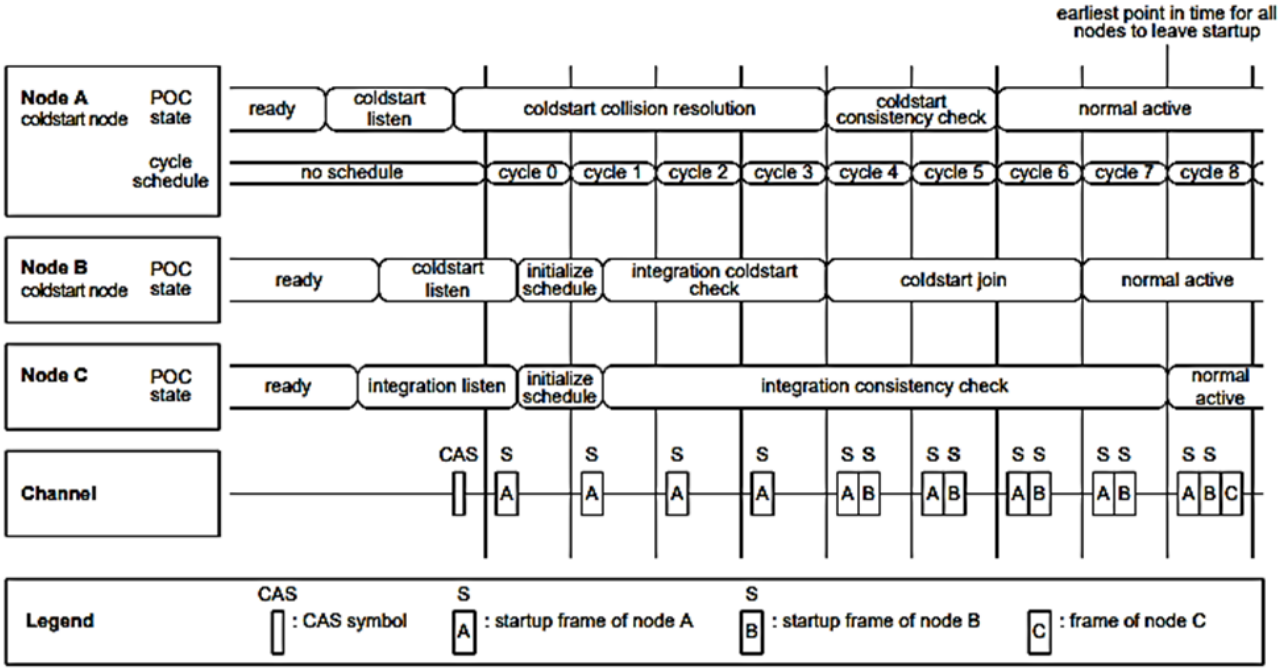

Fig. 9. Example: STARTUP process in TT-D Cluster.
그림 9. 예시: TT-D 클러스터에서의 스타트업 프로세스

지하고 이를 디코딩하는데, 이를 WUP Decoding이라 한다. WUPDEC은 통신 시작 전 클러스터의 WAKEUP 프로세스 중에 전송되는 WUP Symbol 또는 통신 진행 중 특정 주기의 SW에서 전송되는 WUDOP Symbol을 감지하여 이를 디코딩하고, Symbol이 디코딩되었음을 다른 모듈들에 알린다.

CODEC은 프레임을 디코딩한 경우 수신한 프레임에 대한 정보를 vRF 구조체에 담아 CSS와 FSP로 전송하며, WUP를 수신한 경우 POC에, WUDOP를 수신한 경우 FSP에, CAS_MTS를 수신한 경우 POC와 FSP에 해당 Symbol을 수신했다는 신호를 전송한다.

5. Frame and Symbol Processing(FSP_A, FSP_B)

FSP는 CODEC이 수신한 CE가 시간적으로, 문법적으로 올바른지 검사하는 모듈이다. FSP는 다른 모듈에서 특정한 이벤트가 발생할 경우에 해당 이벤트가 발생했다는 신호를 수신하며, 또한 MAC으로부터 매 Segment, Slot Boundary마다 해당 Signal과 Cycle/Slot Counter 값을 수신하여 현재 노드의 시간 정보를 획득한다. 이를 바탕으로 FSP는 채널에서 송/수신되고 있는 모든 CE에 대해 프로토콜적으로 올바른 시간에 수신되었는지 검사한다.

또 만약 CODEC이 Frame을 수신했을 경우, 수신한 vRF 구조체에 대해 추가적인 Syntax 검사를 수행하며 검사 결과를 vSS 구조체에 담아 CHI로 전송함으로써 Host에 수신한 프레임에 대한 정보를 제공하게 된다. 이 때 Timing Error, Syntax Error가 없는 CE를 유효한 CE라고 한다.

6. Clock Synchronization Startup Process (CSS_A, CSS_B)

CSS는 STARTUP 프로세스를 담당하는 모듈이다. Flexray 는 TDMA 통신을 지원하기 위해 처음 통신을 시작할 때 클러스터의 모든 노드가 자신의 통신 일정을 초기화하고 자신의 Local 시간을 클러스터의 Global 시간에 동기화하는 작업을 수행한다. 이를 STARTUP 프로세스라 한다.

STARTUP 프로세스 실행시 수행되는 작업은 클러스터가 채택한 동기화 방식(TT-D, TT-L, TT-E)과, 해당 노드가 클러스터에서 콜드스타트 노드인지 일반 노드인지에 따라 다르다. 본 논문에서는 그림 9와 같이 TT-D 동기화 방식을 사용하는 클러스터의 STARTUP 프로세스를 예시로 들어 설명한다.

TT-D 클러스터에서 노드가 콜드스타트 노드일 경우, 노드는 채널에서 통신이 진행 중인지 정해진 시간 동안 살피고, 채널에서 어떤 통신도 감지되지 않으면 CAS Symbol을 전송한다. 모든 개별 콜드스타트 노드가 CAS

Symbol을 전송하지만, 첫 4주기에 걸쳐 단 하나의 노드만 CAS Symbol을 전송하게 된다. 이 때 처음 CAS Symbol을 전송하게 되는 노드를 선행 콜드스타트 노드라고 한다.

TT-D 클러스터에서 콜드스타트 노드들은 사전에 자신만의 고유한 KeySlotID를 1개 갖도록 Configuration된다. 선행 콜드스타트 노드는 첫 4 주기에서 Frame Header의 SuFI가 1로 표기된 Startup Frame을 자신의 KeySlotID와 일치하는 Static Slot에서 전송한다. 이 Frame들의 시간 정보를 기준으로 후행 콜드스타트 노드들이 그 다음 4주기에서 Startup Frame을 전송하면서 클러스터에 동기화되게 되며, 모든 콜드스타트 노드들이 동기화되면 마지막으로 다른 일반 노드들이 동기화되는 방식으로 STARTUP 프로세스가 수행된다.

8사이클에 걸쳐 STARTUP 프로세스가 정상적으로 종료되면 클러스터의 모든 노드가 POC는 POC: Normal Active State로 진입하게 되고 CE 송수신을 시작할 수 있게 된다.

7, Clock Synchronization Process(CSP)

CSP는 개별 노드의 Local Time과 클러스터의 Global Time 사이의 차이를 측정하고, 클록 동기화를 위한 오차 보정치를 계산하는 모듈이다.

클러스터를 구성하는 노드들 중 Sync 노드는 클러스터 전체의 동기화 프로세스를 수행하기 위해, Header의 SyFI가 1로 표기된 Sync Frame을 전송한다. 이러한 Sync Frame은 Static Segment의 KeySlot 에서만 전송될 수 있기 때문에, 클러스터의 모든 노드는 각 Sync Frame들이 수신되었어야 하는 시간과 실제로 수신된 시간 사이의 차이를 측정할 수 있다.

CSP는 정상 동작 중 수신된 모든 Sync Frame의 시간 정보를 저장할 수 있는 테이블을 내장하고 있다. 각 클러스터는 모든 Sync Frame에 대해 매 주기마다 얼만큼의 시간 오차가 발생했는지를 측정해 테이블에 저장한다.

CSP는 이 테이블의 시간 정보를 기반으로 매 사이클마다 자신이 수정해야 하는 Offset 오차 보정치를, 매 홀수 Cycle마다 Rate 오차 보정치를 계산할 수 있으며, 계산한 오차 보정치를 MTG에게 전송한다. 각 오차 보정치는 음수일 수 있다.

8. MacroTick Generation(MTG)

MTG는 매크로틱을 생성하는 모듈이다. MTG는 POC, CSS, CSP로부터 신호를 받아 Macrotick 생성을 시작하며, CSP에서 계산되어 수신한 Offset / Rate 오차 보정치를 수신받아 매크로틱 생성 주기에 반영한다.

Offset 오차 보정치는 매 홀수 Cycle의 NIT Segment에서 반영되며, Rate 오차 보정치는 매 짝수 Cycle 전체에 걸쳐 반영된다. MTG에서 보정치를 적용해 생성한 매크로틱은 MAC으로 전송되어, MAC이 매크로틱 단위에서 올바른 Timing에 CE 전송을 트리거할 수 있도록 한다.

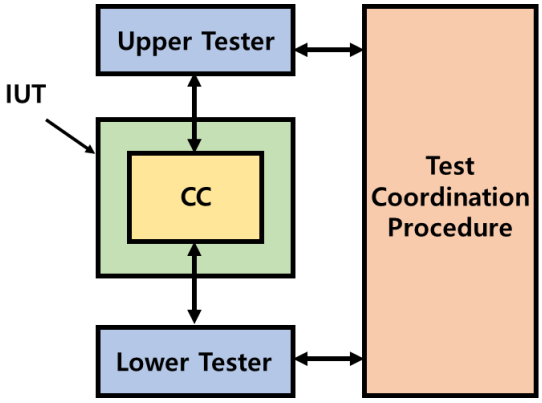

Fig. 10. Test architecture in ISO/IEC 9646-1.
그림 10. ISO/IEC 9646-1의 테스트 아키텍쳐

Ⅵ. FlexRay 적합성 시험

FlexRay 프로토콜의 검증은 FlexRay 컨소시엄에서 발표한 공식 적합성 시험(Conformance Test) 문서인 PCT10[10]의 내용을 바탕으로 한다. PCT10은 설계한 CC가 공식 표준에 따라 올바르게 동작하는지 검증하기 위한 문서이다.

프로토콜 검증은 그림 10과 같이 ISO/IEC 9646-1[12] 표준을 따르는 Test Architecture 하에서 수행된다. 그림 10에서 Upper Tester(UT)는 FlexRay CHI에 연결되며, Lower Tester(LT)는 BD에 연결된다. CC 검증은 시뮬레이션 및 FPGA 환경에서 각각 수행할 수 있으며, 다수의 Test Vector를 통해 구현한 특정 상황에서 CC가 올바르게 동작하는지 각 Case 별로 검증한다.

PCT10은 각 Test Case를 통해 4가지 Segment에서 CE 송수신이 올바르게 이루어지는지, Host가 CHI를 올바르게 Configuration할 수 있는지, POC의 각 State에서 수신된 특정 CHI Command에 대해 올바르게 동작하는지, 메시지 버퍼를 원하는 대로 구성할 수 있는지, 클록 동기화 프로세스가 올바르게 수행되는지, WAKEUP/STARTUP 프로세스가 올바르게 수행되는지 등을 검증할 수 있도록 구성되어 있다.

Ⅶ. 결론

본 논문에서는 고속 실시간 차량 통신 프로토콜인 FlexRay에 대해 프로토콜 개요, 시간 계층, 메시지 프레임, 커뮤니케이션 컨트롤러, 적합성 시험 등을 자세하게 설명하였다. FlexRay 프로토콜은 고속 실시간성으로 인해 차량 통신에서 매우 중요하지만 관련 자료와 설명이 부족하여 연구자들이 어려움을 겪었는데 본 논문이 FlexRay를 분석하고 설계하는데 도움이 될 것으로 생각된다.

References

[1] ISO 11898-1:2015, “Road Vehicles-Controller Area Network (CAN)-Part 1: Data Link Layer and Physical Signaling,” https://www.iso.org/standard/63648.html

[2] ISO 11898-1:2015, “Road Vehicles-Controller Area Network (CAN)-Part 1: Data Link Layer and Physical Signaling,” https://www.iso.org/standard/63648.html

[3] ISO 17458-1:2013, “Road Vehicles-FlexRay Communications System-Part 1: General Information and Use Case Definition,” https://www.iso.org/standard/59804.html

[4] ISO 17458-2:2013, “Road Vehicles-FlexRay Communications System-Part 2: Data Link Layer Specification,” https://www.iso.org/standard/59806.html

[5] ISO 17458-3:2013, “Road Vehicles-FlexRay Communications System-Part 3: Data Link Layer Conformance Test Specification,” https://www.iso.org/standard/59807.html

[6] ISO 17458-4:2013, “Road Vehicles-FlexRay Communications System-Part 4: Electrical Physical Layer Specification,” https://www.iso.org/standard/59808.html

[7] ISO 17458-5:2013, “Road Vehicles-FlexRay Communications System-Part 5: Electrical Physical Layer Conformance Test Specification,” https://www.iso.org/standard/59809.html

[8] FlexRay Consortium, “FlexRay Communications System Protocol Specification, Version 3.0.1,” 2010.

[9] FlexRay Consortium, “FlexRay Communications System Electrical Physical Layer Specification Version 3.0.1,” 2010.

[10] FlexRay Consortium, “FlexRay Communications System Protocol Conformance Test Specification Version 3.0.1,” 2010.

[11] FlexRay Consortium, “FlexRay Preliminary Node-Local Bus Guardian Specification Version 2.0.9,” 2010.

[12] ISO/IEC 9646-1:1994, “Information Technology-Open Systems Interconnection-Conformance Testing Methodology and Framework,” https://www.iso.org/standard/17473.html

BIOGRAPHY

Seokjun Hahn (Member)

2017~2023 : Candidate for BS degree in Electronic Engineering, Soongsil University
2023~ : Candidate For MS degree in Department of Intelligent Semiconductors, Soongsil University
〈Main Interest〉 Automotive SoC, Processor SoC

Sua Shin (Member)

2020~ : Candidate for BS degree in Electronic Engineering, Soongsil University
〈Main Interest〉 Automotive SoC, AI SoC, Security SoC, Processor SoC

Naeun Park (Member)

2020~ : Candidate for BS degree in Electronic Engineering. Soongsil University
〈Main Interest〉 Automotive SoC, AI SoC, Processor SoC

Chan Park (Member)

2017~ : Candidate for BS degree in Electronic Engineering. Soongsil University
〈Main Interest〉 Automotive SoC, AI SoC

Daegi Lee (Member)

2016~2022 : Candidate for BS degree in Electronic Engineering, Soongsil University
2022~ : Candidate For MS degree in Department of Intelligent Semiconductors, Soongsil University
〈Main Interest〉 In-vehicle network IDS, Automotive SoC

Seongsoo Lee (Life Member)

1991 : BS degree in Electronic Engineering, Seoul National University.
1993 : MS degree in Electronic Engineering, Seoul National University.
1998 : PhD degree in Electrical Engineering, Seoul National University.
1998~2000 : Research Associate, University of Tokyo
2000~2002 : Research Professor, Ewha Womans University
2002~Now : Professor in School of Electronic Engineering, Soongsil University
〈Main Interest〉 AI SoC, Automotive SoC, Security SoC, Processor SoC, Power Management SoC, Battery Management SoC, Reliability and Safety