

# SystemC 기반의 FlexRay 통신 제어기 모델 설계

김우식<sup>1</sup>, 안진호<sup>2</sup>, 문병인<sup>1</sup>

<sup>1</sup>경북대학교 전자전기컴퓨터학부, <sup>2</sup>호서대학교 공과대학 전자공학과

전화: (053)950-8580, E-mail: wooss@ee.knu.ac.kr, bihmoon@knu.ac.kr

## FlexRay Communication Controller Model Architecture based on SystemC

Woosik Kim<sup>1</sup>, Jin-ho Ahn<sup>2</sup>, Byungin Moon<sup>1</sup>

<sup>1</sup>School of Electrical Engineering & Computer Science, Kyungpook National University

<sup>2</sup>Department of Electronic Engineering, College of Engineering, Hoseo University

### 요 약

본 논문은 차세대 차량용 백본 네트워크로 주목받고 있는 FlexRay 통신 프로토콜에 대한 연구를 진행하였다. 본 연구는 SystemC 모델링 언어를 사용하여 FlexRay 통신 프로토콜의 핵심부인 통신 제어기를 모델링함으로써 효율적인 설계 및 검증 환경을 구현하였다. 이를 위해 FlexRay 통신 프로토콜 표준 문서의 SDL(Specification and Description Language) 서술을 기초로 SystemC 기반의 모델을 설계하였으며, 이 과정에서 통신 제어기 내부의 코어 구조에 대한 분석과 SystemC 모델을 위한 추가적인 설계를 수행하였다. 설계된 모델은 CoWare사의 Platform Architect의 SystemC IDE를 통해 구현되었으며, 구현된 각 코어에 대한 검증을 수행하였다.

### Abstract

In this paper, we researched about the FlexRay Communication protocol which is rising up as the backbone network for the next generation vehicle. On this research, we implemented the effective design and verification environment for the FlexRay communication controller model based on SystemC which is the most important part of the FlexRay communication protocol. We designed and described the SystemC based model by analyzing the SDL description of FlexRay communication protocol standard documents. And we performed the analysis of the core structure of communication controller and added the additional design needed to implementation. The designed model is implemented and verified by SystemC IDE of CoWare Platform Architect.

**Keywords**: FlexRay, Vehicle network, SystemC, Next generation vehicle

### I. 서 론

최근 들어 주행 안정성을 높여주기 위한 다양한 차량용 제어 장치가 널리 탑재되고 있다. 그 예로, 브레이크

※ 본 연구보고서는 정보통신부 출연금으로 ETRI, SoC 산업진흥센터에서 수행한 IT SoC 핵심설계인력 양성사업의 연구결과입니다.

※ 본 연구보고서의 내용을 발표할 때에는 반드시 ETRI, SoC산업진흥센터 IT SoC 핵심설계인력 양성사업의 결과임을 밝혀야 합니다.

시 바퀴가 잠겨 도로 위를 미끄러지며 장애물을 피하지 못하는 것을 방지하는 ABS(Anti-lock Brake System)와 눈길이나 빙판길과 같은 미끄러운 노면에서 가속이 가능하도록 하는 TCS(Traction Control System)를 들 수 있다. 이러한 차량용 제어 기술은 차량에 장착된 ECU(Electronic Control Unit) 간의 고신뢰성 네트워크를 통해 구현된다. 제어 기술이 발전함에 따라 복잡한 제어와 연산을 하기 위한 더 많은 ECU가 필요하게 되었으며, 이를 위한 차량용 백본 네트워

크는 ECU들의 원활한 데이터 교환을 위한 넓은 대역폭과 데이터를 지연 없이 전달할 수 있는 실시간성이 필수 요소가 되었다.

기존의 차량용 백본 네트워크로 사용되어 오던 CAN(Controller Area Network) 통신 프로토콜은 최대 1Mbps의 속도와 일정 시간 내에 메시지 전달을 보장하지 못하는 CSMA/CD(Carrier Sense Multiple Access/Collision Detect) 전송 방식을 사용하고 있어 발전하고 있는 차량용 제어 기술의 요구사항을 만족하지 못한다.

이러한 CAN의 단점을 극복하여 차세대 차량용 백본 네트워크로 새롭게 떠오른 FlexRay 통신 프로토콜은 2000년, FlexRay 협회가 만들어진 이래로 하드웨어 및 프로토콜과 테스트 방법에 관한 표준안이 마련되어 왔다.

FlexRay는 듀얼 채널을 지원하고 최고 10Mbps의 속도를 지원하며, TDMA(Time Division Multiple Access) 및 FTDMA(Flexible Time Division Multiple Access) 전송을 통해 효율적이고, 신뢰성 높은 데이터의 전달을 지원하여 빠른 속도로 발전하고 있는 차량용 제어 기술을 위한 백본 네트워크로 적합하다.

FlexRay 통신 제어기(CC, Communication Controller)는 FlexRay 통신의 핵심 요소로서 고속, 고신뢰성 통신 네트워크를 구현한다. 하지만 이를 위한 동작이 복잡하며, 검증하기 위한 요구 사항 또한 방대하기 때문에 다양한 응용시스템에 통신 제어기를 적용하기 위해서는 FlexRay의 효율적인 설계 및 검증 환경이 요구된다.<sup>[1],[2],[3],[4]</sup>

본 논문은 SystemC와 CoWare사의 Platform Architect 툴을 사용하여 다양하고 복잡한 요구 사항을 만족하는 FlexRay 통신 제어기 모델을 설계하였다. 이를 통해 HDL 기반의 모델보다 시뮬레이션 속도가 빠른 H/W, S/W 통합 검증이 가능했다.

본 서론에 이어 II장에서는 FlexRay 통신 프로토콜에 대해 간략하게 설명하며, FlexRay 통신 네트워크의 핵심인 통신 제어기의 구조 및 기능에 대해 설명한다. 또한, 이를 SystemC 기반의 통신 제어기 모델로 구현하기 위한 설계 개요와 설계를 통해 구현된 모델에 대해 설명한다. III 장에서는 구현된 모델의 검증 절차와 그 결과에 대해 살펴보고, IV장에서 본 논문의 결론을 맺는다.

## II. 본 론

### 1. FlexRay 통신 프로토콜

FlexRay 통신 프로토콜은 두 개의 채널을 지원하여, 최고 20Mbps의 속도를 지원한다. 또한, TDMA 전송 방식을 사용한 정적 세그먼트 부분과 FTDMA 전송 방식을 사용한 동적 세그먼트 부분을 혼용하여 수신 측에서 특정 노드의 메시지의 도착 시점을 예측 할 수 있으며 동적 대역폭 특성을 만족한다. 이러한 장점으로 FlexRay는 기존의 CAN을 대체 할 수 있는 차량용 백본 네트워크로 주목 받고 있다.

### 2. FlexRay 통신 제어기

FlexRay 통신 네트워크의 노드는 그림 1과 같이 호스트, 통신 제어기와 버스 드라이버(BD, Bus Driver)로 구성된다.<sup>[1]</sup>

FlexRay 통신 제어기는 FlexRay 통신 네트워크의 핵심 부분으로 버스 드라이버를 통해 들어오는 메시지 데이터를 받아 처리하고, 현재 통신 제어기의 상태 정보를 호스트에 전달하며, 호스트로부터 받은 제어 신호 및 데이터를 적절히 처리하여 이를 버스 드라이버에 출력하는 역할을 한다.

통신 제어기는 크게 4개의 코어로 구성된다. 코딩디코딩, 미디어 접근 제어, 프레임과 심볼 처리, 클럭 동기기로 구분되며, 코어들은 더 작은 단위의 모듈로 나누어진다. 각 코어는 모드에 따라 동작하며, POC(Protocol Operation Control)는 CHI(Controller Host Interface)를 통해 전달되는 호스트의 명령을 처리하여 각 코어의 모드 변화를 포함한 현재 통신 제어기의 상태를 호스트에 제공한다.

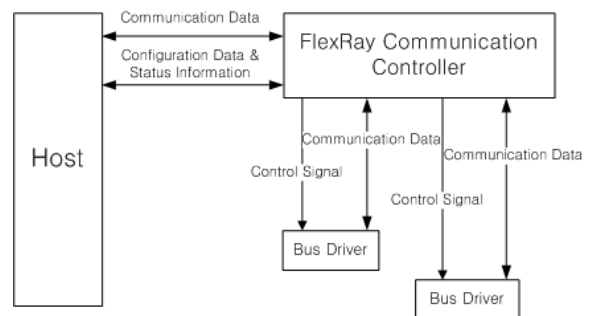


그림 1. FlexRay 노드의 구조

Fig 1. FlexRay node architecture

3. SystemC 기반 설계

SystemC는 시스템 레벨의 설계를 위해 C++ 클래스 라이브러리 형태로 만들어졌으며, 시스템 전체 모델링이 용이하며 HDL 코드 대비 빠른 시뮬레이션 속도와 기존의 C 개발 환경을 사용한다는 장점을 가진다.

본 논문에서는 FlexRay 프로토콜의 사양과 요구사항을 만족하는 통신 제어기를 효율적으로 설계하기 위해 SystemC 기반의 통신 제어기 모델을 설계하였다. 먼저, FlexRay 프로토콜 사양의 통신 제어기에 대한 SDL 서술을 분석하여 그림 2와 같이 각 코어간의 시그널의 교환과 코어가 사용하는 시그널을 정의하는 통신 제어기 모델을 설계하였다. 이 설계는 각 코어를 구성하는 알고리즘과 코어에서 사용하는 데이터 및 데이터 자료형의 정의를 포함하며, 코어 간에 교환하는 시그널의 정의와 이들을 구현하기 위한 클래스를 포함한다.

부가적으로, FlexRay 프로토콜 사양에서 별도로 정의하고 있지 않은 타이머 모듈과 코어들 간의 시그널의 교환을 관리하는 시그널 처리 부분을 포함하여 기존의 SDL 서술을 SystemC 기반으로 구현하는데 필요한 부가요소를 구현하였다. 이를 통해 통신 제어기의 요구사항을 만족하는 SystemC 기반의 FlexRay 통신 제어기를 설계 할 수 있다. 이렇게 설계한 모델을 Platform Architect의 SystemC IDE툴을 사용하여 SystemC 기반으로 구현하였다.

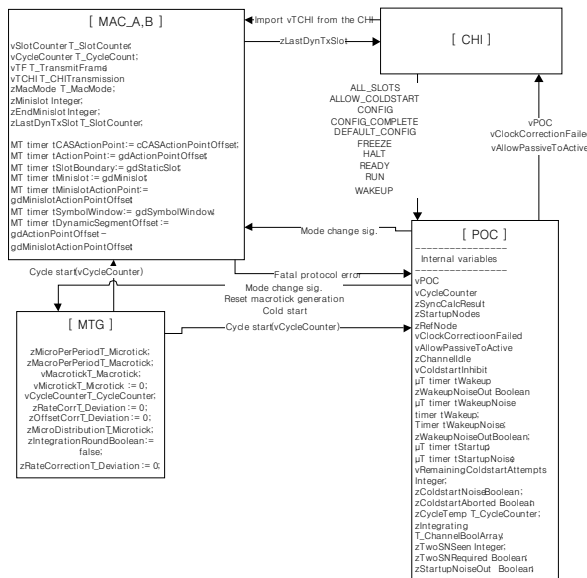


그림 2. FlexRay 통신 제어기 모델의 POC 부분 설계  
Fig 2. POC part of Communication Controller

III. 실험

1. 실험 환경

SystemC 기반으로 설계된 FlexRay 통신 제어기의 검증을 위해 Platform Architect의 SystemC IDE툴을 사용하여 설계한 코어들을 구현하였으며, 그림 3과 같이 Platform Creator를 사용하여 통합 검증용 플랫폼을 생성하여 코어들을 하나의 FlexRay 통신 제어기로 통합하여 동작을 확인하였다.

2. 실험 결과

코어의 동작을 확인하기 위해, SystemC IDE 툴의 trace 기능을 사용하여 코어에서 전달되는 시그널의 흐름을 분석하였다. 그림 4는 코어의 시뮬레이션 중 CODEC 코어의 시그널 교환을 추적한 결과를 나타낸다. CODEC코어에 해당하는 부분은 우측이며, POC코어의 메시지 전달에 대한 CODEC코어의 처리를 검증하기 위해 테스트 시그널을 임의로 발생시켰다.

그림 4는 3가지 구간으로 나누어 분석 할 수 있다. 0 ~ 20ns의 구간에서는 POC 코어가 CODEC 코어의 모드를 STANDBY로 변경하면, CODEC코어 내부에서 대기 중인 POC 코어의 시그널을 처리하는 이벤트 핸들러가 깨어나며, 메시지를 해석하여 CODEC 코어의 처리 루틴을 호출한다. 처리 루틴을 통해 CODEC 코어는 버스 드라이버와 하위 코어인 WUPDEC 코어와

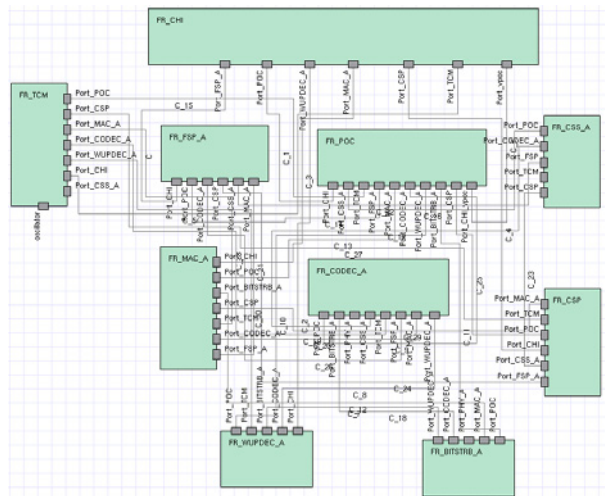


그림 3. 코어들이 통합된 통신 제어기 모델  
Fig 3. Combined Communication Controller model

BITSTRB 코어의 모드를 변경하는 시그널을 총 3회 발생시킨다

20ns 구간에서 POC 코어는 WUP(Wakeup Pattern)의 전송을 위한 인코딩 요청 메시지를 발생시켰으며 CODEC코어의 이벤트 핸들러가 메시지의 분석을 완료하여 처리 루틴을 호출한다.

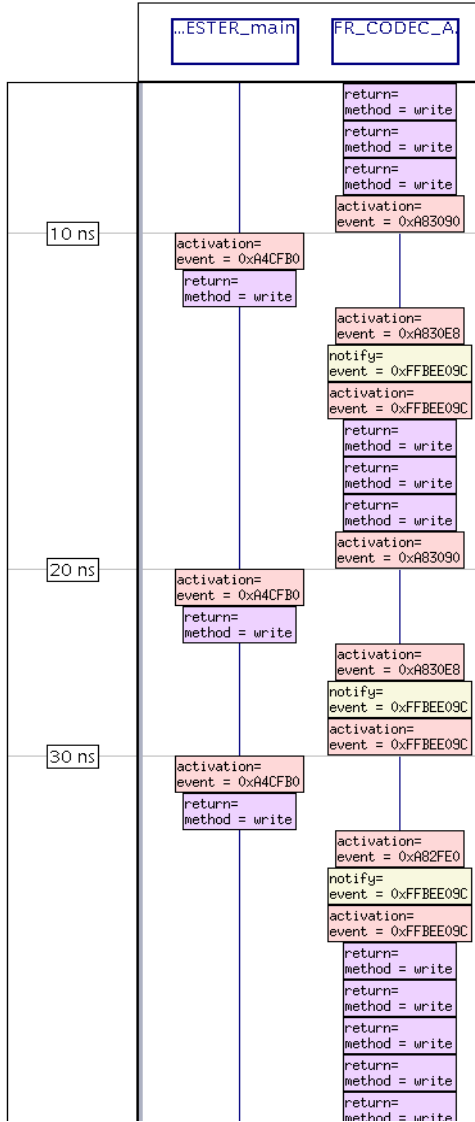


그림 4. CODEC 코어의 trace 결과  
Fig 4. Trace result of CODEC core

30ns이후 구간에서 CODEC 코어의 WUP\_ENCODING 매크로를 통해 버스 드라이버와 연결된 TxD와 TxEN 와이어의 값을 변경하여 FlexRay 통신 네트워크상에 WUP을 전송하며, "method = write" 부분으로 나타났다.

### 3. 결과 분석

HDL 시뮬레이션은 시뮬레이터의 처리 속도가 느리고, 분석 과정에서 파형 분석을 통한 검증만이 가능하며 FlexRay와 같은 복잡한 알고리즘을 구현하는 모델을 검증하는데 적합하지 않다. SystemC를 통해 구현된 FlexRay 통신 제어기 모델은 코어간의 시그널링을 시각적인 trace 과정을 통해 쉽게 분석할 수 있을 뿐만 아니라 구현상의 오류 및 SDL 서술을 SystemC 모델로 설계하는 과정에서 발생할 수 있는 문제점들에 대한 확인이 쉽게 가능하여 모델 설계 시간의 단축이 가능하다.

### IV. 결론

본 논문은 다양한 요구조건을 만족시켜야 하는 FlexRay 통신 제어기를 시뮬레이션 속도가 빠르고 오류의 확인과 수정이 쉬운 SystemC 기반의 모델로 설계하였다. 본 연구를 통하여 FlexRay 통신 제어기의 개발 과정에서 발생할 수 있는 문제점을 본 논문의 모델을 이용해 신속하게 대응 및 수정할 수 있으며, 이를 활용한 다양한 응용시스템의 개발에 기여할 것으로 기대한다.

### 참고 문헌

- [1] FlexRay Communication System Protocol Specification Version 2.1 Revision A, 2005
- [2] FlexRay Communications System Electrical Physical Layer Specification Version 2.1 Revision B, 2006
- [3] FlexRay Requirements Specification Version 2.1, 2005
- [4] FlexRay Electrical Physical Layer Specification Version 2.1 Revision B, 2006