

논문 2008-45TC-6-1

FlexRay 프로토콜 설계 및 로봇 시스템 응용

(Implementation of FlexRay Communication Controller Protocol and its Application to a Robot System)

강 현 수*, 허 일 남*, 김 용 은*, 정 진 균**

(Hyun-Soo Kang, Yi-Nan Xu, Yong-Eun Kim, and Jin-Gyun Chung)

요 약

FlexRay는 차세대 차량 내 전자 장치간의 통신을 위해 고속의 시리얼 통신, time triggered bus, fault tolerant 통신을 제공하는 새로운 네트워크 통신 시스템의 표준이다. FlexRay Communication Controller(CC)는 FlexRay 프로토콜 규격의 핵심 부분이다. 본 논문에서는 먼저 SDL(Specification and Description Language)를 이용하여 FlexRay CC 프로토콜 규격과 기능 부분을 설계한다. 다음 설계한 SDL 소스를 기반으로 Verilog HDL을 이용하여 하드웨어로 설계한다. 설계한 FlexRay CC는 Samsung 0.35 μm 공정을 이용하여 합성하였으며, 그 결과 80 MHz의 속도로 동작하는 것으로 나타났다. 또한 FlexRay 시스템의 동작을 확인하기 위해 로봇에 적용되는 음원위치 추정 시스템에 응용하였다. 응용 시스템은 ALTERA Excalibur ARM EPAX4F672C3을 이용하여 검증하였으며 성공적으로 동작함을 확인하였다.

Abstract

FlexRay is a new standard of network communication system which provides a high speed serial communication, time triggered bus and fault tolerant communication between electronic devices for future automotive applications. FlexRay communication controller (CC) is the core of the FlexRay protocol specification. In this paper, we first design the FlexRay CC protocol specification and function parts using SDL (Specification and Description Language). Then, the system is re-designed using Verilog HDL based on the SDL source. The FlexRay CC system was synthesized using Samsung 0.35 μm technology. It is shown that the designed system can operate in the frequency range above 80 MHz. In addition, to show the validity of the designed FlexRay system, the FlexRay system is combined with sound source localization system in Robot applications. The combined system is implemented using ALTERA Excalibur ARM EPXA4F672C3. It is shown that the implemented system operates successfully.

Keywords : FlexRay, Communication Controller, SDL, Verilog HDL

I. 서 론

FlexRay는 차세대 차량 내 전자 장치간의 통신을 위해 고속의 시리얼 통신, time triggered bus, fault tolerant 통신을 제공하는 새로운 네트워크 통신 시스템의 표준이다.^[1] FlexRay는 time-triggered 방식과 부가

적인 event-triggered 방식을 지원한다. 최대 데이터 전송 속도는 10 Mbps이며, 2개의 채널을 통해 동시에 전송할 수 있다.^[2]

FlexRay는 차세대 자동차를 위한 BMW, Bosch, DaimlerChrysler, Philips 등이 2000년도에 설립한 컨소시엄을 통해 개발되었다.^[3] 컨소시엄은 현재 Freescale Semiconductors, Bosch, General Motors, 현대기아자동차 등 여러 자동차와 반도체 회사들로 구성되어 있다. FlexRay 시스템은 2006년에 BMW X5 시리즈 차량의 전자제어장치에 처음으로 적용되었으며, 현재 FlexRay 규정은 계속 수정 중에 있다.

본 논문에서는 먼저 SDL(Specification and Descrip-

* 학생회원, ** 정회원, 전북대학교 전자정보공학부 (Div. of Electronic & Information Engineering Chonbuk National University)

※ 이 연구에 참여한 연구자는 2단계 BK21 사업의 지원비를 받았음, This work was supported by the second stage of Brain Korea 21 Project.

접수일자: 2008년3월19일, 수정완료일: 2008년6월19일

tion Language)을 이용하여 FlexRay CC 프로토콜 규격과 기능 부분을 설계한다.^[4~5] 설계한 SDL을 기반으로 Verilog HDL을 이용하여 하드웨어로 설계한다. 설계한 FlexRay CC는 Samsung 0.35 μm 공정을 이용하여 합성하였으며, 그 결과 80 MHz의 속도로 동작하는 것으로 나타났다. FlexRay 시스템의 동작을 확인하기 위해 로봇에 적용되는 음원위치 추정 시스템에 응용하였다. 응용 시스템은 ALTERA Excalibur ARM EPAX4F672C3을 이용하여 검증하였으며 성공적으로 동작하는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 먼저 II장에서는 FlexRay 구조를 간략하게 살펴본다. III장에서는 FlexRay 시스템을 SDL을 이용하여 설계하고, IV장에서는 Verilog HDL을 이용하여 설계하고 로봇 시스템에서 사용되는 음원위치 추정 시스템과 연동하여 실험한다. 마지막으로 V장에서는 결론을 맺는다.

II. FlexRay 시스템 구조

1. FlexRay 노드 구조와 버스 구성

FlexRay 노드 구조는 그림 1과 같이 host, communication controller(CC), bus guardian(BG), bus driver(BD)로 구성된다. Host는 사용자의 소프트웨어로 통신프로세스를 제어하고, CC는 노드의 핵심부분으로서 FlexRay 통신 프로토콜을 구성하는 주요한 부분이다. BG는 버스 액세스(access)를 감시하는 역할을 하고, BD는 노드를 채널에 연결하여 데이터를 송, 수신하는 역할을 담당한다.

FlexRay 채널은 기본적으로 두개의 채널로 구성되며 각 노드와 노드사이의 데이터 전송은 두개의 채널을 모

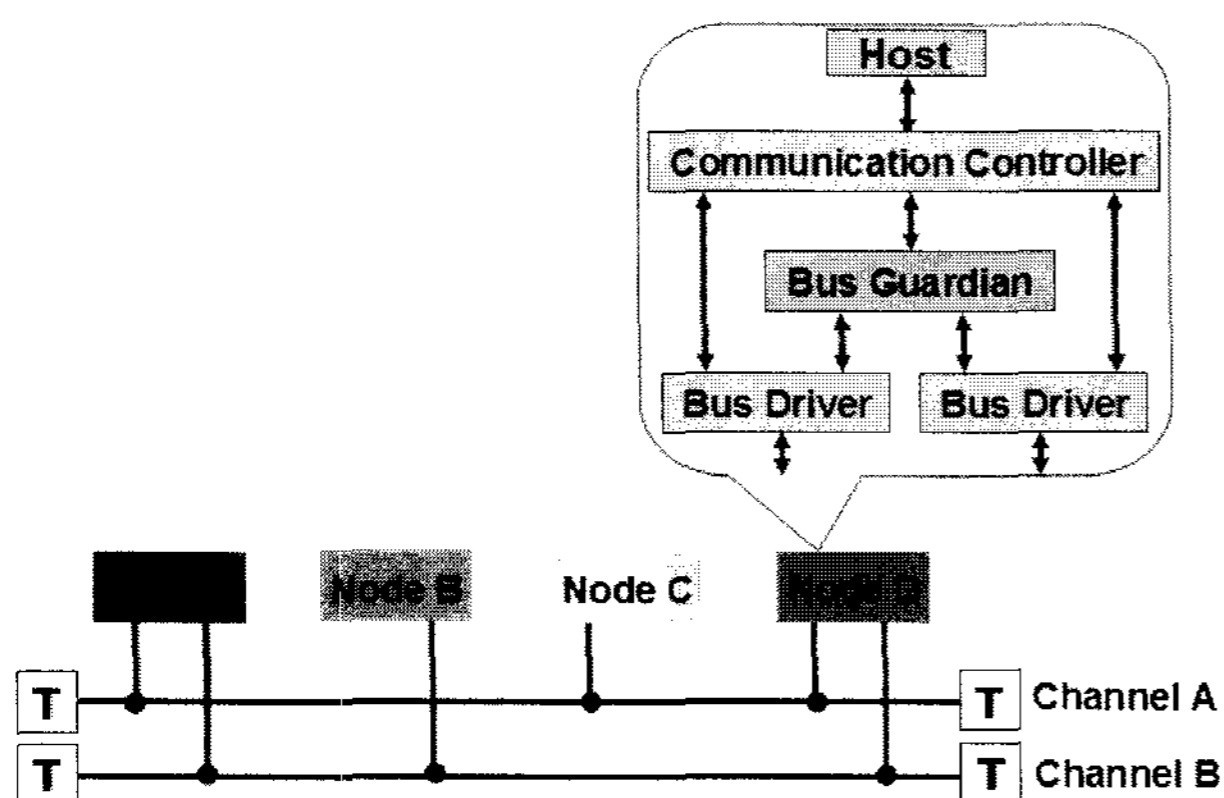


그림 1. FlexRay 노드 구조 및 듀얼 버스의 배치

Fig. 1. FlexRay node architecture and dual bus topology configuration.

두 사용할 수도 있고 하나의 채널만 사용할 수도 있다. 두 채널은 보통 이중화(redundant)를 위해 사용되므로 fault-tolerant한 메시지전송을 위해 쓰이지만 서로 다른 메시지를 전송할 수도 있으며 이 경우 데이터 전송률은 두 배가 된다. 그림 1과 같이 노드 A와 노드 D의 듀얼버스는 채널 A와 채널 B에 각각 연결되어 있고, 노드 B는 채널 B에만, 노드 C는 채널 A에만 연결되어 있으며 채널의 끝단부분에 90-100 Ohm의 종단 저항을 연결하게 된다. FlexRay에서는 싱글버스를 사용할 수도 있으며 이 경우 모든 노드는 싱글버스에 연결된다. 또한 스타(star) 토폴로지를 사용할 수도 있으며 버스와 스타의 다양한 하이브리드 네트워크를 구성하여 사용할 수도 있다^[3].

2. FlexRay 타이밍 계층구조

FlexRay를 사용한 통신방식은 그림 2와 같이 정기적으로 순환되는 통신 사이클로 구성된다. 하나의 통신 사이클은 static segment(ST), dynamic segment(DYN), symbol window(SW), network idle time(NIT)으로 구성된다. 이러한 통신 사이클은 정적인(static) 시분할 다중접속(TDMA)방식과, 동적인(dynamic) 미니슬롯(minislot) 기반 방식인 FTDMA(flexible time division multiple access) 방식을 이용한다. 통신 사이클은 최대 64개의 사이클로 구성되는데 매 사이클은 communication cycle, arbitration grid, macrotick, microtick 레벨의 4 계층으로 나눌 수 있다.

ST에서는 TDMA방식을 이용하고 DYN에서는 동적인 미니슬롯기반 방식 (FTDMA)에 따라서 데이터의 전송이 이루어진다. 정기적 데이터 전송이 중요한(또는 time-critical) 메시지는 ST에 할당하여 안정된 전송이 이루어지게 하며, 이벤트(event)에 의해 생성되는 메시지나 지연 특성이 중요하지 않은 메시지는 DYN에 할당하여 대역폭이 실제로 필요한 경우에만 사용되게 함으로써 버스대역폭을 최대한으로 활용하게 된다. SW에서는 미리 정의된 symbol 세트 중 하나의 symbol이 전송되어 테스트 목적 등에 사용될 수 있다. NIT는 유휴 통신 구간으로 클록의 보정 등에 사용한다.

Arbitration grid 레벨에서 ST는 정적슬롯(static slot)으로 정의된 연속적인 시간구간으로 grid가 구성되고, DYN은 미니슬롯으로 정의된 연속적인 시간구간으로 grid가 구성된다. 시간 슬롯은 미니슬롯 내에서 버스 액세스가 발생할 때 요구되는 시간에 의해서만 확장되므로 대역폭은 실제로 필요할 경우에만 사용된다.

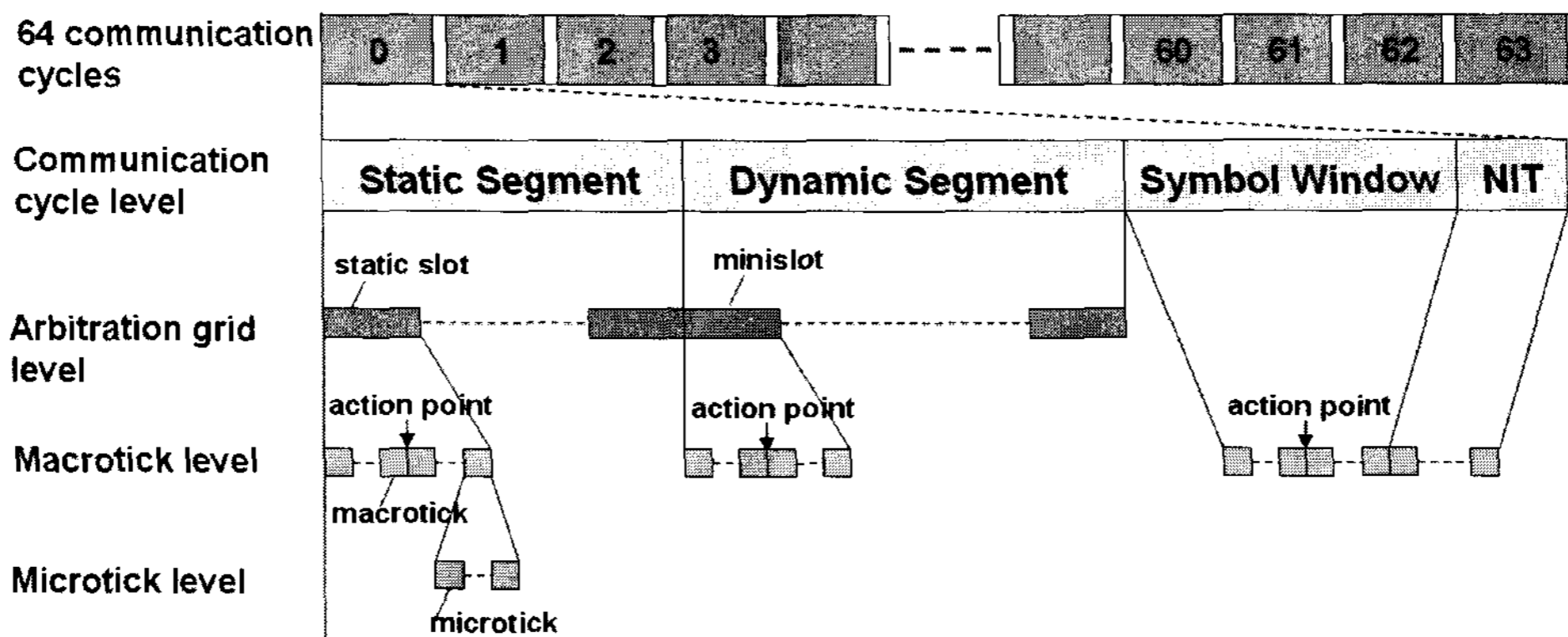


그림 2. FlexRay 타이밍 계층구조
Fig. 2. FlexRay timing hierarchy.

Macrotick 레벨은 macrotick으로 구성되고 ST, DYN, SW에서 action point라고 불리는 경계선을 가지고 있다. ST와 SW에서 action point는 프레임의 시작을 알려주고 DYN에서는 프레임의 시작과 끝을 알려준다.

Microtick 레벨은 microtick이라는 시간단위로 구성되는데 microtick은 컨트롤러의 오실레이터 클록으로부터 prescaler 등을 이용하여 유도되는 시간단위이고 컨트롤러마다 서로 다른 주기를 갖는 microtick을 정의할 수 있다.

3. FlexRay 프레임 형식

FlexRay 프레임은 그림 3과 같이 header segment, payload segment, trailer segment로 이루어진다. 프레임의 처음 5 bits는 프레임의 기본 정보를 포함하고 있다. Frame ID(11 bits)는 ST에서의 slot 위치를 나타내고, DYN에서는 프레임의 우선순위(낮은 ID 값이 더 높

은 우선순위를 가진다)를 나타내기 위해 사용된다. Payload Length(7 bits)는 데이터의 길이(payload length x 2 = data bytes의 길이)를 나타낸다.

Header CRC(11 bits)는 Sync Frame Indicator(1 bit), Startup Frame Indicator(1 bit), Frame ID(11 bits), Payload Length(7 bits)를 가지고 계산한 Cyclic Redundancy Check이다. Cycle Count(6 bits)는 각 노드의 frame 순번이다. Payload Segment(0~254 bytes)에는 실제 데이터가 포함된다. Trailer Segment(24 bytes)는 header segment와 payload segment를 가지고 계산한 Cyclic Redundancy Check이다.

III. SDL을 이용한 FlexRay 설계

SDL(Specification and Description Language)은 ITU Telecommunication Standardization Sector (ITU-T)에서 Z.100으로 표준화 된 추상 시스템 디자인 언어이다.^[4] SDL은 그래픽 다이어그램으로 구현되어 있어 C나 C++ 및 기타 언어로 작성된 문서보다 구조 및 원리를 알아보기 쉽고 SDL로 정의된 시스템을 분석하고 각 단계별로 시뮬레이션이 가능하여 오류나 인터페이스의 불일치를 쉽게 발견할 수 있으므로 시스템의 개발 시간을 줄이고 쉽게 소스를 수정 및 변화시킬 수 있는 장점을 가지고 있다.^[5~8]

FlexRay CC 시스템은 Protocol Operation Control (POC), Coding and Decoding Process (CODEC_A, CODEC_B), Bit Strobing Process (BITSTRB_A, BITSTRB_B), Wakeup Pattern Decoding Process (WUPDEC_A, WUPDEC_B), Media Access Control Process(MAC_A, MAC_B), Frame and Symbol

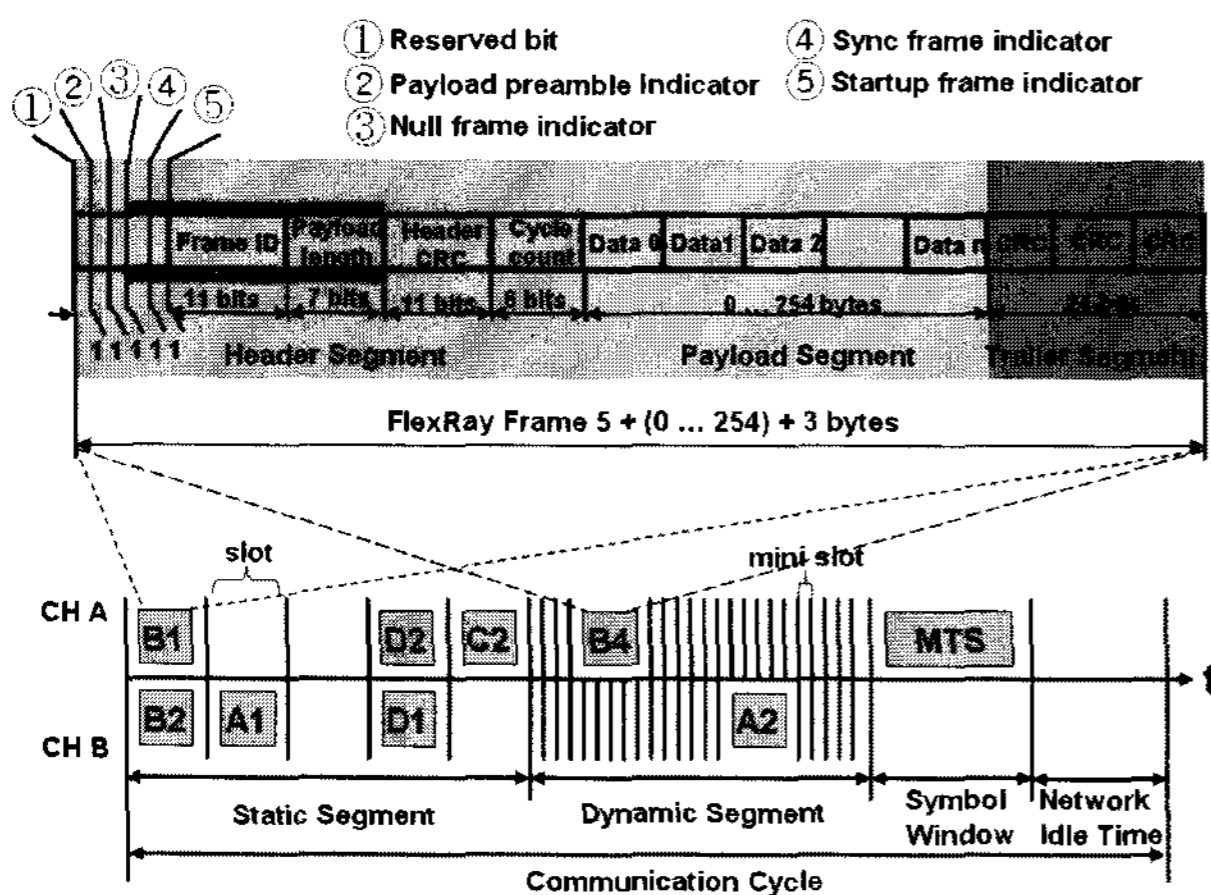


그림 3. FlexRay 프레임 형식 및 통신 사이클
Fig. 3. FlexRay frame format and communication cycle.

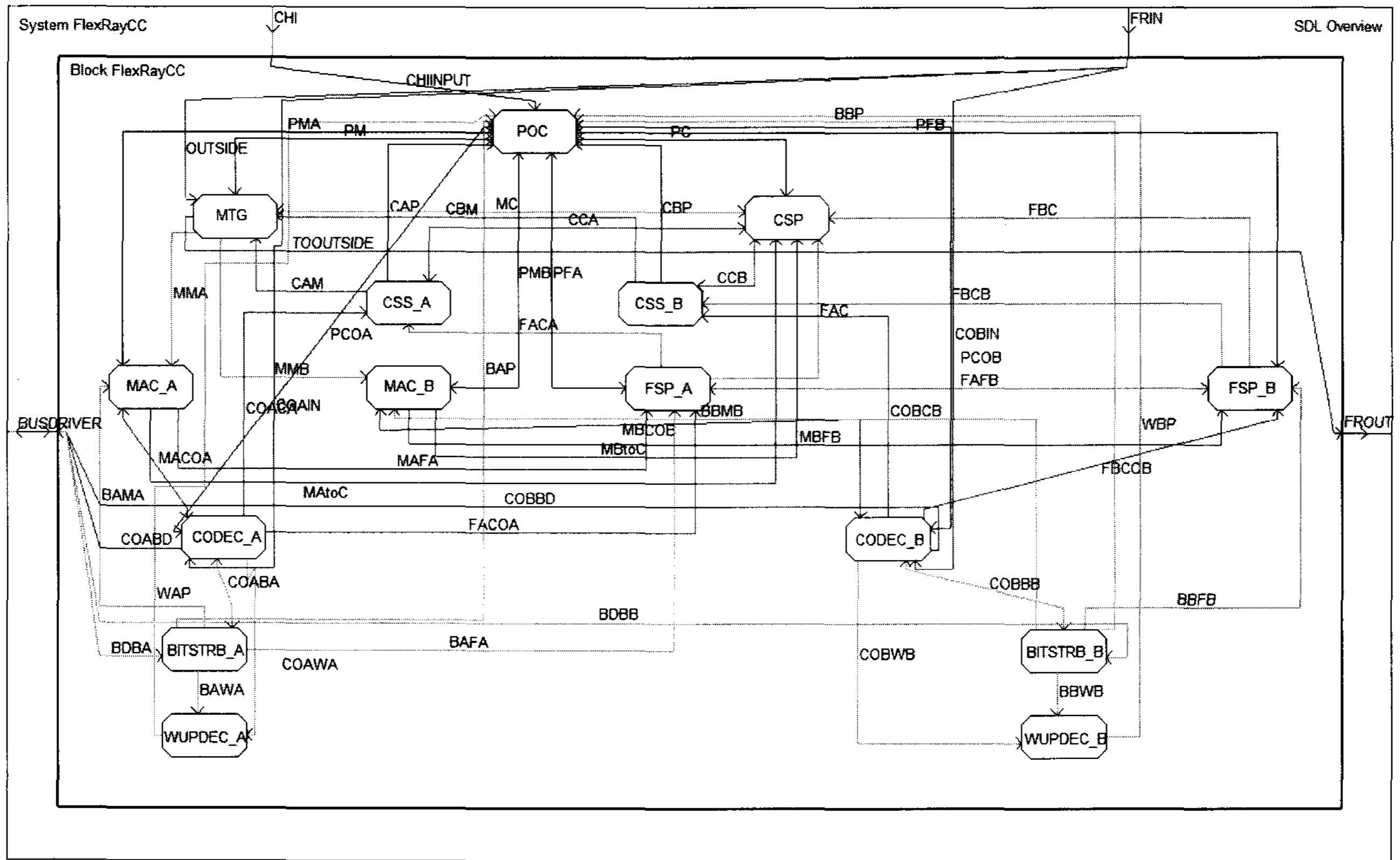


그림 4. FlexRay CC 시스템의 계층적 구조
 Fig. 4. Hierarchical structure of FlexRay CC System.

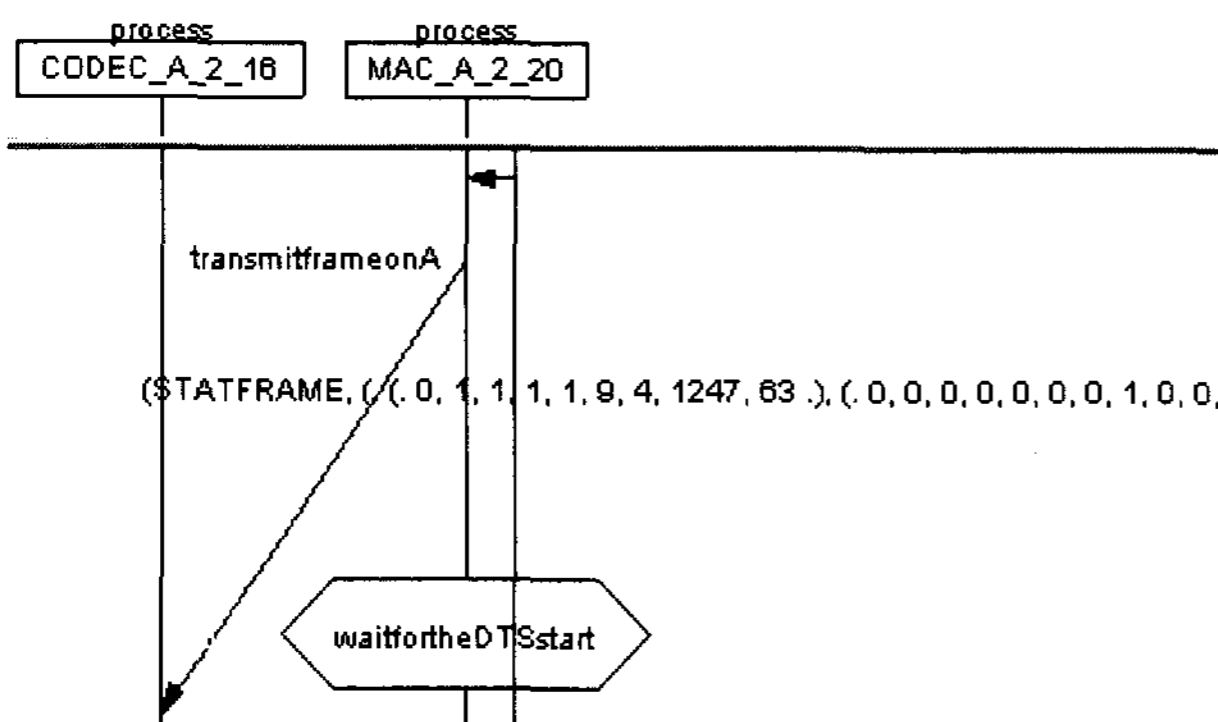


그림 5. MSC 시뮬레이션 결과의 예
 Fig. 5. Example of MSC simulation result.

Processing(FSP_A, FSP_B), Clock Synchronization Process(CSP), Clock Synchronization Startup Processing(CSS_A, CSS_B), Macrotick Generation Process(MTG)의 15개 블록으로 구성된다.

그림 4는 FlexRay CC 통신 프로토콜을 SDL로 상세 설계한 것이다. 이와 같이 구성하여 모의수행을 실행하면 실제 통신에서와 같은 환경으로 동작할 수 있어 검증과 에러 수정에 용이하다.

MSC(Message Sequence Chart)는 SDL로 설계한 FlexRay 시스템의 검증 및 시뮬레이션을 위해 사용된

다. 그림 5는 MSC 시뮬레이션 결과의 예로써 MAC_A 블록에서 CODEC_A 블록으로 프레임이 전송되는 과정이다.

IV. Verilog HDL을 이용한 FlexRay 설계 및 검증

FlexRay 통신 프로토콜은 FSM(Finite State Machine) 형식의 다이어그램으로 구현되어 있다. FSM 방식은 통신 프로토콜과 같은 블록이 크고 복잡한 시스템을 구현할 때 주로 사용하는 방식으로서, 통신 프로토콜을 이해하기 쉽고 소프트웨어 및 하드웨어로 비교적 쉽게 구현할 수 있는 장점을 가지고 있다.^[9~10]

하드웨어 설계를 위해 설계한 SDL 결과를 기반으로 Verilog HDL을 이용하여 FlexRay 시스템을 재설계하였다. 설계한 시스템은 Synopsys Design Compiler로 Samsung 0.35 μm cell library를 이용하여 합성하였다. 합성한 결과 약 53,000 개의 게이트로 구성되며 15 mW의 파워 소비와 80 MHz의 속도에서 동작하는 것으로 나타났다. 그림 6은 FlexRay CC를 합성한 결과이다.

검증을 위해 Cadence Verilog-XL과 SimVision을 이

용하였다.

그림 7은 SimVision으로 타이밍 시뮬레이션한 결과를 나타낸다. 채널 A와 B에서 전송프레임(vTF)을 각각 전송하고 이를 다시 받아 디코딩하여 수신프레임(vRF)을 얻는다. 송신 노드에서는 TxEN 신호가 low일 때만 한 비트의 프레임 데이터를 8 클럭 동안 TxD 신호를 통해 전송된다. 수신 노드에서는 전송된 신호를 RxD 신호와 RxEN 신호로 받으며, RxEN 신호가 low일 때

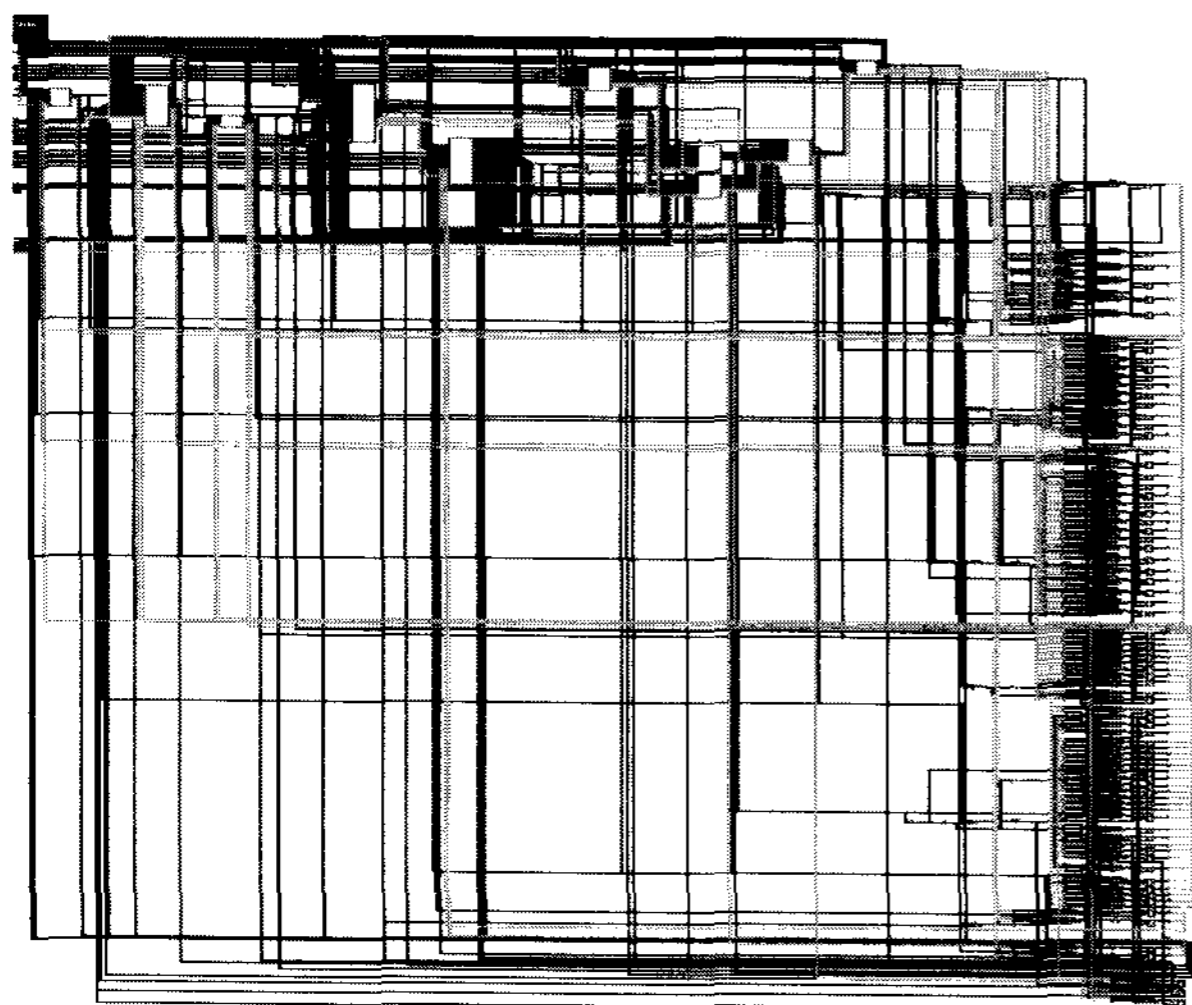


그림 6. 합성한 FlexRay CC 시스템의 구조
Fig. 6. Schematic of synthesised FlexRay CC System.

만 RxD 신호를 받아 디코딩한다. 8 클럭 동안 한 비트의 데이터를 받기 위해 8번 샘플링하여 5번째로 샘플링된 값을 수신된 프레임 데이터로 처리한다. 이 샘플링 위치는 bitstrobedonA 신호로 나타낸다.

설계한 FlexRay CC 시스템의 검증을 위해, 로봇에서 사용되는 음원위치 추정 시스템에 응용하였다. 그림 8은 음원위치 추정 시스템의 구조이다. 외부에서 소리가 발생하면 로봇의 머리에 설치된 4개의 마이크로부터 서로 다른 방향에서 들리는 4개의 데이터를 얻게 된다. 이 데이터들을 FFT, AE-CORDIC, CORDIC, IFFT 과정을 거쳐 실제 소리가 들린 위치를 추정한다.

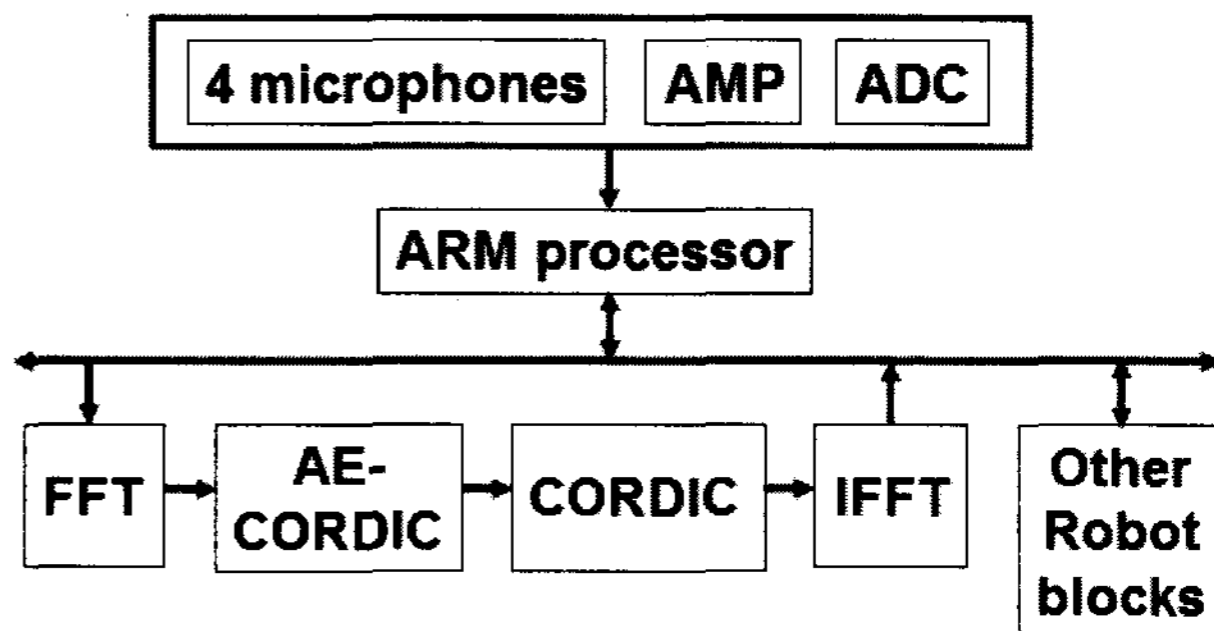


그림 8. 음원위치 추정 시스템의 블록 다이어그램
Fig. 8. Block diagram of sound source localization system.

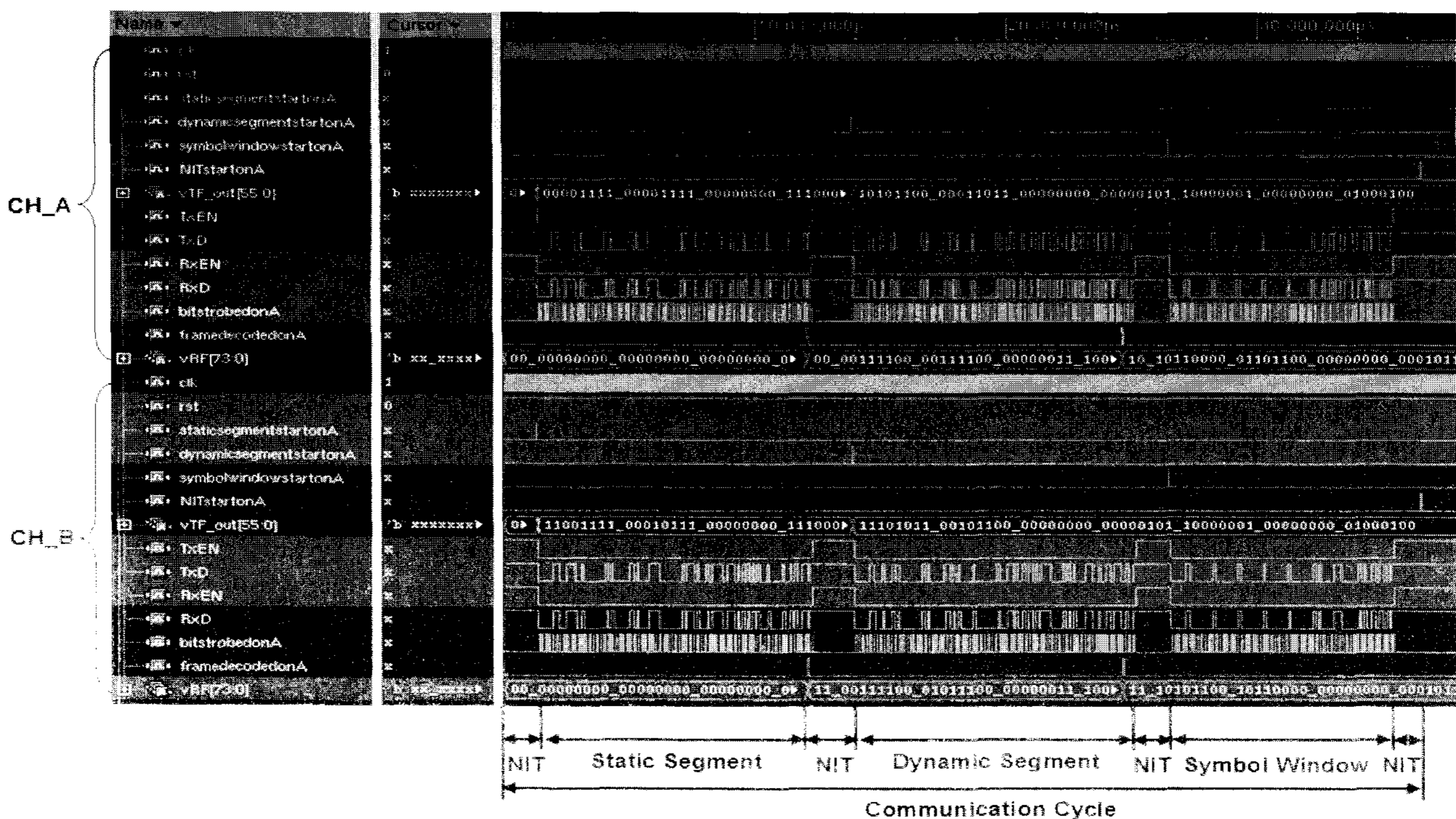


그림 7. 송신 및 수신 프레임의 시뮬레이션 결과
Fig. 7. Simulation result of transmit and received frame.

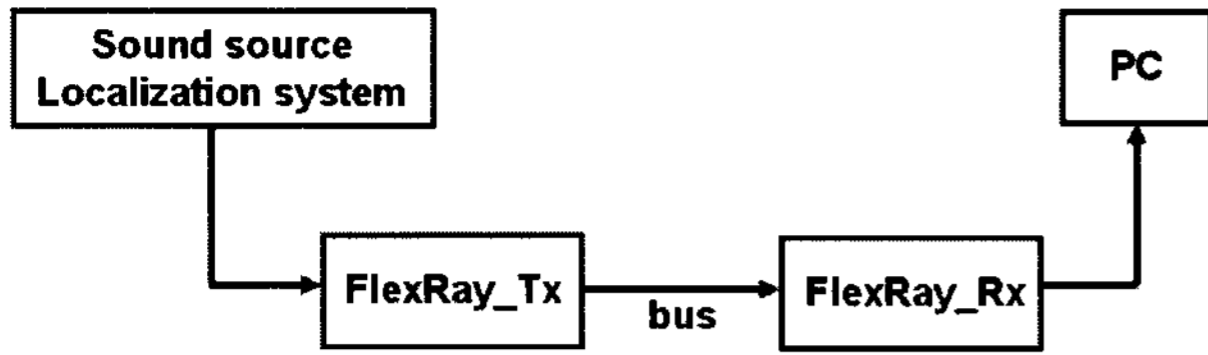


그림 9. 테스트 환경의 블록 다이어그램
Fig. 9. Block diagram of test environment.

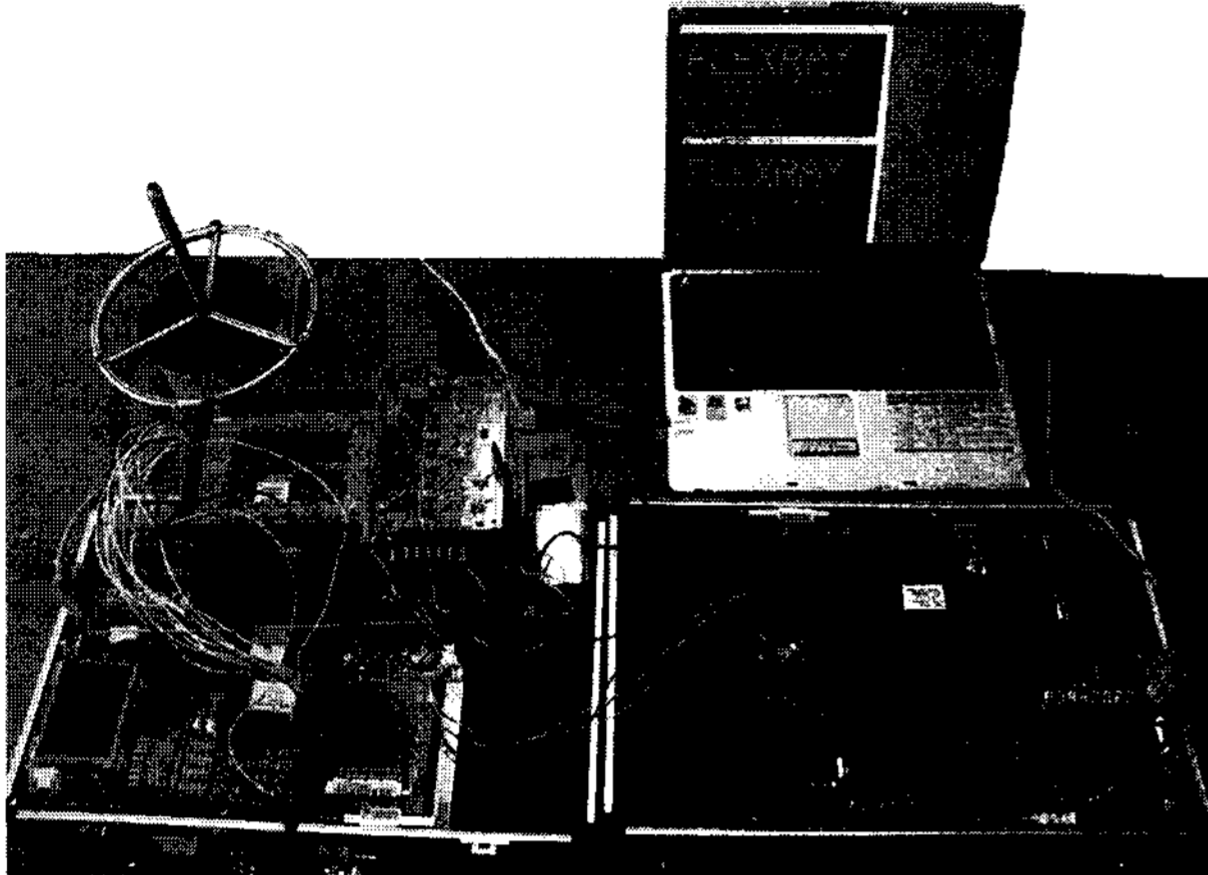


그림 10. FlexRay 네트워크의 실제 환경 테스트
Fig. 10. Actual test environment of FlexRay network.

음원위치 추정 시스템에서 계산된 데이터는 설계된 FlexRay 시스템의 host로 전송된다. 이 데이터를 가지고 송신노드에서 전송프레임을 생성하여 신호를 전송하고, 버스를 통하여 이 신호를 수신노드에서 받아 디코딩하여 얻은 수신프레임으로부터 전송한 데이터를 확인한다. 이 테스트 환경의 블록 다이어그램은 그림 9와 같다. 효율적인 설계를 위해 음원위치 추정 시스템과 FlexRay 시스템을 하나의 칩으로 설계하였다. 설계과정에서 사용된 FPGA는 ALTERA Excalibur ARM EPXA4F672C3이며, 그림 10과 같이 PC를 통하여 데이터 전송이 성공적으로 이루어짐을 확인하였다.

V. 결 론

본 논문에서는 차세대 차량 내 통신네트워크에 사용될 FlexRay Communication Controller 프로토콜의 설계 및 검증에 관한 연구를 수행하였다. FlexRay 프로토콜을 설계하기 위하여 SDL을 이용하여 설계하였다. 설계한 SDL을 기반으로 Verilog HDL을 이용하여 하드웨어로 설계하였다. 설계한 시스템은 Synopsys Design Compiler로 Samsung 0.35 μm cell library를 이용하여 합성하였다. 설계한 시스템은 약 53,000개의 게이트로 구성되며, 동작속도 80 MHz에서 15 mW의 전력을 소

모한다.

설계한 FlexRay CC 시스템의 검증을 위해, 로봇에서 사용되는 음원위치 추정 시스템에 응용하였다. 응용 시스템은 ALTERA Excalibur ARM EPXA4F672C3 FPGA를 사용하여 설계하였다. 음원위치 추정 시스템에서 계산된 데이터는 설계된 FlexRay 시스템의 host로 전송된다. 수신된 데이터를 확인하여 설계한 FlexRay CC 시스템이 성공적으로 동작함을 검증하였다.

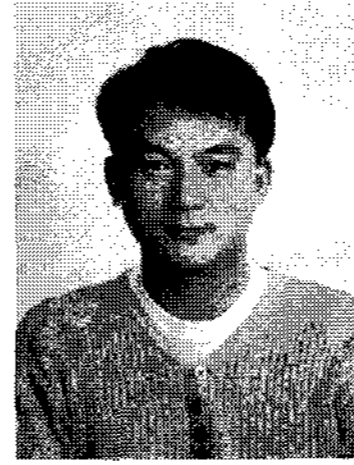
참 고 문 헌

- [1] FlexRay Communications System Protocol Specification v2.1 Revision A, 2005.
- [2] Dominique Paret. *Multiplexed Networks for Embedded Systems*. John Wiley & Sons, Ltd, 2007.
- [3] FlexRay Consortium, <http://www.flexray.com>.
- [4] ITU-T Z.100, CCITT Specification and Description Language (SDL), 1996.
- [5] Homepage of the SDL-Forum, <http://www.sdl-forum.org>.
- [6] Ferenc Belina, Dieter Hogrefe and Amardeo Sarma. *SDL with Applications from Protocol Specification*. Prentice Hall, 1991.
- [7] Laurent Doldi. *Validation of Communications Systems with SDL*. John Wiley, 2003.
- [8] DECOMSYS GmbH, <http://www.decomsys.com>.
- [9] Michael D. Ciletti. *Advanced Digital Design with the Verilog HDL*. Prentice Hall, 2002.
- [10] J. Bhasker. *A Verilog HDL Primer*. Star Galaxy Publishing, 2005.

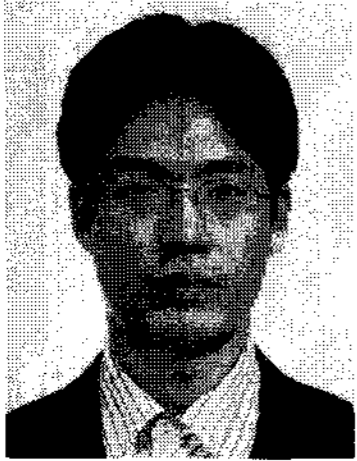
저 자 소 개



강 현 수(학생회원)
 2006년 전북대학교 전자공학과
 학사 졸업
 2006년~현재 전북대학교 전자
 정보공학부 석사
 <주관심분야 : 통신, 신호처리, 반
 도체>



허 일 남(학생회원)
 1999년 중국-연변대학교
 전자정보공학과 학사 졸업
 2003년 전북대학교 물리기술학과
 석사 졸업
 2005년~현재 전북대학교
 전자정보공학과 박사
 <주관심분야 : 통신, 신호처리, 반도체>



김 용 은(학생회원)
 2005년 전북대학교 전자공학과
 학사 졸업
 2007년 전북대학교 정보통신
 공학과 석사 졸업
 2007년~현재 전북대학교 전자
 정보공학부 박사
 <주관심분야 : 통신, 신호처리, 반도체>



정 진 균(정회원)
 1985년 전북대학교 전자공학
 학사 졸업
 1989년 미국 미네소타 주립대학
 전기공학 석사 졸업
 1991년 미국 미네소타 주립대학
 전기공학 박사 졸업
 <주관심분야 : 통신, 컴퓨터, 신호처리, 반도체>