

논문 2008-45TC-8-14

FlexRay 프로토콜 설계 및 자동차 경보 시스템 응용

(Implementation of FlexRay Protocol Specification and its Application to a Automobile Advance Alarm System)

허일남*, 양상훈*, 정진균**

(Yi-Nan Xu, Sang-Hoon Yang, and Jin-Gyun Chung)

요약

FlexRay는 높은 유연성과 신뢰성을 갖는 고속의 통신프로토콜이다. 자동차회사와 반도체회사들에 의해 개발되었고 x-by-wire 시스템을 이용하여 차량 내 LAN으로 구현된다. FlexRay는 차량 내 전자 장치간의 통신을 위해 고속의 시리얼 통신, time triggered bus, fault tolerant 통신을 제공한다. 본 논문에서는 먼저 SDL(Specification and Description Language)을 이용하여 FlexRay communication controller와 bus guardian 프로토콜 규격과 기능 부분을 설계한다. 다음 설계한 SDL 소스를 기반으로 Verilog HDL을 이용하여 하드웨어로 설계한다. 설계한 FlexRay 시스템은 Samsung 0.35 μm 공정을 이용하여 합성하였으며, 그 결과 76 MHz의 속도로 동작하는 것으로 나타났다. 또한 FlexRay 시스템의 동작을 확인하기 위해 차량에 적용되는 자동차 경보 시스템에 응용하였다. FlexRay 시스템은 ALTERA Excalibur ARM EPAX4F672C3을 이용하여 검증하였으며 성공적으로 동작함을 확인하였다.

Abstract

FlexRay is a high-speed communications protocol with high flexibility and reliability. It was devised by automotive manufacturers and semiconductor vendors and implemented as on vehicle LAN protocol using x-by-wire systems. FlexRay provides a high speed serial communication, time triggered bus and fault tolerant communication between electronic devices for automotive applications. In this paper, we first design the FlexRay communication controller, bus guardian 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 system was synthesized using Samsung 0.35 μm technology. It is shown that the designed system can operate in the frequency range above 76 MHz. In addition, to show the validity of the designed FlexRay system, the FlexRay system is combined with automobile advance alarm system in vehicle applications. The FlexRay system is implemented using ALTERA Excalibur ARM EPXA4F672C3. It is shown that the implemented system operates successfully.

Keywords : FlexRay, X-by-wire, SDL, Verilog HDL, Automobile

I. 서론

차량의 편의주행, 안전주행에 대한 요구가 증가됨에 따라 차량 내 전자 제어 유닛 (Electronic Control Unit, ECU)이 많아지면서 배선 무게가 증가되고 이로 인한

호환문제 및 성능저하 등 문제가 발생한다. 이에 따라, 최근에는 차량의 전자제어 유닛 시스템에 통신 네트워크를 접목시킨 차량 내 네트워크 (In-Vehicle Networking, IVN)가 활발히 연구되고 있다.^[1]

FlexRay는 차세대 차량 내 전자 장치간의 통신을 위해 고속의 시리얼 통신, time triggered bus, fault tolerant 통신을 제공하는 새로운 네트워크 통신 시스템의 표준이다.^[2] FlexRay는 time-triggered 방식과 부가적인 event-triggered 방식을 지원한다. 최대 데이터 전송 속도는 10 Mbps이며, 2개의 채널을 통해 동시에 전송할 수 있다.^[3]

* 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: 2008년6월4일, 수정완료일: 2008년8월12일

FlexRay는 차세대 자동차를 위한 BMW, Bosch, DaimlerChrysler, Philips 등이 2000년도에 설립한 컨소시엄을 통해 개발되었다.^[4] 컨소시엄은 현재 Freescale Semiconductors, Bosch, General Motors, 현대기아자동차 등 여러 자동차와 반도체 회사들로 구성되어 있다. FlexRay 시스템은 2006년에 처음으로 BMW X5 시리즈 차량의 전자제어장치에 적용되었으며, 현재 FlexRay 규정은 계속 수정 중에 있다.

본 논문에서는 먼저 SDL(Specification and Description Language)을 이용하여 FlexRay communication controller와 bus guardian 프로토콜 규격과 기능 부분을 설계한다.^[5-6] 설계한 SDL을 기반으로 Verilog HDL을 이용하여 하드웨어로 설계한다. 설계한 FlexRay 시스템은 Samsung 0.35 μm 공정을 이용하여 합성하였으며, 그 결과 76 MHz의 속도로 동작하는 것으로 나타났다. FlexRay 시스템의 동작을 확인하기 위해 차량에 적용되는 자동차 경보 시스템에 응용하였다. FlexRay 시스템은 ALTERA Excalibur ARM EPAX4F672C3을 이용하여 검증하였고, 자동차 경보 시스템은 ALTERA Excalibur ARM EPAX10F1020C3을 이용하여 검증하여 성공적으로 동작하는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 먼저 II장에서는 FlexRay 구조를 간략하게 살펴본다. III장에서는 SDL을 이용하여 FlexRay 시스템을 설계하고, IV장에서는 Verilog HDL을 이용하여 FlexRay 시스템을 설계하고 차량 시스템에 사용되는 자동차 경보 시스템과 연동하여 실험한다. 마지막으로 V장에서 결론을 맺는다.

II. FlexRay 시스템 구조

1. FlexRay 듀얼 버스의 배치

FlexRay 시스템은 통신채널에 여러개의 노드를 연결하여 네트워크를 구성한다. FlexRay 채널은 기본적으로 두개의 채널로 구성되며 각 노드와 노드사이의 데이터 전송은 두개의 채널을 모두 사용할 수도 있고 하나의 채널만 사용할 수도 있다. 두 채널은 보통 이중화(redundant)를 위해 사용되므로 fault-tolerant한 메시지 전송을 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.

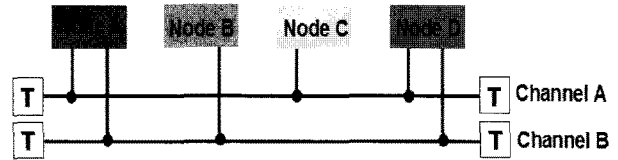


그림 1. FlexRay 듀얼 버스의 배치
Fig. 1. FlexRay dual bus topology configuration.

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.

2. FlexRay 노드 구조

FlexRay 노드 구조는 그림 2와 같이 host, communication controller(CC), bus guardian(BG), bus driver(BD)로 구성된다. Host는 사용자의 소프트웨어로서 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.

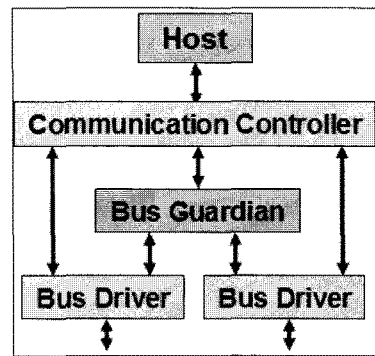


그림 2. FlexRay 노드 구조
Fig. 2. FlexRay node architecture.

3. FlexRay 타이밍 계층구조

FlexRay를 사용한 통신방식은 그림 3과 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

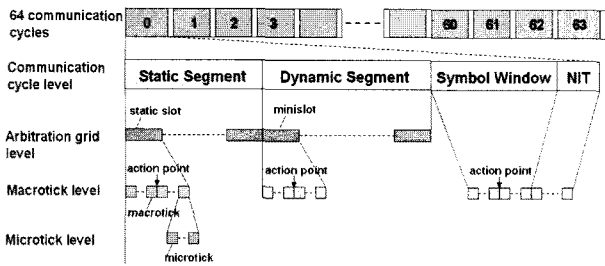


그림 3. FlexRay 타이밍 계층구조
Fig. 3. FlexRay timing hierarchy.

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 최대로 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.

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.

4. FlexRay 프레임 형식

FlexRay 프레임은 그림 4와 같이 header segment, payload segment, trailer segment로 이루어진다. 프레임의 처음 5 bits는 프레임의 기본 정보를 포함하고 있

다. Frame ID(11 bits)는 ST에서의 slot 위치를 나타내고, DYN에서는 프레임의 우선순위(낮은 ID 값이 더 높은 우선순위를 가진다)를 나타내기 위해 사용된다. Payload Length(7 bits)는 데이터의 길이(payload length × 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.

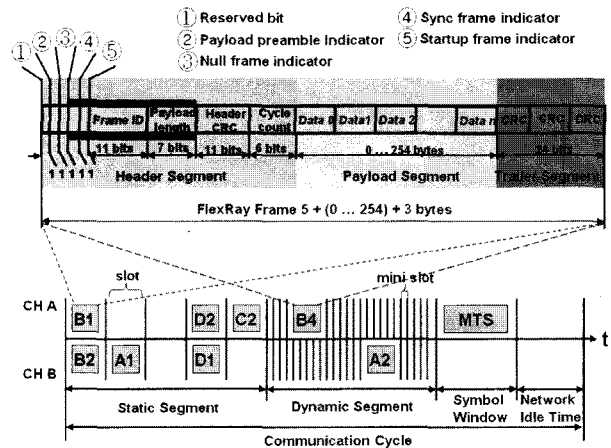


그림 4. FlexRay 프레임 형식 및 통신 사이클
Fig. 4. FlexRay frame format and communication cycle.

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.^[5] 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. 또한, 사용자의 의도 및 목적에 따라 변환 가능하고 실시간 시스템을 구현할 수 있어 주로 통신 분야에서 많이 사용되고 있다.^[6-8]

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 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.

FlexRay BG 시스템은 Bus Guardian Protocol Operation Controller (BG_POC), Communication Controller Supervision (CCS), Transmit Data Enable signal from the CC (TxEN), TxEN Supervision Process (TxENSUP_A, TxENSUP_B), Header check in Normal Supervision (NHC_A, NHC_B), TxEN active length check in Normal Supervision (NTALC_A, NTALC_B), Channel Asymmetry Detection Process (TxENASYM), Bus Guardian Enable (BGE), BGE Generation Process (BGEGEN_A, BGEGEN_B), Header Content Check Process (CHC_A, CHC_B), Startup Supervision Process (STARTUPSUP_A, STARTUPSUP_B)의 17개 블록으로 구성된다.

그림 5는 FlexRay 통신 프로토콜을 SDL로 상세 설계한 것이다. 이와 같이 구성하여 모의수행을 실행하면 실제 통신에서와 같은 환경으로 동작할 수 있어 검증과 에러 수정에 용이하다.

이와 같이 SDL을 이용하여 FlexRay 통신프로토콜을 구현하고 실행하는 과정에서의 특기사항은 다음과 같다.

i: 본 연구에서는 그림 5와 같이 MAC_A (MAC_B) 블록에서 CSP 블록으로 연결되는 채널을 추가하였고,

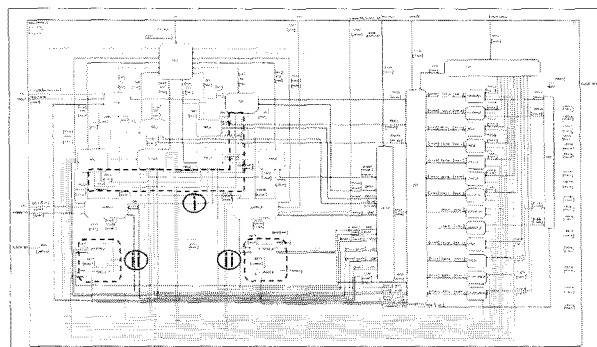


그림 5. FlexRay 시스템의 계층적 구조
Fig. 5. Hierarchical structure of FlexRay System.

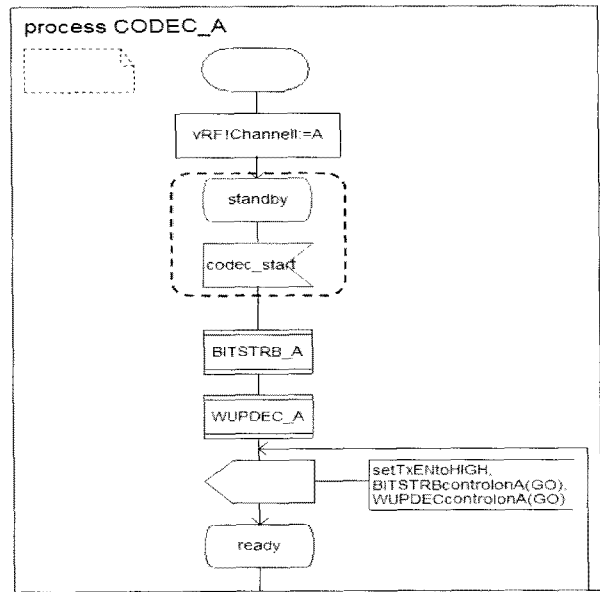


그림 6. CODEC 블록의 수정
Fig. 6. Modification of CODEC block.

여기에 actionpointonA (actionpointonB)시그널을 추가하였다.

ii: 원 스펙에서는 CODEC_A (CODEC_B) 블록 안에 BITSTRB_A (BITSTRB_B)와 WUPDEC_A (WUPEC_B) 블록이 내장되었다고 기술되어 있지만, SDL 구문 상 그림 5와 같이 따로 분리하였다.

iii: 원 스펙의 CODEC 블록에서 standby state가 없지만, SDL 구문 상 그림 6과 같이 standby state를 추가하고 codec_start의 시그널을 추가하였다. SDL을 이용하여 구현 할 때에는 codec_start 시그널은 외부에서 직접 입력 받거나 POC 블록에서 codec_start 시그널을 출력하여 CODEC 블록에서 이를 입력 받을 수 있지만, Verilog HDL을 이용하여 구현 할 때에는 standby state 와 codec_start 시그널을 따로 추가하지 않아도 된다.

iv: FlexRay 통신 프로토콜 스펙에는 크게 12개의 구현되지 않은 function이 포함되어 있다. 이러한 function은 CODEC 블록에서 prepbitstream, prepCASstream, frameCRC, headerCRC, getpayloadlength, getRF, CSP 블록에서 position, abs, length, sort, append, member 등으로 구성되어 있다. 본 연구에서는 이러한 부분을 그림 7과 같이 SDL을 이용하여 구현하였으며 MSC를 이용한 시뮬레이션을 통하여 이를 검증하여 FlexRay 시스템에 적용하였다.

v: FlexRay 통신 프로토콜에서는 클러스터 파라미터 (cluster parameters), 노드 파라미터 (node

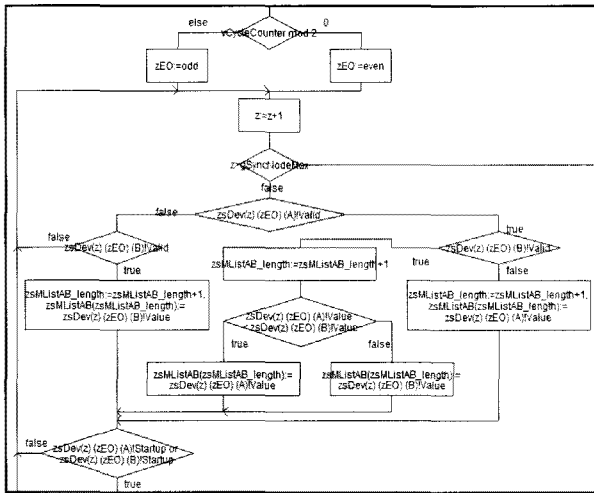


그림 7. 함수의 구현 예
Fig. 7. Example of function implementation.

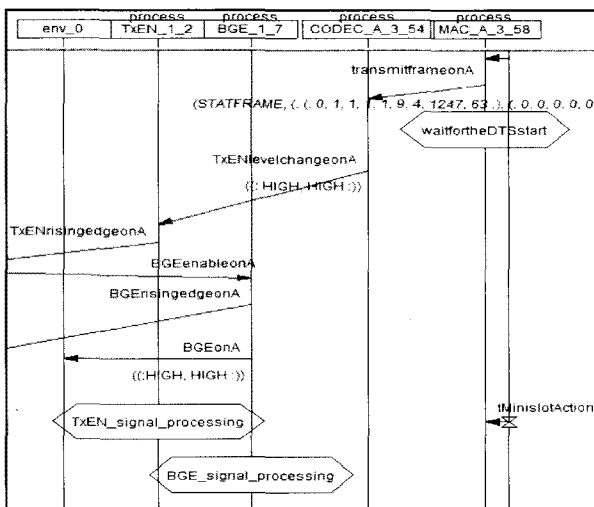


그림 8. MSC 시뮬레이션 결과의 예
Fig. 8. Example of MSC simulation result.

parameters), 프로토콜 상수 (protocol constants) 등 3 가지 종류의 파라미터가 있다. 본 연구에서는 클러스터 파라미터와 노드 파라미터의 수치는 DECOMSYS사의 FlexRay Start Kit에서 설정한 숫자를 사용하였고^[9], 프로토콜 상수는 스펙에서 설정한 상수를 사용하였다.

MSC(Message Sequence Chart)는 SDL로 설계한 FlexRay 시스템의 검증 및 시뮬레이션을 위해 사용된다. 그림 8은 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.^[10-11]

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. 합성할 결과 약 59,489 개의 게이트로 구성되며 16 mW의 파워 소비와 76 MHz의 속도에서 동작하는 것으로 나타났다. 검증을 위해 Cadence Verilog-XL과 SimVision을 이용하였다.

그림 9는 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 때만 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 시스템의 검증을 위해, 차량에서 사용되는 자동차 경보 시스템에 응용하였다.^[12] 그림 10은 자동차 경보 시스템의 알고리즘이다. 주행 중인 자동차와 전방차량과의 거리를 측정하기 위해 주행 중인 차량에 카메라를 설치하여 차선인식을 통하여 전방차량을 검출한다. 거리에 따른 도로의 폭의 변화를 측정하여 주행 중인 차량과 전방차량의 거리를 측정하여 일정한 거리 이내의 거리가 측정되면 운전자에게 알람 시스템을 통하여 경고를 주어 전방차량과의 충돌사고를 피하게 한다.

이러한 자동차 경보 시스템은 그림 11과 같이 ALTERA Excalibur ARM EPXA10F1020C3의 FPGA에서 실행되고 주행 중인 차량과 전방차량과의 거리를 측정한다.

그림 12는 테스트 환경의 블록 다이어그램을 보여준다. 주행 중인 차량에 장착된 카메라에서 출력된 사진 파일은 자동차 경보 시스템의 FPGA로 입력되어 계산된다. 여기에서 출력되는 주행 중인 차량과 전방차량 사이의 거리의 값은 FlexRay 통신 시스템의 FPGA

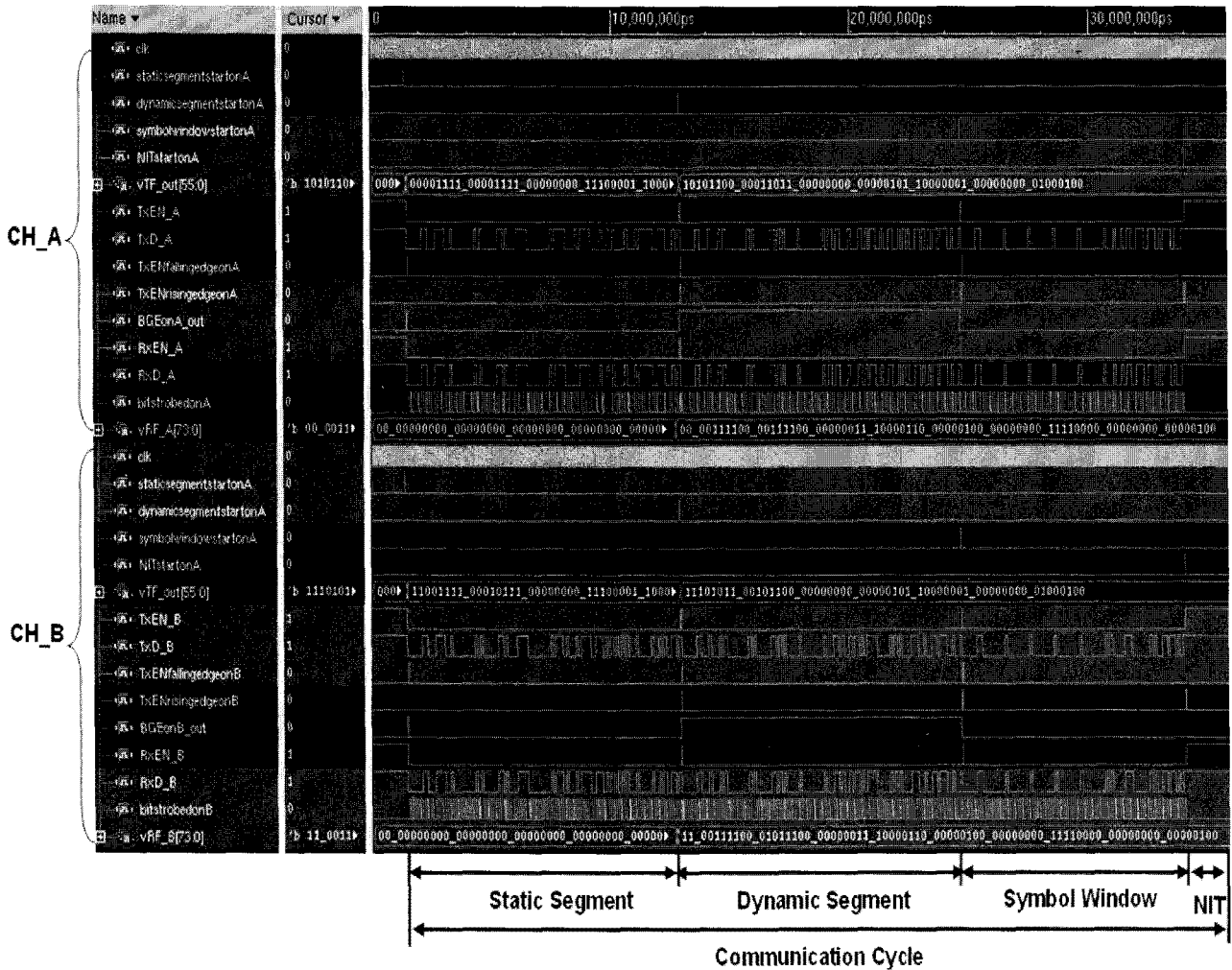


그림 9. 송신 및 수신 프레임의 시뮬레이션 결과
 Fig. 9. Simulation result of transmit and received frame.

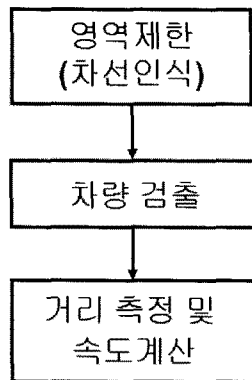


그림 10. 자동차 경보 시스템의 알고리즘
 Fig. 10. Algorithm for automobile advance alarm system.

(ALTERA Excalibur ARM EPXA4F672C3)에 입력되어 FlexRay 시스템의 host로 전송된다. 이 데이터를 이용하여 송신노드에서 전송프레임을 생성하여 신호를 전송하고, 버스를 통하여 이 신호를 수신노드에서 받아

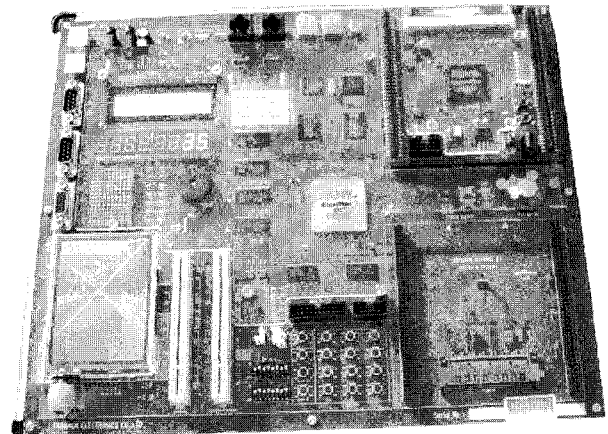


그림 11. 자동차 경보 시스템의 FPGA 구현
 Fig. 11. FPGA implementation for automobile advance alarm system.

디코딩하여 얻은 수신프레임으로부터 전송한 데이터를 확인한다. 수신된 FlexRay 수신데이터를 통하여 주행

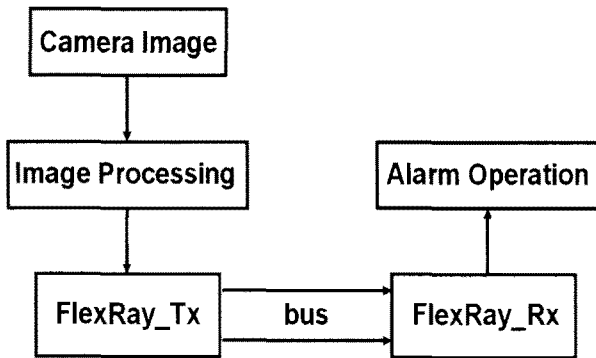


그림 12. 테스트 환경의 블록 다이어그램
Fig. 12. Block diagram of test environment.

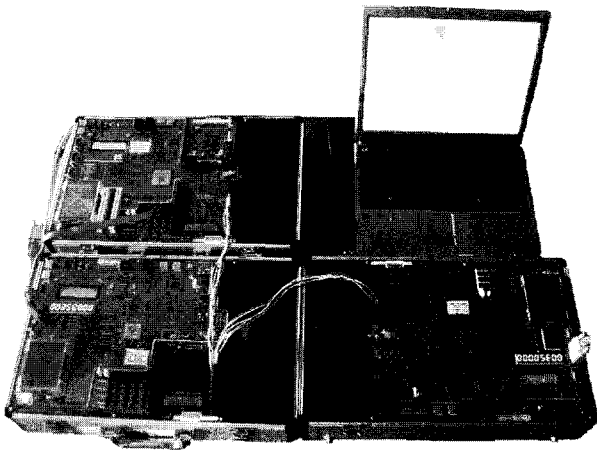


그림 13. FlexRay 네트워크의 실제 환경 테스트
Fig. 13. Actual test environment of FlexRay network.

중인 차량과 전방차량 사이의 거리가 5m 보다 작은 거리가 측정되면 알람시스템이 작동하며, 그림 13과 같이 데이터 전송이 성공적으로 이루어짐을 확인하였다.

V. 결 론

본 논문에서는 차세대 차량 내 통신네트워크에 사용될 FlexRay 통신 프로토콜의 설계 및 검증에 관한 연구를 수행하였다. FlexRay 프로토콜을 SDL을 이용하여 설계하였으며 설계한 SDL을 기반으로 Verilog HDL을 이용하여 하드웨어로 설계하였다. 설계한 FlexRay 시스템은 Synopsys Design Compiler로 Samsung 0.35 μm cell library를 이용하여 합성하였다. 합성된 시스템은 약 59,489개의 게이트로 구성되며, 동작속도 76 MHz에서 16 mW의 전력을 소모한다.

설계한 FlexRay 시스템의 검증을 위해, 차량에서 사용되는 자동차 경보 시스템에 응용하였다. 자동차 경보 시스템은 ALTERA Excalibur ARM EPXA10F1020C3 FPGA를 사용하여 설계하였다. 차량에 장착된 카메라

에서 받은 이미지는 자동차 경보 시스템의 FPGA에서 차선인식, 차량검출, 차량거리 등 계산을 하고, 차량거리 데이터는 설계된 FlexRay FPGA (ALTERA Excalibur ARM EPXA4F672C3)로 전송된다. 수신된 데이터를 처리하여 주행 중 차량과 전방차량과의 거리가 5m 이내의 거리가 측정되면 알람시스템이 작동되는 것을 확인하여 설계한 FlexRay 시스템이 성공적으로 동작함을 검증하였다.

참 고 문 헌

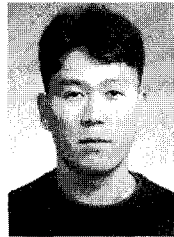
- [1] G. Leen and D. Hefferman, "Digital networks in the automotive vehicle," *IEEE Computer and Control Engineering Journal*, vol. 10, no. 6, pp. 257-266, Dec. 1999.
- [2] FlexRay Communications System Protocol Specification v2.1 Revision A, 2005.
- [3] Dominique Paret. *Multiplexed Networks for Embedded Systems*. John Wiley & Sons, Ltd, 2007.
- [4] FlexRay Consortium, <http://www.flexray.com>.
- [5] ITU-T Z.100, CCITT Specification and Description Language (SDL), 1996.
- [6] Homepage of the SDL-Forum, <http://www.sdl-forum.org>.
- [7] Ferenc Belina, Dieter Hogrefe and Amardeo Sarma. *SDL with Applications from Protocol Specification*. Prentice Hall, 1991.
- [8] Laurent Doldi. *Validation of Communications Systems with SDL*. John Wiley, 2003.
- [9] DECOMSYS GmbH, <http://www.decomsys.com>.
- [10] Michael D. Ciletti. *Advanced Digital Design with the Verilog HDL*. Prentice Hall, 2002.
- [11] J. Bhasker. *A Verilog HDL Primer*. Star Galaxy Publishing, 2005.
- [12] J. Y. Yi, S. H. Yang, I. G. Jang, J. G. Chung, and M. S. Lim, "Automobile advance alarm system based on monocular vision processing," *2007 IEEE Intelligent Vehicles Symposium*, pp. 428-432, Istanbul, Turkey, June. 2007.

저 자 소 개



허 일 남(학생회원)
 2003년 중국 연변대학교 전자
 정보공학과 학사 졸업
 2005년 전북대학교 물리기술학과
 석사 졸업
 2005년~현재 전북대학교
 전자정보공학과 박사

<주관심분야 : 통신, 신호처리, 반도체>



양 상 훈(학생회원)
 2005년 전북대학교 전자공학과
 학사 졸업
 2007년 전북대학교 정보통신
 공학과 석사 졸업
 2008년~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>