

論文2002-39SD-3-10

# DMA 인터페이스를 갖는 블루투스 기저대역 모듈의 설계 및 구현

## (Design and Implementation of a Bluetooth Baseband Module with DMA Interface)

千翊宰\*, 吳從煥\*, 林智淑\*, 金備鎬\*\*, 朴仁哲\*\*\*

(Ik Jae Chun, Jong Hwan Oh, Ji Suk Lim, Bo Gwan Kim, and In Cheol Park)

### 요약

블루투스 무선 기술은 음성 및 데이터 전송을 위한 단거리 일대다중 무선 주파수 통신을 위해 제안된 범용적으로 사용 가능한 무선통신 기술이다. 블루투스는 2.4GHz ISM 밴드에서 동작하며 약 10m 범위의 다양한 이동 장치와 휴대용 장치를 위한 저가격의 기저대역 무선 접속을 제공한다. 본 논문은 DMA 방식의 블루투스 기저대역 모듈을 개발하고 그 구조와 테스트 결과를 보인다. 개발된 모듈은 링크 컨트롤러, UART 그리고 오디오 코덱의 세가지 블록으로 구성되며 메인 프로세서 사이의 정보 전달 및 DMA 지원을 위한 버스 인터페이스와 RF 모듈과의 데이터 송수신을 위한 RF 인터페이스를 지원한다. DMA의 사용은 FIFO를 이용한 데이터의 송수신 방법을 사용하는 기저대역 모듈에 비하여 모듈의 구현 크기 및 데이터의 처리 속도에 있어서도 많은 차이점을 갖는다. 각 블록을 DMA를 지원하도록 설계함으로써 작은 크기의 모듈을 설계할 수 있다. 이러한 작은 크기의 모듈은 생산 비용의 절감과 함께 다양한 응용분야에 사용될 수 있는 범용성을 제공한다. 또한 본 모듈은 UART를 이용한 펌웨어 업그레이드 방식을 지원하고 소프트 IP로 설계되었으며 FPGA와 ASIC으로 구현하여 개인용 컴퓨터 사이의 파일 전송과 비트-스트림 전송을 통해 테스트 되었다.

### Abstract

Bluetooth technology is a publicly available specification proposed for Radio Frequency (RF) communication for short-range and point-to-multipoint voice and data transfer. It operates in the 2.4GHz ISM(Industrial, Scientific and Medical) band and offers the potential for low-cost, broadband wireless access for various mobile and portable devices at range of about 10 meters. In this paper, we describe the structure and the test results of the bluetooth baseband module with direct memory access method we have developed. This module consists of three blocks; link controller, UART interface, and audio CODEC. This module has a bus interface for data communication between this module and main processor and a RF interface for the transmission of bit-stream between this module and RF module. The bus interface includes DMA interface. Compared with the link controller with FIFOs, The module with DMA has a wide difference in size of module and speed of data processing. The small size module supplies low cost and various applications. In addition, this supports a firmware upgrade capability through UART. An FPGA and an ASIC implementation of this module, designed as soft IP, are tested for file and bit-stream transfers between PCs.

**Key words** : Bluetooth, baseband, link controller.

\* 學生會員, \*\* 正會員, 忠南大學校 電子工學科  
(Dept. of Electronics Eng., Chungnam National Univ.)  
\*\*\* 正會員, 韓國科學技術院 電子電算學科  
(Dept. of Electrical Eng. and Computer Science, KAIST)

※ 본 연구는 KAIST MICROS 센터를 통한 한국과학기술  
재단의 우수연구센터 지원금과 반도체 설계 교육  
센터(IDEA) 그리고 BK21 충남대학교 정보통신 인  
력 양성사업단의 지원에 의하여 수행되었습니다.  
接受日字:2001年12月3日, 수정완료일:2002年1月17日

## I. 서 론

사회적인 요구와 관련 기술 및 테크놀러지의 발달에 힘입어 무선 통신기술은 전화 서비스, 의료 기기, 가전 제품뿐만 아니라 많은 분야에 적용되고 있다. 기존에 유선으로 그 기능을 충분히 하던 분야들도 새로운 기능을 추가하고 사용자에게 더 많은 편의를 제공하기 위하여 무선 기술을 도입하고 있다. 특히, ISM 대역폭을 사용하는 무선 대역 확산 통신은 주파수 대역을 무료로 사용할 수 있기 때문에 이 대역폭을 사용하는 무선 통신 방식을 채용하는 응용 분야는 폭발적으로 늘어날 것이다. 이처럼 무선통신 기술이 여러 분야에 적용되는 상황에서 차세대 무선통신의 핵심이 될 블루투스의 기저대역 모듈의 구현은 중요한 과제라 할 수 있다.

블루투스는 현재 세계적으로 하드웨어와 소프트웨어의 초기 단계 개발이 이루어지고 있는 ISM 대역을 이용한 단거리 무선통신 기술 규격으로써 이동 장치와 휴대용 장치를 위한 저 가격 무선통신 네트워크 구축을 목적으로 개발되었다. 블루투스 규격은 Ericsson, IBM, Intel, Nokia, Toshiba가 주축이 되어 시작된 Bluetooth Special Interest Group(SIG)에 의해 개발되었으며 현재 3COM, Lucent, Microsoft 등과 같은 텔레커뮤니케이션, 컴퓨터, 네트워크 분야의 대표적인 기업을 포함해 1800여 개의 회사가 참여하고 있다<sup>[1]</sup>.

본 논문에서는 블루투스 물리 계층의 프로토콜 처리 부분인 링크 컨트롤러와 UART, 오디오 코덱을 통합한 블루투스 기저대역 모듈을 DMA를 이용하여 데이터를 입출력 할 수 있도록 함으로써 작은 크기의 모듈을 설계하였고 비트 스트림 처리에 대해서 블루투스 스펙 1.0b를 기준으로 설계하고 검증하였다.

서론에 이어서, II장에서 블루투스의 기본 특징과 모듈의 구성 및 구조에 대하여 간략히 알아보고 III장에서 DMA를 사용한 기저대역 모듈의 설계 시 고려 사항과 설계 및 구현 내용에 대하여 설명하고 IV장에서 구현 및 실험 결과를 소개한 후 V장에서 결론을 맺도록 한다.

## II. 블루투스의 특징 및 구조

### 1. 블루투스의 기본 특징

블루투스는 휴대용 기기나 데스크 탑 전자 장치들을 연결하는 케이블을 대체하기 위한 단거리 무선 통신으로써 제안되었으며 낮은 복잡도, 저 전력, 저 가격화에 중점을 두어 개발되고 있다.

블루투스 시스템은 점대점 접속과 점대다중 접속을 지원한다. 점대다중 접속에 있어서 채널은 블루투스 장치들에 의해서 공유되며 이러한 장치들은 피코넷을 형성한다. 이렇게 형성된 하나의 피코넷은 하나의 마스터와 최대 7개의 슬레이브를 가질 수 있으며 모든 장치들은 마스터 장비를 기준으로 하는 클럭을 공유하게 된다. 스캐터넷은 다른 피코넷과의 연결을 이루는 피코넷의 그룹을 정의한다<sup>[2]</sup>.

블루투스는 2.4GHz 대역의 unlicensed ISM(Industrial, Scientific, Medical) 밴드에서 동작하며 간섭(interference)과 감쇠(fading)를 줄이기 위하여 주파수 홉 송수신기를 적용하고 있다. 주파수 호핑(hopping)의 순서는 블루투스의 마스터 클럭에 의해서 결정된다. 또한 전이중 전송을 수행하기 위하여 TDD(Time-Division Duplex) 방식을 사용하고 있다<sup>[2]</sup>.

블루투스 프로토콜 스택은 데이터 전송을 위한 ACL(asynchronous connection-less) 링크와 음성 전송을 위한 SCO(synchronous connection-oriented) 링크, 그리고 데이터와 음성을 모두 전송할 수 있는 링크를 지원한다. 데이터 전송을 위한 비트 레이트(bit-rate)는 1Mb/s이며 하나의 채널 슬롯은 625us의 슬롯 길이를 가진다. 각각의 음성 채널(voice channel)은 64Kbps 동기식 채널(synchronous channel)을 지원하며, 비동기 링크(asynchronous link)를 이용한 최대 전송율은 723.2Kbps이다<sup>[2~5]</sup>.

### 2. 블루투스 모듈의 구성

일반적으로 블루투스 모듈은 크게 무선 데이터 전송을 위한 무선 채널을 생성하는 RF 모듈과 링크를 관리하고 비트 스트림 처리를 수행하는 기저대역 모듈의 두 부분으로 나누어진다. 기저대역 모듈에 있어서 링크 컨트롤러는 기본적으로 기저대역 프로토콜과 저수준 연결 처리과정을 수행하며 그림1에서 보이는 것과 같이 기저대역 모듈을 구성하는 주요 블록 중에서 베이스밴드 유닛을 포함한다. 베이스밴드 유닛은 블루투스 비트 스트림의 인코딩 및 디코딩과 저수준 타이밍 제어를 통한 비트 레벨 처리를 수행하는 부분으로써 링크 컨트롤러에 대한 하위레벨 기능을 수행한다<sup>[3]</sup>. 프

로세서(MCU)는 링크 매니저 기능을 수행하며 각 페리퍼럴들을 제어한다. UART는 호스트 컨트롤러와 기저대역 모듈사이의 인터페이스를 담당하는 부분으로 직렬 통신 방식을 제공하며 PCM CODEC은 A-law,  $\mu$ -law 및 CVSD 방식의 음성신호 처리를 담당한다. 그림 1은 일반적인 블루투스 모듈의 구성을 보인다.

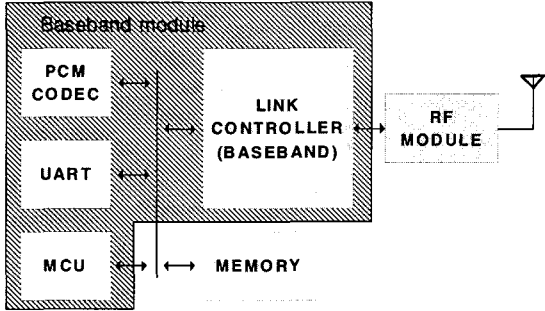


그림 1. 블루투스 모듈의 구성  
Fig. 1. Architecture of Bluetooth module.

3. 블루투스 기저대역 프로토콜의 기본 구조

블루투스의 기저대역 프로토콜은 일반적으로 라디오(radio), 베이스밴드(baseband), 링크 컨트롤러(link controller), 그리고 링크 매니저(link manager)와 호스트 터미널 인터페이스를 위한 호스트 컨트롤러 인터페이스(Host Controller Interface)로 구성된다. 링크 매니저는 호스트 컨트롤러 인터페이스(HCI)로부터 받은 명령어 및 데이터를 베이스밴드 수준의 동작으로 변환시켜 주며 다른 블루투스 장치와 링크를 만들고 제어하는 역할을 한다. 이 부분은 프로세서가 담당하는 부분으로써 대부분의 기능은 소프트웨어에 의해 수행된다. 링크 컨트롤러는 링크 매니저로부터 명령어 및 데이터를 받아 실제적으로 베이스밴드가 동작을 할 수 있도록 물리적인 링크(physical link)를 제어하며 패킷들을 재결합하는 역할을 수행한다. 베이스밴드는 데이터를 실제적으로 인코딩하고 디코딩하며 데이터 전송 타이밍을 제어하는 부분으로 링크 컨트롤러로 구분되기도 한다[3],[4]. 라디오는 무선 송수신을 위하여 데이터를 GFSK(Gaussian Frequency Shift Keying)로 변복조하며 주파수 호핑(Frequency hopping)을 수행한다. 본 논문에서 설계한 링크 컨트롤러는 베이스밴드 기능을 포함한다. 그림 2는 블루투스 시스템의 기저대역 프로토콜 스택을 보인다[3].

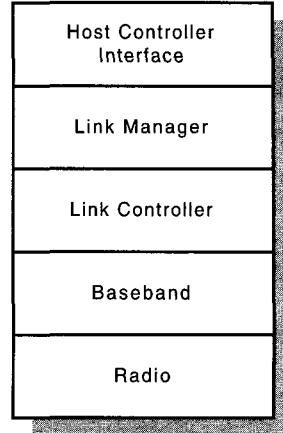


그림 2. 기저대역 프로토콜 스택  
Fig. 2. Baseband protocol stack.

III. 기저대역 모듈의 설계 및 구현

1. 기저대역 모듈의 구현 시나리오

본 논문에서는 기저대역 모듈을 다음의 그림 3에서 보이는 시나리오를 기반으로 구현하고 테스트 하였다. 기저대역 모듈은 링크 컨트롤러, UART, 그리고 오디오 코덱의 세 부분으로 구성되며 이 것은 프로세서에 대하여 각각 독립적인 페리퍼럴로써 동작한다. 프로세서와 페리퍼럴들 사이의 인터페이스는 ARM프로세서의 인터페이스를 기준으로 하였으며 프로세서는 레지스터 설정을 통하여 각각의 페리퍼럴 유닛들 즉, 링크 컨트롤러, UART, 그리고 오디오 코덱을 제어할 수 있다. 효율적인 데이터 처리를 위하여 모든 데이터는 DMA를 이용하여 처리하도록 하였다. 이를 위하여 메모리 관리 유닛(MMU)을 두어 프로세서, 메모리, 그리고 각 페리퍼럴들 사이의 정보 전달을 수행하고 제어한다. 이와 같이 DMA를 사용하여 UART 및 오디오 코덱으로부터 입력된 음성 데이터를 메모리로부터 링크 컨트롤러가 직접 받아들임으로써 각 페리퍼럴들에서 중복 사용된 버퍼를 없앨 수 있다. 그 결과로 버퍼로 인한 모듈의 크기를 줄일 수 있으며 데이터 처리에 있어서의 유연성을 부여할 수 있다는 장점을 가진다. 각각의 페리퍼럴들 사이의 데이터 송, 수신은 기본 단위는 1바이트이며 이것은 베이스밴드의 데이터 패킷 처리 정보의 기본 단위가 1바이트를 기준으로 하기 때문이다. 그리고 각 페리퍼럴을 제어하는 레지스터 설정을 위한 버스 인터페이스는 2바이트 데이터 폭을 기준으로 설계

스 인터페이스는 2바이트 데이터 폭을 기준으로 설계하였다.

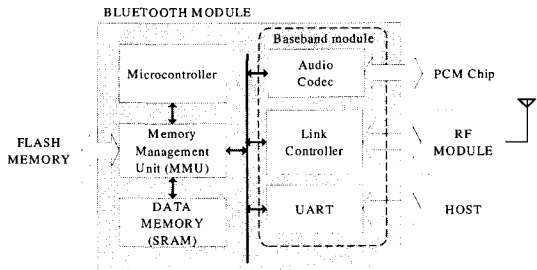


그림 3. 기저대역 모듈 블록 다이어그램  
Fig. 3. Block diagram of the Bluetooth baseband module.

ARM 프로세서의 명령어 패치 및 프로그램 데이터는 16비트 인터페이스를 통하여 플래시 메모리에서 가져온다. 만약 32비트가 필요한 경우 'nWAIT' 신호를 이용하여 두 사이클에 데이터를 가져온다. 일반적으로 페리퍼럴은 프로세서에 비해서 데이터의 부하가 작기 때문에 프로세서에 우선권을 주더라도 페리퍼럴들의 데이터 처리에 있어서의 처리 지연 및 송수신 오류는 발생하지 않는다. 프로세서의 인터럽트는 모든 페리퍼럴에 대하여 우선한다.

메모리 관리 유닛을 통한 데이터 처리의 내용은 다음과 같다.

1) 플래시 메모리 -> 프로세서

다른 오퍼레이션과 독립적으로 수행되며 프로세서가 인스트럭션을 가져올 때 사용된다.

2) DATA RAM <-> 프로세서

프로세서가 데이터 램에 read/write 동작을 하는 경우로 모두 한 사이클에 처리된다.

3) 프로세서 <-> 페리퍼럴

프로세서가 MMU와 페리퍼럴들의 레지스터에서 데이터를 읽고 쓸때의 동작으로써 페리퍼럴들 내부의 레지스터를 설정함으로써 각 페리퍼럴들을 제어할 수 있다.

4) 페리퍼럴 -> 데이터 램

각 페리퍼럴은 데이터를 메모리에 써야하는 상황이 되면 메모리 관리 유닛에 리퀘스트를 요청하며 해당 페리퍼럴은 각 각의 이벤트에 해당하는 리퀘스트를 요청하게 된다. 메모리 관리 유닛은 요구된 리퀘스트의 처리가 가능하면 다음 사이클에 'ack' 신호를 보낸다.

페리퍼럴은 이 'ack' 신호가 온 클럭 사이클에 데이터를 메모리 관리 유닛으로 보내고 메모리 관리 유닛은 같은 사이클에 해당 메모리 번지에 데이터를 저장한다. 이때 모든 데이터 전송의 기본 단위는 1바이트이다.

5) 데이터 램 -> 페리퍼럴

프로세서는 데이터 처리를 원하는 대상 페리퍼럴에 수신해야 하는 데이터의 크기를 알려주고 페리퍼럴은 설정된 데이터의 크기 만큼의 해당 리퀘스트 신호를 보내고 'ack' 신호시에 데이터를 받는다. 이때 데이터 전송의 기본단위는 1바이트이다.

2. 기저대역 모듈 인터페이스

기저대역 모듈의 버스 인터페이스는 DMA 인터페이스를 기반으로 설계하였다. 각각의 페리퍼럴들이 DMA 인터페이스를 갖도록 설계 함으로써 각 모듈의 자체 FIFO 블록 사용을 줄였다. 현재 상용으로 제공되는 기저대역 모듈 IP의 경우 인터페이스 블록간에 자체 FIFO를 가지고 있으며 데이터 처리를 위하여 외부 SRAM의 사용을 지원하고 있다. 이러한 구조는 패킷의 송수신 데이터 처리를 FIFO를 이용하여 처리하고 프로세서와 외부 SRAM을 이용해 데이터의 흐름을 제어하는 구조를 갖는다. 그러나 본 논문에서 설계한 모듈은 링크 컨트롤러와 UART 및 오디오 코덱을 거쳐 송수신 되는 데이터를 DMA 인터페이스를 통하여 메모리에 저장하고 읽도록 함으로써 바로 UART 인터페이스나 오디오 코덱을 통하여 패킷을 전달할 수 있다는 장점을 갖는다. 이와 같이 기저대역 모듈의 인터페이스를 DMA를 사용할 수 있도록 설계함으로써 본 IP와 메모리 관리 유닛(MMU), SRAM을 포함하여 하나의 칩으로 구현 할 경우 상용 링크 컨트롤러에 비하여 블루투스 모듈의 크기를 줄일 수 있으며 비용절감 효과를 기대할 수 있다는 장점을 갖는다. 이와 같이 링크 컨트롤러, UART 그리고 오디오 코덱을 모두 DMA 기능을 갖도록 설계하여 각 블록별 FIFO의 사용을 없애므로써 작은 크기의 기저대역 모듈을 구현하였다.

블루투스 기저대역 모듈을 구성하는 링크 컨트롤러와 UART, 오디오 코덱 페리퍼럴과 메모리 사이의 데이터 송수신 인터페이스는 메모리 관리 유닛을 통하여 이루어지며 메모리 관리 유닛 사이에는 그림 4와 그림 5의 타이밍을 가지는 버스 인터페이스를 이용해 데이터와 정보를 주고 받는다. 그림 4는 링크 컨트롤러가 메모리 관리 유닛을 통하여 데이터를 읽고 쓰는 타이

지스터 설정 타이밍을 보인다.

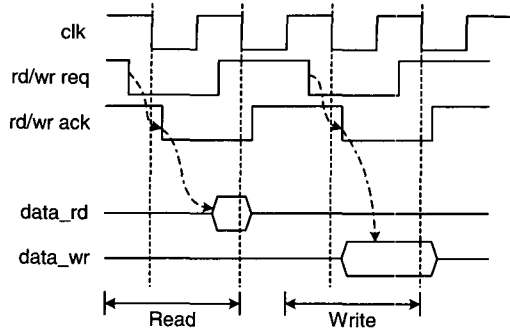


그림 4. 링크 컨트롤러와 MMU의 타이밍 다이어그램  
Fig. 4. Timing diagram between link controller and MMU.

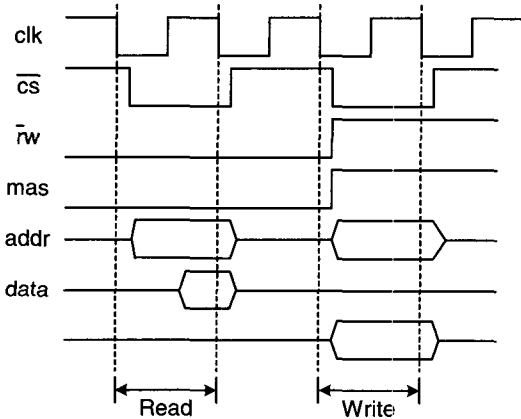


그림 5. 링크 컨트롤러 레지스터 설정을 위한 타이밍 다이어그램  
Fig. 5. Timing diagram for register setting of link controller.

3. 링크 컨트롤러의 기능 결정

링크 매니저와 링크 컨트롤러 간의 처리 과정에 있어서 소프트웨어와 하드웨어의 트레이드 오프는 설계 시 중요한 부분으로서 나타난다. 링크 컨트롤러에 의한 하드웨어적인 물리적 링크의 제어는 좀더 빠른 처리 속도와 더불어 프로세서의 부하를 줄일 수 있다는 장점이 있으나 하드웨어의 크기 증가로 인한 비용증가 및 유연성이 떨어짐을 단점으로 들 수 있다. 반면 프로세서가 물리적 링크 생성에 관한 여러 신호들을 모두 제어하는 방법을 사용한다면 프로토콜 처리에 있어서의 유연성은 확보할 수 있으나 많은 부하가 프로세서에 걸림으로써 발생하는 속도 저하 및 병목 현상을 단

점으로 들 수 있다.

본 논문에서 설계한 링크 컨트롤러는 빠른 시간에 블루투스 기능을 칩에 구현하여 프로세서와 RF 모듈사이의 인터페이스를 구현 할 수 있도록 하는데 중점을 두었으며 이를 위하여 소프트웨어 또는 하드웨어를 통하여 베이스밴드 기능을 제어할 수 있도록 하였다. 프로세서의 병목 현상을 방지하기 위하여 레지스터의 설정을 통해서 RF 모듈과의 데이터 송수신에 관련된 신호들을 하드웨어적인 방법으로 지정된 타이밍에서 발생시킬 수 있으며 또한 프로세서에 의한 제어를 통해서 관련 신호들을 제어 할 수 있도록 함으로써 유연성을 부여하였다. 그리고 프로세서의 타이밍 제어를 원활히 할 수 있도록 내부 타이머를 설계하여 내부 타이머의 설정에 의해서 링크 컨트롤러를 제어 할 수 있도록 하였다. 링크 컨트롤러는 데이터 송수신에 따른 다양한 인터럽트를 생성할 수 있도록 설계 함으로써 각 상황에 맞는 다양한 인터럽트를 프로세서에 제공할 수 있으며 이 인터럽트 정보를 이용해 프로세서는 링크 컨트롤러의 상황을 좀 더 빨리 확인하고 제어 할 수 있도록 하였다.

4. 링크 컨트롤러 모듈의 구조

링크 컨트롤러는 하위레벨 프로토콜을 처리하는 부분으로써 링크 연결(link connection)에 관한 정확한 정보(TX/RX 타이밍, 인터럽트 타이밍, 이벤트 상황 등)를 링크 매니저에 전달하고 링크 매니저의 제어에 따라 프로토콜을 처리하며 데이터 송수신을 위한 신호를 자동으로 생성 제어하는 기능을 갖는다. 그림 6은 블루투스 링크 컨트롤러 모듈의 블록 다이어그램을 보인다.

설계한 링크 컨트롤러의 특징은 다음과 같다.

- 인코딩/디코딩, 비트 스트림 처리
- 시스템 클록 : 12MHz
- 보조 클록 : 3.2KHz, 1MHz, 4MHz
- 비트 스트림 처리
  - Tx 동기 : RF 모듈로부터 생성되는 1MHz 클록을 이용
  - Rx 동기 :
    - ◆ PLL을 이용하여 수신된 데이터 스트림으로부터 1MHz 클록을 추출하여 사용 가능
    - ◆ RF 모듈로부터 동기 신호를 수신하여 사용 가능
- RF 인터페이스 : 에릭슨 PBA313 칩 호환

- RF 인터페이스 : 에릭슨 PBA313 칩 호환
- 4 공통 패킷 / 4 SCO 패킷 / 6 ACL 패킷 지원
- 블루투스 클럭 제어 및 내부 하드웨어 타이머
- 액세스 코드 생성 및 검출
- Forward Error Correction (FEC)
- Header Error Check (HEC)
- Cyclic Redundancy Generation and Check (CRC)
- Encryption/Decryption
- DMA를 사용한 데이터 처리
- 송수신 흐름 제어를 위한 컨트롤 로직

링크 컨트롤러의 모든 기능은 레지스터에 의해 제어되므로 프로세서가 링크 컨트롤러의 LM/LC 컨트롤 레지스터 즉, 링크 매니저 관련 레지스터와 링크 컨트롤러 관련 레지스터를 설정함으로써 비트-스트림 처리와 인터럽트, 암호화 등과 같은 기능을 제어할 수 있다. 데이터 전송에 따른 모든 패킷의 상태는 인터럽트 신호(nIRQ)에 의해서 컨트롤러를 호출하고 인터럽트 레지스터에 의해 컨트롤러로 전달된다.

시스템 클럭은 12MHz로 동작되며, 그 이외에 데이터 송수신과 RF 모듈과의 직렬 통신을 위해서 3.2KHz, 1MHz, 4MHz의 3가지 클럭이 사용된다. 1MHz는 데이터 송신 시에 데이터 동기를 위해 사용된다. 비트 스트림 처리에 있어서 송신의 경우 RF 모듈에서 제공하는 1MHz의 클럭에 동기되고 수신은 라디오 인터페이스(radio interface) 부의 PLL블록에서 생성된 1MHz 클럭에 동기된다. 3.2KHz는 블루투스의 native clock으로 사용되며 송수신 타이밍과 블루투스 장치간 동기화를 위해 사용된다. 4MHz는 링크 컨트롤러와 RF모듈간의 직렬 통신을 위해 사용되는 클럭이다.

송수신 버퍼(Tx/Rx buffer) 블록은 메모리 관리 유닛을 통해 메모리로부터 받아들이는 데이터를 더블 버퍼링(double buffering)을 이용하여 비트 스트림을 처리하는 데이터 패스 처리 블록으로 전달하는 역할을 수행한다.

설계된 모듈은 블루투스 클럭 제어를 위하여 클럭 옵셋을 제어하는 'CLK Offset Control logic'을 가지고 있다. 이 블록은 블루투스 스펙에서 정의하는 CLK, CLKN, CLKE의 세 가지 클럭을 제어하는 역할을 수행한다. 블루투스의 클럭은 호핑 신호를 생성하는 'HOP Selection logic'의 79-hop system에서 호핑 순서(hopping sequence)를 생성하는 블록의 입력으로 사용

됨으로써 클럭의 변화에 따라 주파수 변환을 수행한다.

패킷의 수신 시에 패킷을 정확히 인식하고 정보를 추출하기 위하여 RF 인터페이스 블록에 PLL 로직을 포함하였으며, 패킷의 액세스 코드(access code) 내의 싱크 워드(sync word)를 이용한 동기화를 위하여 코릴레이터(correlator)를 사용하고 있다. 처리된 데이터는 RF 인터페이스를 통하여 링크 컨트롤러로 수신된다. RF 인터페이스는 바운드리 스캔 방식의 직렬 인터페이스를 이용하여 RF모듈을 제어한다.

비트 스트림의 송수신 과정 수행 중에 발생하는 송수신 정보는 인터럽트에 의해서 프로세서에 전달된다. 인터럽트가 발생하는 대부분의 경우는 패킷의 송수신과 관련되어 발생한다. 이때 발생하는 인터럽트 상태는 인터럽트 상태 레지스터(Interrupt Status Register)에 저장되며 처리된 인터럽트는 인터럽트 클리어 레지스터(Interrupt Clear Register)에 의해 해제된다. 블루투스 클럭 제어를 위해 사용되는 'Compare timer'에 의해 발생하는 이벤트뿐만 아니라 송수신에 관한 모든 이벤트는 링크 컨트롤러의 시스템 인터럽트를 활성화시킬 수 있다. 각 인터럽트에 관련된 이벤트는 인터럽트 상태 레지스터의 대응되는 비트에 래치(latch)되고 이벤트가 발생할 때 활성화 된다. 이러한 이벤트들은 펌웨어(firmware)에 의해 설정된 인터럽트 마스크 레지스터(Interrupt Mask Register)에 대응되는 비트에 대해서만 발생한다.

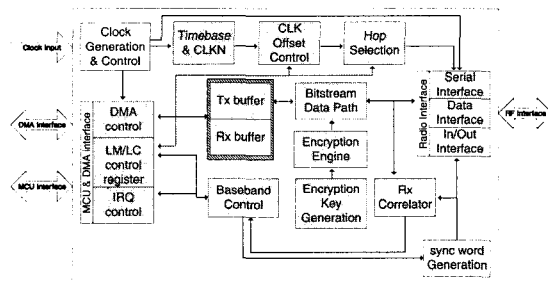


그림 6. 링크 컨트롤러 모듈의 블록 다이어그램  
Fig. 6. Block diagram of link controller module.

5. 링크 컨트롤러의 인터페이스

그림 7은 링크 컨트롤러와 메모리 관리 유닛 그리고 RF 모듈 사이의 버스 인터페이스와 RF 인터페이스를 보인다. 링크 컨트롤러와 프로세서 그리고 링크 컨트롤러와 메모리 사이의 인터페이스는 메모리 관리 유닛을

그림 8은 링크 컨트롤러와 RF 모듈 사이의 타이밍 다이어그램을 보인다. RF 모듈과의 인터페이스는 에릭슨의 RF 모듈을 기준으로 하여 설계하였으며 RF 모듈의 제어는 바운드리 스캔 구조(Boundary Scan Architecture(IEEE std 1149.1))에 기반한 시리얼 인터페이스에 의해 이루어진다<sup>[6]</sup>. 인터페이스 신호는 링크 컨트롤러 레지스터의 설정에 따라서 관련 신호의 발생을 제어할 수 있으며 프로세서의 직접적인 제어를 통하여 설정 될 수 있도록 하였다. 수신 관련 신호들의 발생 타이밍은 그림에서 보듯이 'comptimer', 'PU\_RX/TX', 'RX\_ON\_set'의 레지스터를 설정함으로써 하드웨어 또는 소프트웨어 수준에서 제어 할 수 있다. 송신 관련 신호의 경우도 수신 관련 신호와 같은 방법으로 제어 한다.

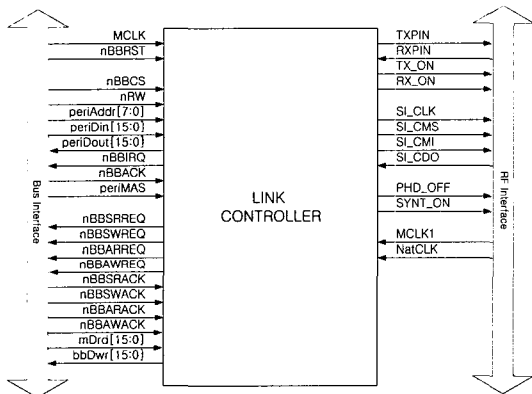


그림 7. 링크 컨트롤러의 인터페이스  
Fig. 7. Link controller interface.

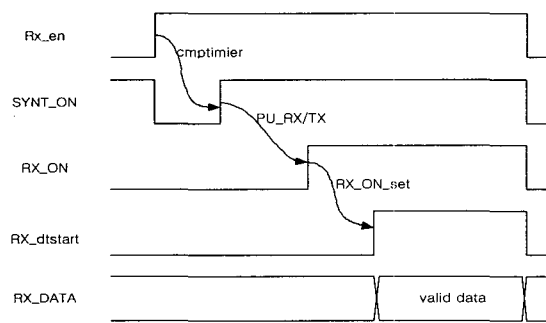


그림 8. RF 모듈의 타이밍 다이어그램  
Fig. 8. Timing diagram for RF module.

6. 블루투스 클럭 제어

클럭 제어 블록은 블루투스 유닛의 설계에 있어서 가장 중요한 부분으로 두 블루투스 유닛이 서로 통신

채널을 형성하고 통신을 수행하기 위해서는 마스터 클럭에 기준을 두고 슬레이브의 클럭을 동기 시켜야 한다. 이를 위해 링크 컨트롤러 모듈은 3.2KHz 클럭을 사용하여 송수신기의 송수신 시점을 결정하는 기준 클럭으로 이용하며 이를 28 비트 카운터로 구현한다. 블루투스는 CLKN, CLKE, CLK의 3가지 클럭을 정의하고 사용한다<sup>[2,3,4]</sup>. CLKN은 수정되지 않는 클럭으로써 모든 클럭의 기준이 되고 CLKE와 CLK는 각각 추정된 클럭과 마스터 클럭을 나타내며 CLKN에 옵셋을 더함으로써 얻어진다. 피코넷에서 채널의 타이밍과 주파수 호핑은 마스터 장치의 클럭에 의해 결정되므로 이 과정이 제대로 수행되지 않으면 타이밍뿐만 아니라 주파수 호핑에 까지 영향을 주게 되므로 송수신에 있어서 가장 기본적이고 중요한 부분이 된다. 그림 9는 블루투스 클럭 생성을 위한 블록이며 그림 10은 클럭의 페이즈 값과 마스터 클럭 값에 의한 클럭 보정 과정을 보여주는 타이밍 다이어그램이다.

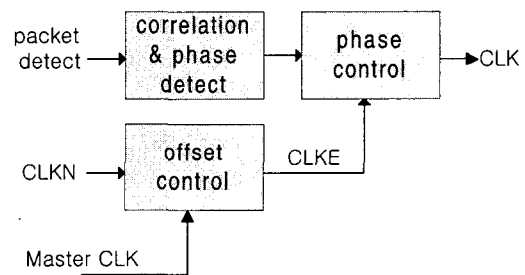


그림 9. 블루투스 클럭 생성  
Fig. 9. Block diagram for Bluetooth clock generation.

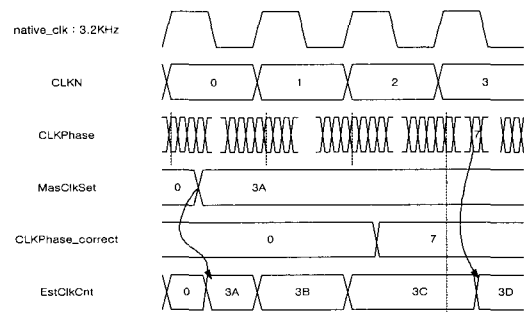


그림 10. 클럭 제어 카운터의 타이밍 다이어그램  
Fig. 10. Timing diagram for clock control counter.

7. 링크 컨트롤러의 비트 스트림 처리

RF를 통해 전송을 하기 전에 데이터를 인코딩하는

과정은 불완전한 채널(RF)을 통해 데이터가 전송될 때 발생할 수 있는 데이터의 손상에 대한 처리를 하기 위한 중요한 부분으로 그림 11에서 링크 컨트롤러의 비트 스트림 처리를 위한 일반적인 블록 다이어그램을 보인다. 블루투스의 경우 TDD 방식의 사용으로 인해 TX와 RX의 타이밍이 정해지며 지정된 타이밍에서 하나의 패킷을 전송한다. 이러한 특성을 고려하여 비트 스트림 처리 기능 블록(HEC, CRC, Whiten, Encryption, FEC 등)간 데이터 전달 타이밍을 TX/RX 시퀀서(Sequencer)가 제어함으로써 버퍼의 사용 없이 비트 스트림을 처리 할 수 있도록 하였다. 데이터의 수신 및 송신 시에 발생할 수 있는 다양한 상황에 대한 정보는 인터럽트로써 처리하였다.

송/수신 블록의 시퀀서는 처리 과정에서 발생하는 여러 가지 정보를 링크 컨트롤러의 컨트롤 레지스터에 전달하고 패킷의 종류에 따라 각 기능 블록을 제어한다. 그림 12와 그림 13은 설계된 링크 컨트롤러 모듈의 송수신 비트 스트림 처리 블록을 보인다.

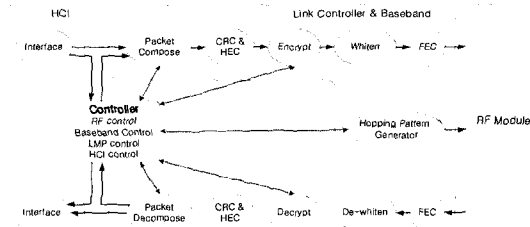


그림 11. 비트 스트림 처리를 위한 블록 다이어그램  
Fig. 11. Block diagram for bit stream processing.

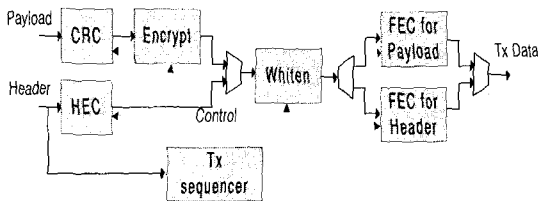


그림 12. 비트 스트림 송신 블록  
Fig. 12. Bit stream transmit block.

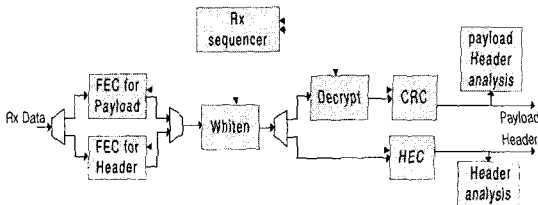


그림 13. 비트 스트림 수신 블록  
Fig. 13. Bit stream receive block.

오류 검출을 위해 사용되는 패킷의 처리 부분은 액세스코드(access code), 헤더(header)의 HEC, 그리고 유효부하(payload)의 CRC 이렇게 세 부분으로 구분된다<sup>[3]</sup>. HEC는 수신된 패킷의 오류를 검출할 수 있는 수단의 하나로써 사용된다. HEC 코드를 생성하기 위하여 먼저 HEC 생성기의 쉬프트 레지스터들을 8비트 상위 어드레스 부분(UAP)으로 초기화하고 10비트 헤더 정보(header information)를 쉬프트 한다. 다음의 그림 14는 HEC생성을 위한 다항식(polynomial)과 LFSR회로를 보인다. 이렇게 HEC 과정을 거친 헤더 정보는 1/3 FEC 과정을 거쳐 RF 인터페이스로 전송된다<sup>[2]</sup>.

$$G(D)=(D+1)(D^7+D^4+D^3+D^2+1)=(D^8+D^7+D^5+D^2+D+1)$$

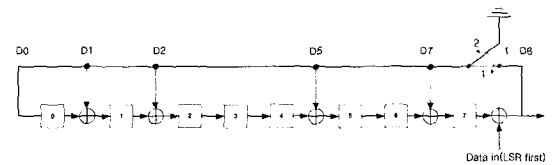


그림 14. HEC생성을 위한 LFSR  
Fig. 14. The LFSR circuit generating the HEC.

CRC(Cyclic Redundancy Check)는 보낼 데이터를 프로토콜에서 정해진 다항식에 의하여 처리한 다음 그 결과를 보낼 데이터에 추가해서 보내며, 수신측에서는 데이터를 수신할 때 오류의 발생 유무를 검사한다. CRC 처리를 위한 16 비트 LFSR 은 'CRC-CCITT generator polynomial'  $g(D)=D^{16}+D^{12}+D^5+1$  과 같은 구조를 갖는다<sup>[2]</sup>.

Whiten 블록은 데이터를 난수화(randomize)하고 DC 바이어스를 최소화하기 위하여 데이터의 비트 스트림과 슈도 랜덤 비트(pseudo random bit)를 혼합하는 것으로 '1' 또는 '0'이 긴 시간동안 하나의 값으로 반복되는 스트림 생성 가능성을 줄여준다. Whiten의 생성 다항식은  $g(D)=D^7+D^4+1$ 이고 그림 15는 생성 다항식에 의한 LFSR을 보인다<sup>[2]</sup>.

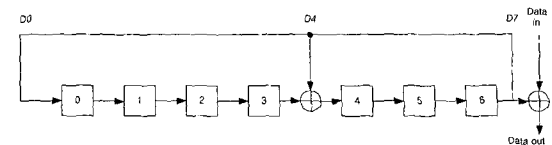


그림 15. 비트 스트림 whiten을 위한 LFSR  
Fig. 15. Bit stream whitening LFSR.



블루투스는 여러 보정을 위하여 1/3비율 FEC (Forward Error Correction), 2/3 비율 FEC, 그리고 ARQ 방법을 사용한다. 1/3 FEC는 단순히 코드를 3번 반복하는 것이며 2/3 FEC는  $g(D)=(D+1)(D^4+D+1)$ 의 다항식을 이용하여 생성한다. FEC는 오류에 대한 제한된 검출과 정정 기능을 가지며 2/3 FEC에서 최대 2 비트 오류 검출과 1비트 오류 정정 기능을 수행한다. 이는 오류의 보정을 통한 재전송의 횟수를 줄이기 위한 방법으로 이용되고 있으나 에러가 존재하지 않는 (error-free) 환경에서의 FEC는 불필요한 오버헤드를 가져올 수 있다. 따라서 FEC의 사용여부에 따른 적절한 패킷 종류의 선택이 중요하다고 할 수 있다<sup>[3]</sup>. 패킷의 종류는 링크 컨트롤러의 컨트롤 레지스터를 설정함으로써 결정되며 패킷의 종류에 따라 서가지 FEC 방법이 사용된다(none, 1/3, 2/3).

블루투스의 암호화는 SAFER+ 알고리즘을 사용하고 유료부하에만 적용되며 헤더나 액세스 코드는 암호화를 하지 않는다. 암호화 엔진은 링크 컨트롤러 모듈의 레지스터를 설정함으로써 활성화되고 랜덤 넘버에 의해서 초기화된다. 초기화가 이루어진 후 암호화 엔진은 스트림 데이터, 마스터 블루투스의 장치 주소(Master Address), 마스터 블루투스의 슬롯 클럭(slot clock), 그리고 양쪽 장치에 의해 공유되는 암호 키(secret key)를 입력으로 가진다. 암호화는 암호화 엔진에 의해서 만들어진 사이퍼 스트림(cipher stream)을 단순히 비트 스트림과 exclusive OR함으로써 수행한다<sup>[2]</sup>.

데이터의 수신은 수신되는 패킷에 따라 비트 스트림의 처리 과정이 바뀐다. Rx는 Tx와 달리 수신되는 데이터의 정보(패킷의 종류, 길이 등)를 수신 전에 알 수 없으므로 수신과정에서 이러한 정보를 인식해야 한다. 이를 처리하기 위하여 Rx 블록은 수신된 데이터의 형태를 분석하고 원하는 패킷만을 수신할 수 있도록 하는 패킷 여과 블록(filtering block)이 필요하다. 그림 13의 유료부하 헤더 분석기(payload header analysis)와 패킷 헤더 분석기(header analysis)가 이와 같은 일을 수행한다. 이 블록은 수신된 패킷이 자기 자신에게 전달되는 것인지를 확인하고 패킷의 종류가 무엇인지 그리고 전달되는 데이터의 크기 정보를 검사하여 프로세서에게 정보를 전달한다.

8. 오디오 코덱의 설계

오디오 코덱은 사용자가 레지스터를 설정하여 손쉽게

필요로 하는 코딩 방법을 고를 수 있으며 제어가 가능하게 설계하였다. 블루투스는 CVSD와 A-law, u-law 두가지 log PCM 방법의 서가지 코딩 방법을 사용하고 있다. 오디오 코덱의 경우도 링크 컨트롤러와 마찬가지로 버스 인터페이스에 DMA를 지원하도록 하였으며 PCM 칩과의 인터페이스는 Texas Instrument의 TLV320AC36칩을 기반으로 하여 설계하였다. 일반적으로 상용 기저대역 칩의 경우 오디오 코덱을 위하여 4msec 정도의 음성 데이터를 저장하는 FIFO를 갖는다.

Log PCM과 CVSD는 처리 주파수가 서로 다르다. Log PCM은 13비트 또는 14비트의 샘플을 8KHz의 주파수로 처리하여 8비트 심볼을 만들어낸다. 반면에 CVSD는 16비트의 샘플을 64KHz의 주파수로 처리하여 1비트의 심볼을 만들어낸다. 그래서 오디오 시스템을 8KHz로 동작시킬 것인지 64KHz로 할지 결정해야 한다. 64KHz의 16비트의 샘플을 처리하는 것은 log PCM의 경우는 과잉의 양이며 단순히 CVSD를 위한 부가적인 정보일 뿐이다. 그리고 오디오 시스템으로써 너무 높은 데이터 전송률이다. 그래서 8KHz 샘플과 64KHz 샘플 사이의 변환을 위한 upsampler와 downsampler가 필요하게 된다. 이 downsampler와 upsampler는 4KHz 이상의 노이즈를 생성하면 안 된다. 이를 위해 low pass filter가 필요하게 된다.

본 설계에서 downsampler에 사용된 LPF는 5차 elliptic IIR 필터로 구현하였다. 다음은 오디오 코덱에 사용된 필터의 스펙이다.

- Direct form with 5-tap, 13-bit-width, elliptic filter
- Passband frequency : 4016Hz
- Stopband frequency : 5790Hz
- Passband ripple : -2.0026dB
- Stopband attenuation : -40.2295dB

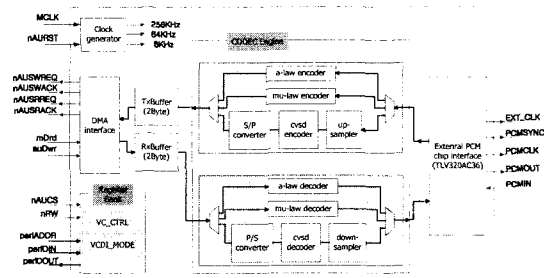


그림 16. 오디오 코덱의 블록 다이어그램  
Fig. 16. Block diagram of the audio codec.

Upsampler의 경우는 선형 interpolator를 사용하였다. 그림 16은 오디오 코덱의 블록 다이어그램을 보인다.

9. UART 인터페이스

이 UART 는 industry standard 16C450 을 기본모델로 하여 설계하였으며 300, 600, 900, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 750000, 921600, 1.5M bits/s의 baud rate을 제공한다<sup>[7]</sup>. 또한 HCI 명령어 해석을 수행할 수 있는 기능과 firmware 수정기능을 제공한다. 이 기능은 UART 를 통해서 flash rom에 저장된 firmware를 갱신할 수 있는 기능으로써 설계 시나 스펙의 변경에 따른 프로그램의 수정시 빠르게 대처할 수 있는 인터페이스를 제공할 수 있다.

IV. 기저대역 모듈의 검증 및 테스트

본 논문에서 구현한 링크 컨트롤러 모듈은 데이터 처리 및 채널 생성 시 필요한 데이터의 생성을 하드웨어적으로 처리하도록 설계하였으며 기저대역 모듈을 구성하는 링크 컨트롤러, UART, 그리고 오디오 코덱의 FIFO 사용을 없앴으로써 크기를 줄였다. 표 1의 0.35  $\mu$ m공정의 셀 라이브러리를 사용하여 합성한 결과를 보면 각 블록 별로 64바이트의 FIFO를 사용한 모듈에 비하여 약 42%의 게이트 감소를 보였다. 그러나 DMA를 사용할 경우 메모리 관리 유닛이 필요하며 설계 시 메모리 관리 유닛의 크기는 6,198게이트였다. 이 결과는 블록별로 자체 FIFO를 갖는 IP의 사용에 비하여 메모리를 데이터 버퍼로 직접 사용할 때 MMU의 추가 설계가 요구되나 이를 고려하더라도 그 크기 면에서 많은 차이를 보임을 알 수 있다.

또한 본 모듈의 동작 검증을 위하여 FPGA 테스트와 함께 ASIC을 구현하여 테스트를 수행 하였다.

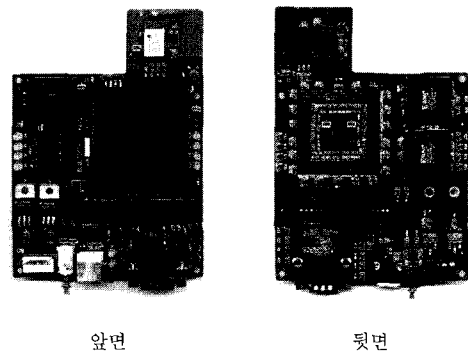
FPGA 테스트를 위하여 자일링스(Xilinx)사의 100만 게이트 virtex 칩을 사용하였으며 이 칩에 프로세서(MCU)를 함께 구현하고 에릭슨의 RF 모듈인 PBA313 01/2를 PCB에 실장하여 테스트 보드를 구현하였다. 이 FPGA에 ARM 프로세서와 링크 컨트롤러 모듈, UART, USB를 함께 구현 하였다. 그림 17은 구현된 블루투스 테스트 보드를 보여준다. 테스트 보드는 UART와 USB인터페이스를 가지며 프로그램 메모리, RF 모듈과 안테나, 테스트용 POD를 갖는다.

표 1. 기저대역 모듈의 크기 비교

Table 1. Comparison of gate sizes of the baseband modules.

	링크 컨트롤러	UART	오디오 코덱	Total size
FIFO사용(6 4bytes)	39,676	16,582	19,337	75,595
DMA사용	28,471	6,274	9,036	43,781

[단위 : gates]



앞면

뒷면

그림 17. 블루투스 테스트 보드  
Fig. 17. Bluetooth test board.

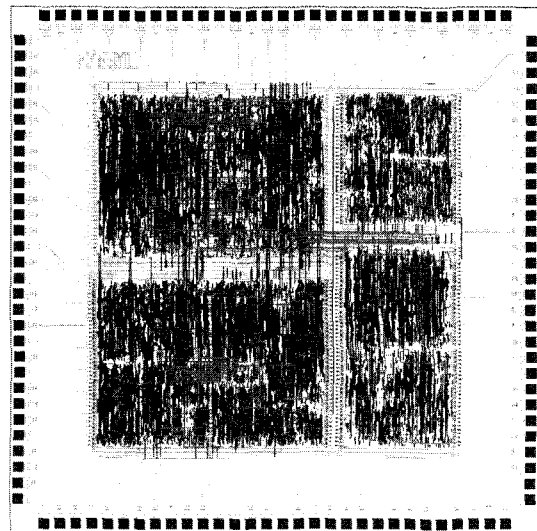


그림 18. 블루투스 모듈의 칩 레이아웃  
Fig. 18. Chip layout of the Bluetooth module.

ASIC은 반도체 설계 교육센터(IDEC)의 MPW를 이용하였으며 0.25um 5-메탈 공정의 IDEC 라이브러리를 사용하였다. 설계는 세미-커스텀(semi-custom)방법을

사용하여 구현하였다. 그림 18은 세미-커스텀으로 설계된 블루투스 기저대역 모듈의 레이아웃을 보여준다. 코어 크기는  $2.79 \times 2.8 \text{ mm}^2$  이다.

설계된 모듈을 테스트하기 위하여 두 개의 테스트 보드를 구성하고 각각을 서로 다른 PC에 UART 인터페이스를 이용하여 연결하고 파일 전송테스트 프로그램과 비트 스트림 전송 프로그램을 이용하여 링크 컨트롤러 모듈의 동작을 확인하고 검증하였다.

## V. 결 론

블루투스는 기존의 장치에 블루투스 장치를 덧붙임으로써 간단히 무선네트워크를 구축할 수 있으며 다양한 어플리케이션 및 이더넷(Ethernet)과 같은 프로토콜을 적용할 수 있어 산업 및 일상 전반에 걸쳐 광범위하게 사용될 수 있다. 또한 ISM 밴드를 사용함으로써 전세계 공통으로 사용할 수 있는 송수신 모듈을 제조할 수 있다는 장점이 있다. 그러나 이를 뒷받침하기 위해서는 작고 빠른 기저대역 모듈의 개발이 필수 요건이라 할 수 있다.

본 논문에서 설계된 모듈은 FPGA에 실장 되어 테스트되었고 ASIC으로도 구현하였다. 각 블록을 모듈화하여 IP로 사용될 수 있도록 하였으며 프로세서의 부하를 줄이고 크기가 작은 기저대역 모듈을 설계하는데 중점을 두어 설계하였다. 이는 블루투스 모듈 구현에

있어서 크기와 비용 절감 측면에서 이점을 제공할 것이다. 또한 사용자 편의를 위하여 UART를 이용한 firmware 수정을 지원함으로써 소프트웨어 개발에 따른 적용을 손쉽게 할 수 있다는 장점을 제공하고 있다.

## 참 고 문 헌

- [1] <http://www.bluetooth.com>.
- [2] Specification of the Bluetooth System V1.0B, December 1st. 1999.
- [3] Jennifer Bray and Charles F Sturman, *BLUETOOTH Connect Without Cables*, Prentice Hall PTR, 2001.
- [4] Brent A. Miller and Chatschik Bisdikian, *BLUETOOTH REVEALED*, Prentice Hall PTR, 2001.
- [5] Shorey, R. and Miller, B.A., "The Bluetooth Technology: Merits and Limitations," *IEEE International Conference on Personal Wireless Communications*, pp. 80~84, 2000.
- [6] ERICSSON, "PBA313 01/2 Bluetooth Radio," Nov, 1999.
- [7] National Semiconductor, "PC16550D Universal Asynchronous Receiver/Transmitter with FIFOs," Jun, 1995.

## 저 자 소 개



千翊宰(學生會員)

1973년 11월 5일생. 1998년 2월 충남대학교 전자공학과 공학사. 2000년 2월 충남대학교 전자공학과 공학석사. 2000년 3월~현재 충남대학교 전자공학과 대학원 박사과정. <주관심분야 : CAD알고리즘, 디지털 시스템 설계, VLSI 및 ASIC설계>



吳從煥(學生會員)

1977년 1월 3일생. 2000년 2월 충남대학교 전자공학과 공학사. 2002년 2월 충남대학교 전자공학과 공학석사. <주관심분야 : 무선통신, 디지털 통신 기술 및 디지털 시스템 설계>



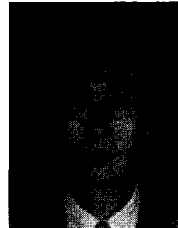
林 智 淑(學生會員)

1977년 10월 3일생. 2000년 2월 충남대학교 전자공학과 공학사. 2002년 2월 충남대학교 전자공학과 공학석사. <주관심분야 : 통신 시스템 및 디지털 시스템 설계>



金 甫 鎭(正會員)

1976년 서울대학교 전자공학과 공학사. 1978년 한국과학기술원 전기 및 전자공학과 공학석사. 1989년 미국 University of Wisconsin-Madison 전자 및 컴퓨터공학과 공학박사. 1978년~1980년 한국과학기술연구소 연구원. 1981년~1991년 금오공과대학. 1991년 3월~현재 충남대학교 전자공학과 교수. <주관심분야 : VLSI & CAD, 디지털 시스템 설계>



朴 仁 哲(正會員)

1986년 2월 서울대학교 전기공학과 공학사. 1988년 2월 한국과학기술원 전기 및 전자공학과 공학석사. 1992년 2월 한국과학기술원 전기 및 전자공학과 공학박사. 1995년 5월~1996년 5월 미국 IBM T.J. Watson Research Center 연구원. 1996년 6월~현재 한국과학기술원 전자전산학과 교수. <주관심분야 : CAD 알고리즘, VLSI 설계>