

논문 2009-46SD-6-6

차세대 무선랜 구현을 위한 MAC 엔진 설계 및 구현

(Design and Implementation of MAC Engine for Next-Generation WLAN)

이 영 곤*, 정 용 진**

(Yeong-Gon Lee and Yong-Jin Jeong)

요 약

본 논문에서는 차세대 무선랜인 802.11n의 구현에 필요한 802.11 MAC을 두 가지 버전으로 설계하였다. 설계한 첫 번째 MAC 엔진은 송신과 수신블록, 백오프, 채널관리블록을 하드웨어로 설계하였고, 프로토콜 컨트롤 블록과 MLME, MSDU처리블록을 소프트웨어로 설계하였다. 설계된 MAC 엔진은 40Mbps의 최대전송속도를 갖지만, SIFS 구간, ACK 프레임 전송과정에 소프트웨어 오버헤드로 인하여 실제 네트워크에서 정확한 동작을 보장하지 못하고 802.11n에 적용할 수 없었다. 이러한 문제를 개선한 두 번째 MAC 엔진은 MSDU처리블록, 프로토콜 컨트롤 블록을 포함한 프레임 전송에 관련된 블록들을 하드웨어로 설계하였다. 개선된 MAC 엔진은 802.11 환경에서 73Mbps의 최대 전송속도를 갖고, 802.11n 환경에서 프레임 Aggregation을 이용하면 129Mbps의 최대 전송속도를 갖는다. 본 논문에서 설계한 MAC 엔진은 차세대 무선랜에 적합한 구조임을 알 수 있다.

Abstract

This paper presents implementation of two types of the 802.11 MAC engine for the next generation WLAN, 802.11n. The first version of MAC engine consists of hardwired logic and embedded firmware. Hardwired logic includes Tx block, Rx block, Backoff block, and ChannelManage block. Embedded firmware contains Protocol Control block, MLME block, and MSDU processing block. The first version has a time-critical fault during the atomic transmission caused by software overhead, so it can not be applied to 802.11n MAC. For that reason, the second version has additional blocks with hardwired logic modules to reduce software overhead of the first version. This enhanced version has 73Mbps throughput and it is expected to be further improved up to 129 Mbps with frame aggregation which is one of the key additional features of 802.11n. As a result, the second version of MAC engine can be applied to 802.11n MAC.

Keywords : WLAN, 802.11, Legacy MAC, 802.11n, SoC

I. 서 론

최근의 통신 시장은 멀티미디어 데이터가 주를 이루고 있다. 이러한 시장의 흐름에 맞추어 기존의 통신방식에서 보안을 강화하고 전송속도를 향상시킨 새로운 기술이 많이 나오고 있다. IEEE에서는 차세대 무선랜의 표준인 802.11n을 제정하였고 현재 승인이 진행 중

이다. 802.11n 표준은 802.11 a/b/g와의 호환성을 보장하면서 최대 200~600Mbps의 전송속도를 제공하기 위하여 다양한 기법을 추가로 정의하고 있다. 물리계층에서는 LDPC(Low Density Parity Check) 코딩기법, MIMO(Multiple-Input Multiple-Output) 안테나기법 등이 추가되었고, MAC(Medium Access Control)에서는 프레임 Aggregation과 Block Ack, Long NAV(Network Allocation Vector)기법이 추가되었다^[3].

802.11n은 업계와 학계 모두에 큰 이슈가 되고 있다. 업계에서는 802.11n의 초안(Draft)버전을 이용한 칩셋을 이용해 시장을 선점하기위한 노력을 하고 있고, 학계에서는 802.11n의 다양한 기법을 적용하여 전송속도를 예

* 학생회원, ** 정회원, 광운대학교 전자통신공학과
(Department of Electronics and Communication
Engineering, Kwangwoon University)

※ 본 논문은 IDEC의 틀 지원과 IT-SOC 및 서울시
나노클러스터(IP-SoC)의 지원으로 수행되었습니다.
접수일자: 2008년12월24일, 수정완료일: 2009년4월28일

측하고, MAC의 프로토콜을 개선하는 논문들이 발표되고 있다^[4, 10~11, 14~15]. 참고논문 [4]는 802.11n의 특징에 대한 언급과 업계에서 개발하고 있는 초안버전의 무선랜 칩셋에 대한 분석을 다루고 있고, 참고논문 [10~11]은 802.11과 802.11n의 최대전송속도를 결정하는 요인들을 분석하고, 프레임 Aggregation방법을 이용하여 MAC의 최대전송속도를 향상 시키는 방법과 최대전송속도의 제한에 대한 내용을 다룬다. 참고논문 [14~15]는 MAC 프로토콜을 개선하여 MAC이 가지고 있는 고유한 오버헤드를 줄이고 성능을 향상시키는 방법에 대한 내용을 다룬다.

802.11n은 전송속도를 높이기 위해 다양한 기능이 추가되었지만 802.11 MAC 기능을 필수적으로 포함하고 있기 때문에 802.11 MAC을 설계하고 성능을 검증하는 것은 802.11n의 설계에 있어 필수적이다. 802.11 MAC 구현은 전통적으로 하드웨어와 소프트웨어의 통합설계로 이루어지고 있고, 802.11의 구현을 다루는 다양한 논문들이 발표되었다^[16~19]. 참고논문 [16]은 802.11b를 임베디드 환경에서 구현한 것으로 Xilinx사의 Micro Blaze프로세서와 PCMCIA(Personal Computer Memory Card International Association) 인터페이스를 하드웨어로 설계하였고, RTOS(Real Time Operating System)기반의 소프트웨어를 이용해서 MAC을 설계하였다. 참고논문 [17~19]는 802.11 MAC에서 타이밍에 민감한 송수신 블록과 프로토콜 콘트롤 블록 및 암호화 블록을 하드웨어로 설계하고, 메모리 액세스와 암호화의 일부기능과 인증 및 접속 등과 같이 타이밍에 민감하지 않은 블록을 소프트웨어로 구현하였다.

기존의 논문은 802.11 MAC의 구현에 초점을 맞추었고 802.11n MAC의 적용에 대한 내용은 다루고 있지 않다^[16~19]. 또한 설계된 MAC의 성능에 대한 구체적인 데이터를 제공하지 않기 때문에 MAC 구현에 있어서 비교자료로 사용하기는 힘들다. 따라서 본 논문에서는 먼저 데이터의 송수신과 백오프, 채널관리모듈을 하드웨어로 설계한 802.11 MAC 엔진 v.1을 구현을 시작으로 MAC 엔진 v.1이 갖는 문제점을 개선하여 성능을 높인 802.11 MAC 엔진 v.2를 설계하였고 802.11n 환경에 적용하여 성능을 예측하였다.

II. 802.11 MAC 구조

1. 802.11의 버전과 특징

IEEE 802.11은 물리 계층과 MAC의 표준 기술로 802.11b를 시작으로 다양한 표준이 나왔고, 최근에는 고속 무선 데이터 서비스를 위한 802.11n이 등장하였다. 표 1은 현재까지 진행된 802.11의 다양한 표준들의 성능과 특징을 보여준다. 802.11a/g의 전송 속도는 물리계층에서 최대 54Mbps이고, 802.11n은 600Mbps의 속도를 갖는다^[4]. 이 외에도 802.11 기반에 QoS(Quality of Service)를 추가하여 전송속도 향상을 보장하는 802.11e와 TKIP(Temporal Key Integrity Protocol)와 CCMP(Counter Mode with Cipher Block Chaining Message Authentication Code Protocol)를 추가하여 보안을 강화시킨 802.11i가 존재한다^[1].

표 1. 802.11 표준에 따른 특징
Table 1. Various standard for 802.11.

	802.11b	802.11a	802.11g	802.11n
최대전송률	11Mbps	54Mbps	54Mbps	600Mbps
변조방식	CCK	OFDM	OFDM, CCK	MIMO- OFDM
주파수 대역	2.4GHz	5GHz	2.4GHz	2.4GHz, 5GHz
MAC 최대전송률	5Mbps	27Mbps	27Mbps	100Mbps

2. 802.11 MAC의 구조

802.11 MAC에서는 무선 매체에 접근하기 위한 방법으로 분산 조정 함수(DCF)와 포인트 조정 함수(PCF)를 사용한다. DCF(Distributed Coordination Function)는 표준 CSMA/CA(Carrier Sensing Multiple Access with Collision Avoidance)를 이용한 경쟁 기반의 서비스를 제공하고, 각 스테이션은 프레임을 전송하기 전에 채널 사용 여부를 검사하며 충돌을 피하기 위하여 프레임 송신의 종료 시점에서 백오프를 수행한다^[2]. PCF(Point Coordination Function)는 포인트 조정자(Point Coordinator)가 채널의 접근을 제어하는 무경쟁 서비스를 제공한다. PCF는 DCF보다 우선권을 가지고 있지만 이 기능은 선택적으로 구현될 수 있으며, 현재 출시된 일반제품에서는 대부분 구현되어 있지 않다^[2]. 따라서 본 논문에서 설계한 MAC은 DCF를 이용하였다.

MAC은 무선 매체의 접근 관리 외에도 MPDU(MAC Protocol Data Unit) 송수신, 주소할당(Addressing), 프레임 형성, 에러검사, 조각화(Fragment)와 조립

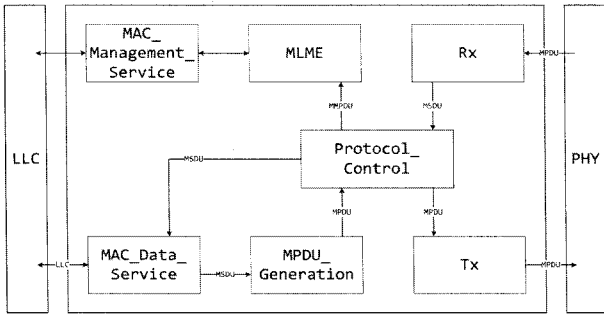


그림 1. 802.11 MAC 시스템 구조도
Fig. 1. Block diagram for 802.11 MAC.

(Defragment)을 수행한다. 이러한 기능을 구현하기 위한 세부 기능 블록구조는 그림 1에 나타나 있다^[1]. 802.11 MAC은 MPDU 프레임을 생성하는 모듈(MPDU_Generation)과 생성된 프레임을 송신 하는 모듈(Transmission), MPDU 프레임을 수신하는 모듈(Reception), 프레임의 송신과 수신을 제어하고 프로토콜을 제어하는 모듈(Protocol_Control)등이 있고 MAC의 상태를 관리하는 모듈(MLME), MAC의 상위 계층인 LLC(Logic Link Control)계층과의 데이터 송수신을 담당하고 프레임 조각화를 수행하는 모듈(MAC Data Service), MAC관리 프레임들을 담당하는 모듈(MAC Management Service)로 구성되어 있다.

III 802.11 MAC 엔진 v.1 구현

1. 하드웨어와 소프트웨어 구성

802.11 MAC 엔진 v.1은 소프트웨어 설계를 기반으로

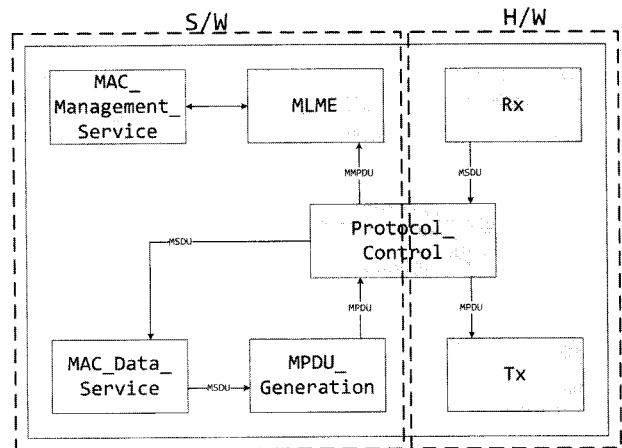


그림 2. 802.11 MAC 엔진 v.1 구조
Fig. 2. Block diagram for MAC engine v.1.

물리계층과 직접 데이터를 송수신 하는 블록을 하드웨어로 설계하여 처리속도를 향상 시켰다^[13]. 추가적으로 백오프 블록, 채널관리 블록, 64비트 로컬타이머와 MAC에서 사용되는 여러 가지 타이머들을 하드웨어로 설계하였다. 그림 2는 802.11 MAC 엔진 v.1의 하드웨어와 소프트웨어 시스템 구성을 나타낸다.

2. 802.11 MAC 엔진 v.1 구조

802.11 MAC 엔진 v.1의 하드웨어 구조는 그림 3과 같이 AMBA AHB를 이용하여 소프트웨어와 인터페이스를 하도록 설계되었고, 프레임 송신 블록(TX)과 수신 블록(RX), 무선매체 관리 블록(CHANNEL_MANAGE), 무선매체에서의 충돌을 피하기 위해 랜덤 백오프를 수행하는 블록(BACKOFF)으로 구성되어 있다. 또한 64비

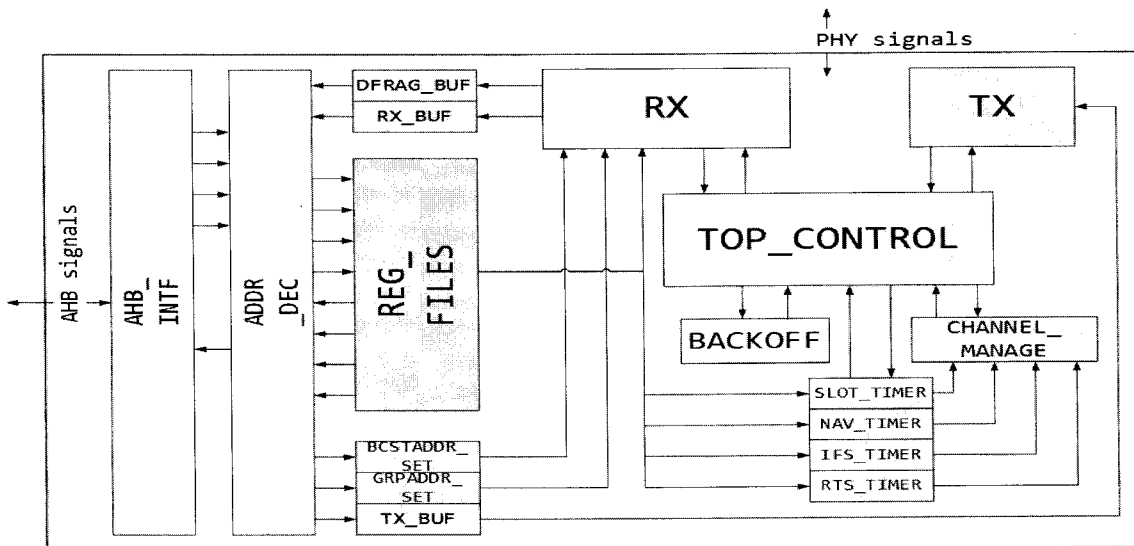


그림 3. 802.11 MAC 엔진 v.1 하드웨어 구조도
Fig. 3. Block diagram for MAC engine v.1 hardware.

트 로컬 타이머(LOCAL_TIMER)와 MAC 동작에 필요한 타이머블록들, 하드웨어 제어블록(TOP_CONTROL)으로 구성되었다.

3. 802.11 MAC 엔진 v.1 하드웨어 합성

설계된 802.11 MAC 엔진 v.1 하드웨어의 검증에 사용된 플랫폼은 ARM926EJ-S 코어타일과 Virtex-5 FPGA 모듈을 탑재한 ARM사의 EB보드(Emulation Baseboard)로 구성되었다. 그림 4에서 보듯이 코어타일과 FPGA모듈은 AMBA AHB방식으로 연결되어 있고, EB에는 메모리와 각종 인터페이스를 포함하고 있다. 동작 주파수는 33MHz로 ARM코어와 FPGA모듈, AHB 버스 모두 동일 클럭을 사용하고 있다. MAC 엔진 v.1 하드웨어의 검증을 위해 MAC 엔진 v.1 하드웨어 2개와 검증용 물리계층 로직 2개를 연결하여 두개의 스테이션으로 구성하였다. 또한 DMA(Direct Memory Access)컨트롤러를 이용하여 MPDU 데이터를 하드웨어에 전달하였다.

설계된 MAC 엔진 v.1 하드웨어는 Xilinx사의 ISE 10.1 툴을 이용하여 Virtex5 FPGA를 타겟으로 합성하였고, 그 결과는 그림 5에 나타나 있다. 최대 동작 속도

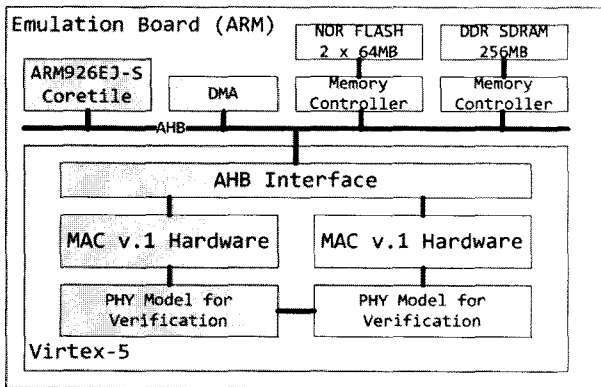


그림 4. ARM사의 EB 플랫폼 구조도
Fig. 4. Block diagram of EB platform.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	3475	207360	1%
Number of Slice LUTs	4356	207360	2%
Number of fully used LUT-FF pairs	1107	6724	16%
Number of bonded IOBs	199	1200	16%
Number of Block RAM/FIFO	36	298	12%
Number of BUF6/BUF6CTRLs	1	32	3%

그림 5. 802.11 MAC 엔진 v1. 하드웨어 합성 결과
Fig. 5. Synthesis result of MAC engine v.1 hardware.

는 148MHz이고 3,475개의 레지스터와 4,356개의 LUT, 262Kbits의 메모리를 사용하였다.

4. 802.11 MAC 엔진 v.1 하드웨어 ASIC

802.11 MAC 엔진 v.1 하드웨어의 ASIC은 Synopsys사의 Design Compiler툴에서 매그너칩스의 0.35공정 라이브러리를 이용하여 합성하였다. MPW의 다이 크기인 4mm x 5mm의 크기 제한 때문에 조립화를 위한 메모리 개수를 1개로 줄였고, 송수신에 사용된 메모리의 크기도 18Kbits로 크기를 줄였다. 또한 최대 동작 속도를 검증 플랫폼의 동작 클럭인 33MHz로 낮추어서 로직의 크기를 줄였다. 표 2는 MAC v.1 하드웨어의 FPGA와 ASIC 합성 결과를 보여준다.

그림 6은 Synopsys사의 Astro툴을 이용하여 백엔드 작업을 한 레이아웃이다.

표 2. MAC v.1 하드웨어의 FPGA와 ASIC 합성 결과
Table 2. Synthesis result for FPGA and ASIC of MAC engine v.1 hardware.

	ASIC	FPGA
Library	Magna chip, 0.35	Xilinx, Virtex5
Size	12만 게이트	4,356 LUT
Frequency	33MHz	148MHz
Memory Size	78Kbits	262Kbits

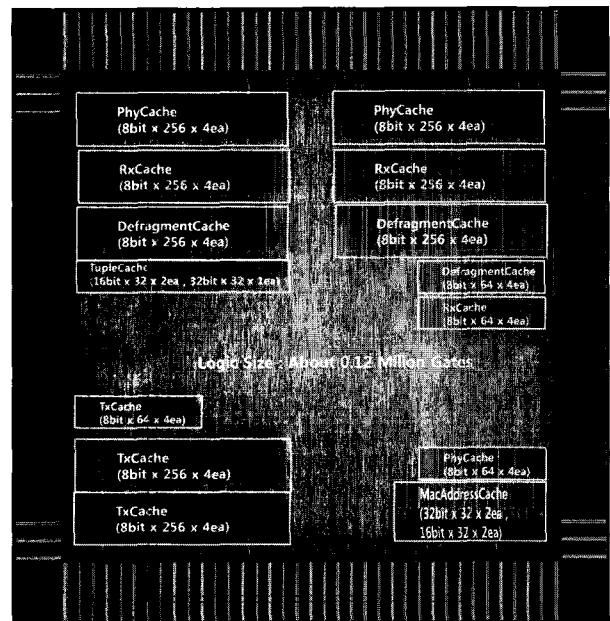


그림 6. 802.11 MAC 엔진 v.1 하드웨어 레이아웃
Fig. 6. 802.11 MAC engine v.1 hardware ASIC layout.

5. 802.11 MAC 엔진 v.1의 검증

MAC에서 최대전송속도를 구하기 위해서는 채널의 상태가 에러 없이 깨끗해야 하며, 네트워크에 오직 두 개의 스테이션만 존재하여 하나의 스테이션은 데이터를 계속 보내고 다른 스테이션은 계속 수신과 응답만 하는 동작을 수행해야 한다^[12]. 본 논문에서는 물리계층의 특징을 SIFS구간은 16us, DIFS구간은 34us, SLOT구간은 9us로 하여 802.11a의 표준에 따르도록 하였다^[12].

본 논문에서 검증을 수행하기 위해 다음과 같은 과정을 수행하였다. 송신 스테이션은 LLC계층으로부터 2,304바이트의 데이터 전송 요청을 받아 MPDU 프레임을 생성해서 송신한다. 수신 스테이션에서는 MPDU 프레임을 수신한 후, SIFS구간 동안 대기하고 ACK 프레임 송신 스테이션에 전송한다. 송신 스테이션에서는 ACK 프레임 수신 후 DIFS구간 동안 대기한 이후 백오프를 수행한다. 이러한 과정을 아토믹 전송이라 한다^[2]. MAC의 최대 전송속도를 위해서 물리계층의 전송 속도는 무한대로 하였다. 또한 LLC계층에서 MAC으로 전송할 데이터를 저장하는 시간은 MAC의 동작시간에서 제외하였는데, MAC의 성능은 LLC계층에서 데이터를 전송하는 시간과는 무관하기 때문이다. 프로토콜 컨트롤 블록은 소프트웨어로 설계하였기 때문에 프로토콜 컨트롤 블록의 동작시간에는 ACK 프레임을 MAC 하드웨어에 전송하는 시간이 포함된다. 설계된 MAC 엔진 v.1의 최대 전송속도는 그림 7에 나타난 것과 같

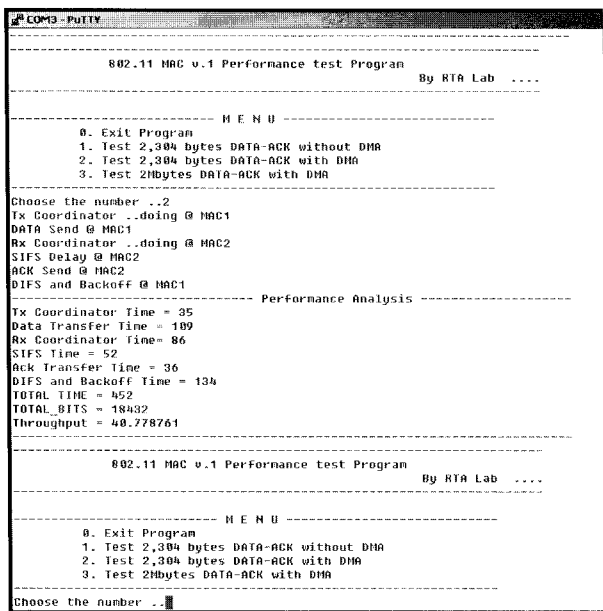


그림 7. 802.11 MAC 엔진 v.1 하드웨어의 동작 검증
Fig. 7. Emulation result of MAC engine v.1 hardware.

이 40Mbps가 된다.

설계된 MAC 엔진 v.1 하드웨어의 동작을 분석해보면, SIFS구간은 16us인데 실제 수행된 시간은 52us이고 ACK 프레임 전송은 1us 미만이나 실제로 36us가 걸렸다. 이러한 문제는 소프트웨어의 동작 속도가 33MHz로 매우 낮고, 아토믹 전송 과정에 소프트웨어와 하드웨어의 빈번한 액세스로 인한 오버헤드 때문에 발생한다. 특히 SIFS구간이 DIFS구간보다 길기 때문에 실제 네트워크 환경에서 다른 스테이션과 충돌이 발생할 수 있다. 이 문제를 해결하기 위해서는 아토믹 전송과정에서 소프트웨어 오버헤드를 줄이는 것이 필요하다. 본 논문에서는 오버헤드를 줄이기 위해 아토믹 전송과정에 필요한 블록들을 하드웨어로 구성한 802.11 MAC 엔진 v.2를 설계하였다.

IV. 802.11 MAC 엔진 v.2 구현

1. 하드웨어와 소프트웨어 구성

802.11 MAC 엔진 v.2는 아토믹 전송을 진행하는 동안 발생하는 소프트웨어 오버헤드를 없애기 위해 프로토콜 컨트롤 블록과 MSDU(MAC Service Data Unit) 처리 블록을 하드웨어로 설계하였다. 그림 8은 802.11 MAC 엔진 v.2의 구조를 보여준다.

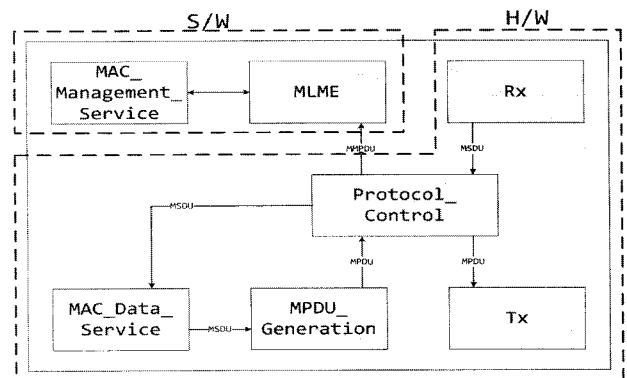


그림 8. 802.11 MAC 엔진 v.2 구조도
Fig. 8. Block diagram for MAC engine v.2.

2. 802.11 MAC 엔진 v.2 하드웨어 구조

802.11 MAC 엔진 v.2 하드웨어는 MAC 엔진 v.1과 같은 AHB 인터페이스 방식으로 설계되었다. 그림 9의 TxCoord 블록과 RxCoord블록은 MAC의 가장 핵심이 되는 프로토콜 컨트롤 블록으로 빠른 처리속도를 요구하는 연산 블록은 없지만 복잡한 컨트롤 신호를 처리

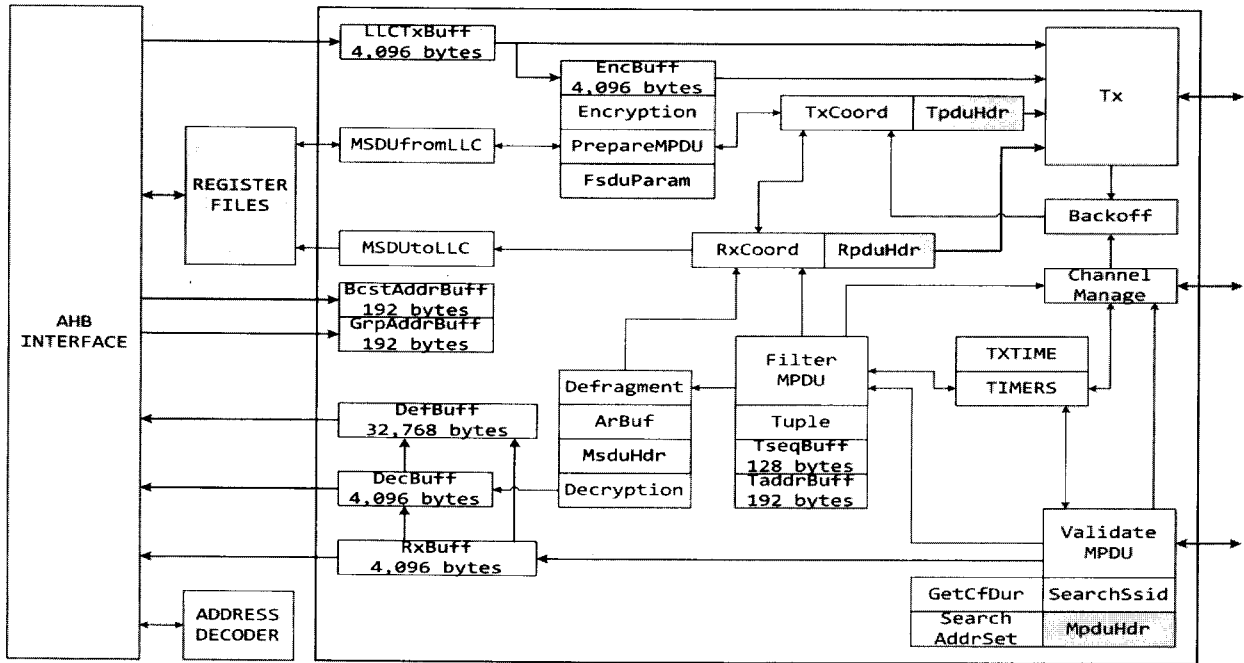


그림 9. 802.11 MAC 엔진 v.2 하드웨어 구조도
 Fig. 9. Block diagram for MAC engine v.2 hardware.

할 수 있도록 FSM(Finite State Machine)기반의 구조로 설계를 하였다. TxCoord 블록은 DCF와 PCF 제어 신호 및 SIFS구간 타이머와 랜덤 백오프를 제어한다. 또한 RTS/CTS 모드를 결정하는 역할과 재전송 처리를 담당한다. RxCoord 블록은 물리계층에서 데이터 수신 완료 후 ACK 응답과 CTS 응답 혹은 LLC계층과 MLME 블록으로 데이터의 수신확인을 요청하는 역할을 수행한다. 조각화 블록은 프레임데이터의 액세스 없이 MSDU 프레임의 길이만을 이용하여 최대 10개까지 조각화가 가능하고, 조립화 버퍼는 802.11 MAC 엔진 v.1보다 2개가 추가된 8개를 이용하여 더 많은 MSDU를 처리할 수 있도록 하였다. 또한 하드웨어 블록들 사이의 복잡한 신호를 Queue구조를 이용하여 순차적으로 처리할 수 있게 하였다. Queue구조는 입력과 출력에 각각 디코더가 들어있고 레지스터를 통해서 최대 8개까지 신호를 저장할 수 있고, 필요한 경우에는 검색을 하고 검색된 신호를 출력 할 수 있다.

3. 802.11 MAC 엔진 v.2 하드웨어 합성

설계된 802.11 MAC 엔진 v.2 하드웨어는 Xilinx사의 ISE 10.1 툴을 이용하여 Xilinx Virtex5 FPGA를 타겟으로 합성 하였고 그 결과는 그림 10에 나타나 있다. MAC 하드웨어의 최대 동작 속도는 101MHz이고 16,733개의 LUT와 9,717개의 레지스터, 398Kbits의 메모리를 사용하였다.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	9717	207360	4%
Number of Slice LUTs	16733	207360	8%
Number of fully used LUT-FF pairs	5592	20858	26%
Number of bonded IOBs	167	1200	13%
Number of Block RAM/FIFO	16	288	5%
Number of BUFG/BUFGCTRLs	1	32	3%

그림 10. 802.11 MAC v.2 하드웨어 합성 결과
 Fig. 10. Synthesis result of MAC engine v.2 hardware.

모리를 사용하였다. 프레임을 저장하기 위한 버퍼는 4,096 바이트가 저장 가능한 듀얼포트 SRAM을 이용하였다.

4. 802.11 MAC v.2의 동작 검증

설계된 802.11 MAC 엔진 v.2 는 LLC계층과 물리계층간 데이터전송을 담당하고 이를 위해서 소프트웨어를 이용하여 LLC계층의 인터페이스를 설계하였다. 소프트웨어 LLC계층은 DMA를 이용하여 MAC 엔진 v.2 하드웨어의 메모리에 전송할 데이터를 저장하거나 수신한 데이터를 읽어와도록 하여 데이터 전송으로 인한 오버헤드를 줄였다. 그림 11은 아토믹 전송에 대한 MAC 엔진 v.2 하드웨어의 처리속도이다. 이것은 MAC 최대 전송속도와는 다른 것으로, MAC 하드웨어의 처리속도는 LLC의 데이터 전송요청 이후부터 데이터 전송에 대한

결과를 확인하는 시간에 근거하여 MAC 하드웨어의 최대 성능을 구한 것으로 그림 11에서 보는바와 같이 44Mbps 가 나온다. 구체적으로 살펴보면 LLC계층에서 MAC으로 데이터 전송을 하는 시간이 246us 걸리고 프레임 전송에 169us가 걸린다. 데이터 전송시간을 제외한다면 약 122Mbps의 처리속도를 갖는다.

V. 결과 및 분석

1. 결과 및 분석

802.11 MAC 엔진 v.1은 아토믹 전송에 있어서 소프트웨어의 오버헤드가 발생하는 것을 확인하였고, SIFS 구간이 DFS 구간보다 길기 때문에 실제 네트워크에서는 충돌(Collision)이 발생할 수 있다.

반면 802.11 MAC 엔진 v.2는 아토믹 전송에 관련된 블록들을 하드웨어로 설계하여 전송과정에 소프트웨어 오버헤드가 존재하지 않고 프로세서의 성능에 관계없이 동작 성능을 보장받을 수 있다. 반면 802.11 MAC 엔진 v.2는 아토믹 전송에 관련된 블록들을 하드웨어로 설계하여 전송과정에 소프트웨어 오버헤드가 존재하지 않고 프로세서의 성능에 관계없이 동작 성능을 보장받을 수 있다. MLME(MAC Layer Management Entity)와 관련

된 블록은 인증과 접속 등과 같이 제한된 요청이 있을 경우에만 동작을 하기 때문에 하드웨어로 설계할 필요는 없다.

표 3은 두 가지 버전의 MAC 하드웨어의 최대 전송 속도를 비교한 것으로 MAC 엔진 v.1의 동작 속도는 소프트웨어를 이용하여 측정하였고, MAC 엔진 v.2는 아토믹 전송 과정에 소프트웨어의 액세스가 전혀 없기 때문에 소프트웨어를 이용해서 동작을 측정하지 않고 Xilinx에서 제공하는 칩스코프를 이용하여 동작 시간을 측정한 결과 값이다. 동작 시간의 측정에는 MAC 하드웨어에 설계된 64비트 로컬타이머를 이용하였다.

2. 802.11n 적용 및 성능 예측

설계된 MAC 엔진 v.2 하드웨어를 이용하여 802.11n의 전송속도를 예측하였다. 본 논문에서 사용한 프레임 Aggregation 방법은 MPDU Aggregation기법이다.

이 방법은 MPDU를 최대 65,535바이트의 크기를 갖는 A-MPDU(Aggregation MPDU)로 확장시키는 것으로, 802.11n에서 최대 전송속도를 위해 사용되는 기법이다^[13]. A-MPDU 프레임과 A-MPDU 서브프레임 포맷을 각각 그림 12와 13에 나타내었다[3].

표 3. 설계된 MAC 하드웨어 성능 비교
Table 3. Comparison MAC engine v.1 with MAC engine v.2 hardware performance.

	MAC v.1	MAC v.2
DATA 전송	230 us	138 us
SIFS	52 us	15 us
ACK 전송	36 us	1 us
DFS	134 us	33 us
랜덤 백오프		63 us
최대전송률	40 Mbps	73.7 Mbps

```

CDM3 - PuTTY
4. Test 2,304 Bytes RTS-CTS with Fragmentation
-----
Choose the number ..2
Input LLC length ( Maximum 4096 bytes ) : 2304
LLC Frame Generated
LLC Frame Copied with DMA
LLC Request @ MAC1
LLC Request Confirm @ MAC1
  Send Address = 0x5544332211
  Destination Address = 0x5040302010
LLC DataIndicate @ MAC2
  rxStatus = o.k.
  rxLength = 2304
  rxDataAddr = e0030000
  Send Address = 0x5544332211
  Destination Address = 0x5040302010
-----
Performance Analysis
LLC Write start Time = 68838403196
LLC Write end Time = 68838403442
LLC Write Time = 246
LLC Request Time = 68838413999
LLC Confirm Time = 68838414168
Transmission Time = 169
TOTAL TIME = 415
TOTAL BITS = 18432
Throughput = 44.414458 Mbps
-----

802.11 MAC v.2 Performance test Program
By RTA Lab

-----
H E N U
0. Exit Program
1. Test DATA-ACK without DMA
2. Test DATA-ACK with DMA
3. Test 2 Mbytes DATA-ACK with DMA
4. Test 2,304 Bytes RTS-CTS with Fragmentation
-----
Choose the number ..
    
```

그림 11. 802.11 MAC v.2 하드웨어 동작 결과
Fig. 11. Emulation result of MAC engine v.2 hardware.

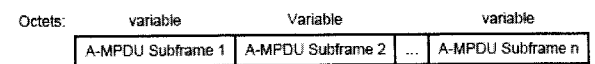


그림 12. A-MPDU의 프레임 포맷
Fig. 12. A-MPDU format.

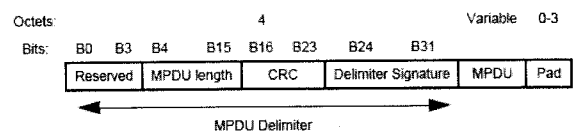


그림 13. A-MPDU의 서브 프레임 포맷
Fig. 13. A-MPDU subframe format.

802.11 MAC에서 MPDU의 최대 크기는 2,332바이트이고 이를 A-MPDU 서브 프레임으로 확장 시키면 2,336바이트의 크기가 된다[3]. 28개의 A-MPDU 서브 프레임을 이용하여 최대 65,408바이트의 A-MPDU를 전송 할 수 있다는 것을 알 수 있다. 2,332바이트 전송하는데 138us가 걸리는 것을 기준으로 65,408바이트 전송에는 3,871us가 걸리는 것으로 예측 할 수 있다. MAC 하드웨어에서 Aggregation처리로 인한 지연시간은 최대 1us로 가정한다. 표 3을 근거로 802.11n에서 Aggregation을 이용한 최대 전송속도는 129Mbps로, 802.11n MAC의 최대 전송속도인 100Mbps을 만족시키고, 본 논문에서 설계한 802.11 MAC 엔진 v.2 하드웨어는 802.11n의 구조로 적합한 것을 알 수 있다^[4].

IV. 결 론

본 논문에서는 802.11n을 구현하는데 핵심이 되는 802.11 MAC 엔진을 설계하였고, 하드웨어와 소프트웨어의 구성을 변경하여 802.11 MAC구조를 802.11n에 적용시킬 수 있음을 확인하였다. MAC 엔진 v.1은 40Mbps의 전송속도를 갖지만 소프트웨어로 인한 오버헤드로 실제 네트워크에서는 적용하기 힘들다. MAC 엔진 v.2는 프로토콜 컨트롤 블록과 MSDU처리 블록을 하드웨어로 설계하여 79Mbps의 최대 전송속도를 갖도록 하였다. 또한 프레임 Aggregation을 이용하면 802.11n에서 129Mbps의 최대 전송속도를 갖는 것으로 예측된다.

현재는 설계된 MAC 엔진 v.2의 구조에 802.11i에서 사용되는 보안엔진과 Block Ack, Aggregation, QoS를 추가한 802.11n MAC을 설계하고 있다.

참 고 문 헌

[1] IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE standard P802.11-REVma, 2006.
 [2] Matthew S. Gast, "802.11 Wireless Networks : The Definitive Guide," 2nd Edition, O'Reilly, April 2005.
 [3] IEEE P802.11n/D2.00, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," Feb. 2007.
 [4] 최은영 외, "IEEE 802.11n 차세대 무선 LAN 칩셋

개발 동향," 전자통신동향분석 21권 제3호, June 2006.
 [5] Telelogic SDT 3.2, "SDT getting Started Part 1 : Tutorials on SDT Tools," September 1997.
 [8] Goran Panic, Daniel Dietterle, Zoran Stamenkovic, Klaus Tittlebach-Helmrich, "A System-on-Chip Implementation of the IEEE 802.11a MAC Layer," Proceedings of the Euromicro Symposium on Digital System Design(DSD'03), March 2003.
 [9] ARM, "RealView Emulation Baseboard HBI-0140 Rev C User Guide," ARM DUI 0303C, 2007
 [10] Youngsoo KIM, "Throughput Enhancement of IEEE 802.11 WLAN via Frame Aggregation," 2004.
 [11] Yang Xiao, "IEEE 802.11n: ENHANCEMENTS FOR HIGHER THROUGHPUT IN WIRELESS LANS," IEEE Wireless Communications, December 2005.
 [12] Yang Xiao, "Throughput and Delay Limits of IEEE 802.11," IEEE Communications letters VOL.6 NO.8, August 2002.
 [13] Daniel Dietterle, "Design of a Hardware Accelerator for the 802.11 MAC Layer," Diploma Thesis, June 2002.
 [14] Seongkwan Kim, Youngsoo Kim, Sunghyun Choi, "A High-Throughput MAC Strategy for Next-Generation WLANs, " Proceedings of the Sixth IEEE WoWMoM'05, 2005.
 [15] Min Sheng, Jaingdong Li, Fan Jiang, "Hybrid Splitting Algorithm for Wireless MAC," IEEE Communications Letters VOL.9 No.5, May 2005.
 [16] Yang LIU, Boan LIU, "MAC Implementation with Embedded System," ASIC, 2003. Proceedings. 5th International Conference VOL.2, October 2003.
 [17] Youjin Kim, Haewon Jung, Hyeong Ho Lee, Kyoung Rok Cho, "MAC Implementation for IEEE 802.11 Wireless LAN," ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001. Joint 4th IEEE International Conference, April 2001.
 [18] J. Thomson, et al., "An Integrated 802.11a Baseband and MAC Processor," ISSCC Dig. Tech. Papers, Feb. 2002.
 [19] Teresa H. Meng, et al., "Design and Implementation of an All-CMOS 802.11a Wireless LAN Chipset," IEEE Communication Magazine, August 2003.

저 자 소 개



이 영 곤(학생회원)
 2007년 광운대학교 전자공학과
 학사 졸업.
 2009년 2월 광운대학교
 전자공학과 석사 졸업
 예정
 <주관심분야 : 무선통신, SoC설
 계, 신호처리, 임베디드 시스템>



정 용 진(정회원)
 1983년 서울대학교 제어계측
 공학과 학사 졸업.
 1983년 3월~1989년 8월 한국전자
 통신연구원.
 1995년 미국 UMASS 전자전산
 공학과 박사 졸업
 1995년 4월~1999년 2월 삼성전자 반도체 수석
 연구원.
 1999년 3월~현재 광운대학교 전자통신공학과
 교수
 <주관심분야 : 무선통신, 정보보호, SoC 설계, 영
 상처리 및 인식, 임베디드 시스템>