

---

# EPON에서 공평한 광 채널 공유를 지원하는 RC-DBA 알고리즘의 FPGA 구현

장종욱\* · 강현진\*\*

The FPGA implementation of the RC-DBA algorithm in the EPON

Jong-wook Jang\* · hyun-jin Kang\*\*

---

본 연구는 2006년도 동의대학교 일반연구과제에 의해 지원되었음.

---

## 요 약

EPON의 상향전송방식에서는 다수의 ONU가 공유된 광 채널에 대한 권한을 공평하고 효율적으로 할당받기 위해서 동적대역할당 알고리즘을 사용한다. RC-DBA 알고리즘은 MPCP를 기반으로 QoS를 지원하면서 동일한 우선 순위일 경우 모든 ONU에게 공평한 대역할당을 지원하기 위해서 제안된 DBA 알고리즘이다. 본 논문에서는 RC-DBA 알고리즘을 적용한 OLT의 MAC 모듈과 MAC 제어 패킷을 송·수신하는 ONU 모듈을 하드웨어 기술 언어 (Hardware Description Language)로 설계하였다. 또한 두 모듈을 UTP 케이블로 연결한 ONU/OLT 테스트베드 시스템을 구축하여 RC-DBA 알고리즘을 통해 상향전송을 위한 타임슬롯의 할당이 어떻게 이루어지는지 확인하였다. 본 연구에서는 Corebell사의 LDS2000 FPGA Expansion 보드를 통하여 ONU/OLT 하드웨어 모듈과 임베디드 리눅스 기반의 검증 프로그램의 개발이 이루어졌다.

## ABSTRACT

In the upstream link of EPON, numerous ONUs In the reverse link of the EPON network, numerous ONUs receive the privileges to use the optical medium from the scheduler of the LOT, but not through the competition with others. Therefore, it is very important to select a proper DBA algorithm to allocate the frequency band to each ONU in an effectively and fair manner. In our preceding study, we proposed the RC-DBA algorithm that complements many problems in existing DBA algorithms. In this paper, we designed the MAC scheduler for the OLT, which the proposed algorithm was applied to and implement it in the FPGA. In addition, in order to verify the operation of the scheduler, we developed the embedded Linux based testbed.

## 키워드

Ether-PON, Request Counter-DBA, FPGA, Embedded Linux

---

\* 동의대학교 컴퓨터공학과 교수

접수일자 : 2006. 11. 21

\*\* 동의대학교 대학원 석사과정

## I. 서론

광 네트워크 기술은 적용 영역에 따라 광 가입자망과 광 코어망으로 분류되며, 원래는 코어망 기술로서 연구되었으나 최근에는 교환기에서 가정까지 이르는 가입자망에서도 매체 사용의 대용량화가 시작되면서 여러 가지 가입자망 기술들이 광 네트워크 기술 발전을 이끌어 가고 있다[1]. 특히, 중앙 사무국에 OLT(Optical Line Termination)가 설치되고, 1:N의 수동 광분기기를 통하여 다수의 가입자 장치인 ONU(Optical Network Unit)가 OLT에 연결되는 형태인 PON(Passive Optical Network)구조가 가입자 망에 알맞은 구조로서 활발히 연구되고 있다.

Ethernet-PON은 FTTH를 저가에 고속으로 실현할 수 있는 기술로서 2004년 6월, IEEE802.3ah EFM TF에 의해 표준화가 완료되었다. 이 기술은 가입자 장치들이 Ethernet이 주종을 이루는 것에 착안하고, 거대한 시장 규모에 의해 이미 저가격화된 장점을 살릴 수 있도록 Ethernet 프로토콜을 가입자망으로까지 확장한 것이다.

EPON은 하향으로는 방송구조, 상향으로는 ONU, OLT간의 1:1 채널이 이루어지는 상·하향 비대칭 공유 구조를 갖는다. EPON의 상향 전송방식에서는 다수의 ONU가 한 가닥의 광섬유를 공유하기 때문에 서로 충돌 없이 공유하는 방법이 필요한데 IEEE802.3ah EFM에서는 EPON의 매체접근제어방식으로 MPCP(MultiPoint Control Protocol)를 정의하고 있다. 각 ONU의 사용시간은 MPCP를 이용하여 ONU의 요구에 따라 OLT가 시간을 할당하는 REQUEST-GRANT 방식으로 할당받는다[2]. MPCP에서는 EPON의 성능을 결정하는 요소인 패킷 스케줄링 알고리즘에 관한 부분은 표준화 대상에서 제외하고 있는데, 이는 가입자 서비스를 제공하는 업체에 유연성을 제공하기 위함이다. 실제 각 개발 업체들은 고유의 방식으로 하드웨어 혹은 임베디드 프로세서 기반 위에 소프트웨어 방식으로 DBA를 구현하고 있다.

이에 본 논문에서는 EPON의 모든 가입자들이 공평한 광 대역폭을 할당받기 위해 제안된 RC-DBA 알고리즘[3]을 소개하고, 이것을 FPGA칩에 구현하였다. 또한 REQUEST-GRANT MAC Control 메시지를 송·수신할 수 있는 ONU를 제작하고 RC-DBA모듈과 TCP 연결을 통해 대역폭 할당이 어떻게 이루어지는지 확인하였다.

본 연구에서 사용된 장비는 Cyclone EP1C12F324C8

FPGA칩이 내장된 보드와 Intel PXA255 프로세서를 탑재한 임베디드 보드로 구성된 Corebell사의 LDS2000 FPGA Expansion ver 1.0 보드를 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구인 MPCP와 RC-DBA 알고리즘에 대해서 좀 더 자세히 살펴보고, 3장에서 RC-DBA 알고리즘을 적용한 OLT의 MAC 모듈과 ONU의 FPGA 구현에 대해서 다룬다. 4장에서는 설계된 두 모듈의 동작 검증을 위해 개발된 임베디드 리눅스 기반의 검증 프로그램들을 설명하고 5장에서 결론을 맺는다.

## II. 관련 연구

### 2.1. Multipoint Control Protocol (MPCP)

MAC 프로토콜은 매체를 공유하는 장치들 간의 통신을 허용하기 위해서 매체접근을 제어하는 프로토콜을 의미하며, 따라서 점대다점 링크를 사용하는 곳에서는 유·무선에 상관없이 구현이 되어야 통신이 가능하다. MPCP[4]는 EPON에서 사용하는 MAC 프로토콜로서 집근 프로토콜의 핵심인 상향전송에서의 동적 할당을 위해서 사용된다.

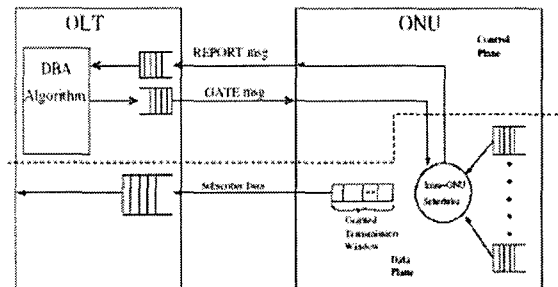


그림 1. MPCP 전달 절차[5]  
Fig. 1. Procedure of MPCP Transfer

그림1은 MPCP에서 TDMA 방식을 통한 상향 채널을 공유하기 위하여 OLT와 ONU사이에서 REPORT와 GATE 메시지가 상호 교환되는 모습이다. 그림의 ONU 부분에 표시된 바와 같이, 각 ONU는 이더넷 프레임들을 우선순위에 따라 보유한 큐의 집합을 가지고 있다. ONU는 내부 큐에 쌓여 있는 데이터 프레임들을 전송하기 위하여 OLT에게 주기적으로 REPORT 메시지를 통해 큐의 상

태를 보고한다. RI PORT 메시지를 수신한 OLT는 DBA 알고리즘 모듈에 ONU의 큐 상태 정보를 알려주고, DBA 모듈은 매체 충돌을 피하기 위하여 모든 ONU들의 상향 전송 일정을 산정하게 된다. DBA 알고리즘이 실행되고 난 후, OLT는 프레임 전송을 승인하는 의미의 GATE 메시지를 각 ONU에게 전송하고, ONU는 큐 제어 모듈인 내부 스케줄러에 의해 지정된 시간동안 큐의 데이터를 OLT로 전송할 수 있도록 한다. GATE 메시지는 전송 시작 시간과 전송 허용 길이 정보가 포함되는데, 본 연구에서는 각 ONU의 전송 시작 시간은 고려하지 않고 ONU가 전송할 수 있는 시간 길이(time slot의 개수)의 배분에 대한 논의만 이루어진다.

2.2. Request-Counter DBA 알고리즘

RC-DBA 알고리즘은 [3]에서 제안된 대역할당 방식으로서 MPCP 프로토콜을 바탕으로 다양한 트래픽에 따라 대역폭을 동적으로 할당할 수 있는 기능을 제공한다. 즉 QoS를 지원하기 위해 IEEE 802.1d[6]에 의거한 트래픽 우선순위에 따라 대역폭을 할당하되, 동일한 우선순위일 경우에 ONU들 사이의 공정한 대역할당을 지원하기 위한 절차가 추가된다.

RC-DBA 알고리즘은 먼저 우선순위가 가장 높은 큐의 요청을 처리한다. High priority 큐의 프레임들은 Fixed bandwidth를 요구하는 응용 프레임들이기 때문에 ONU가 요청한 양만큼의 대역폭이 모두 할당되고 이후에 Middle과 Low priority에 해당하는 서비스 요청량에 대한 대역폭을 차례대로 계산한다. 즉, High priority 큐의 요청량을 처리하고 남은 대역폭을 다음 순서인 Middle priority 큐에 대한 target bandwidth로 설정하는 것이다. 주목할 부분은 동일한 우선순위 큐에 대해서 특정한 ONU가 계속해서 다른 ONU에 비해 상당히 많은 양의 대역폭을 요구하게 되면 다른 ONU가 대역폭을 보장받지 못하는 경우가 발생할 수 있는데 RC-DBA 알고리즘에서는 동일한 우선순위를 가지는 큐에 대하여 ONU별로 가중치를 부여하고 가중치가 높은 순서대로 상향대역폭을 할당하는 방법을 사용한다.

본 연구의 RC-DBA 알고리즘에서 가중치를 부여할 때는 대역폭 요청의 유무와 요청량이 많은 순서의 두 가지 기준에 의거한다. 먼저 상향 대역폭을 요청한 ONU에게 정해진 가중치를 증가시킨 후, 다음 단계에서 요청량이 많은 ONU 순으로 정렬하여 가중치를 부여한다. 그림

2와 3은 ASM차트로 설계된 전체 RC-DBA 모듈에서 가중치를 부여하는 부분을 발췌한 그림이다.

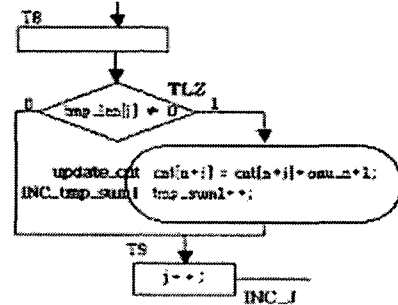


그림 2. 대역폭 요청의 유무  
Fig. 2. Request of Bandwidth

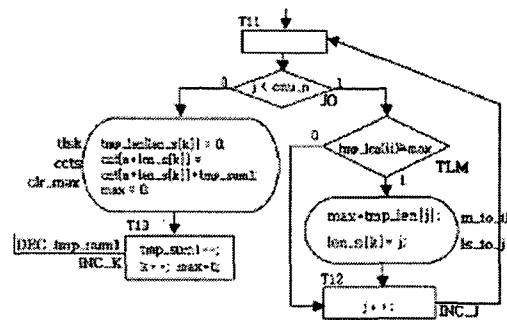


그림 3. 요청량이 많은 순서  
Fig. 3. Order of the amount of request

III. FPGA 구현

본 연구의 궁극적인 목표는 RC-DBA 알고리즘을 사용하는 OLT의 MAC 모듈을 설계하는 것이고, 구현된 MAC 모듈의 기능을 검증하기 위해서 간단한 ONU 모듈을 제작하고 Ethernet 통신을 통하여 두 모듈간의 상호 동작을 확인해 보는 것이다.

일반적인 디지털 시스템은 설계사양의 기술, ASM도표 유도, 데이터패스 및 제어로직 설계 등의 흐름으로 이루어진다. 본 연구에서는 RC-DBA 기반의 OLT MAC 모듈과 제어 메시지를 송·수신하기 위한 ONU 모듈을 HDL 언어로 작성하고, 이에 대한 test vector를 생성하여 Modelsim 툴을 이용해 프로그램 디버깅 및 소프트웨어적 동작검증을 수행하였다. 그리고 Altera 사의 Quartus

II v4.0틀을 이용하여 Cyclone EP1C12F324C8 FPGA 칩에 합성하여 하드웨어로 제작하였다. 다음 절에서는 OLT와 ONU 모듈 설계에 대해서 설명하도록 한다.

3.1. OLT의 MAC 모듈

RC-DBA MAC 모듈의 동작은 크게 네 부분으로 나눌 수 있는데 첫째 REPORT 메시지의 입력, 둘째 우선순위 큐의 요청량에 따른 가중치 계산, 셋째 대역폭 할당 및 GATE 메시지의 출력이다. 그림4는 RC-DBA 모듈의 전체적인 동작과정을 간략화한 그림으로서 IDLE 상태에서 start 신호에 1이 인가되면 REPORT 메시지의 입력을 시작으로 모듈이 동작하고, ONU측으로 GATE 메시지의 출력이 끝나면 한 번의 동작이 끝나는 것이다. 다수의 ONU는 일정한 주기로 REPORT 메시지를 계속 보내게 되므로 OLT의 DBA모듈은 이와 같은 동작을 반복하게 된다. ASM도표를 바탕으로 구현된 RC-DBA모듈의 회로는 그림5와 같이 데이터 처리 연산을 수행하는 데이터패스와 데이터 처리를 감독하는 제어논리 부분으로 이루어진다. OLT 모듈은 VerilogHDL을 사용하여 프로그램 되었다.

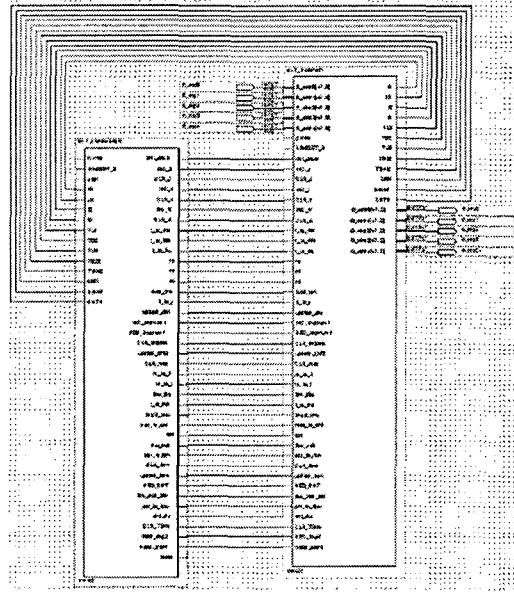


그림 5. 전체 블록도  
Fig. 5. Total Diagram

3.2. ONU 모듈

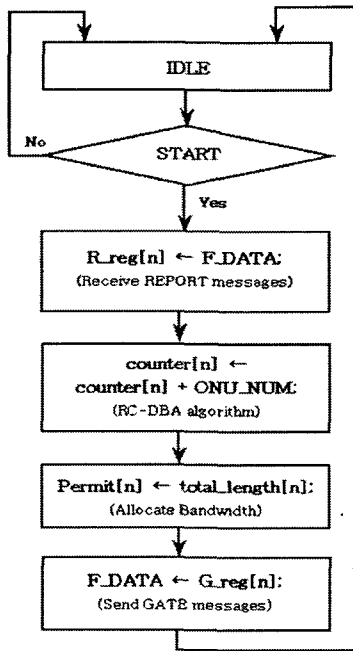


그림 4. 간략화된 ASM도표  
Fig. 4. ASM Table

OLT에게 REPORT 메시지를 전송하고 상향대역폭의 할당량 정보를 가진 GATE 메시지를 수신하는 역할을 하는 ONU는 데이터 패킷을 입력받아 우선순위로 분류하기 위한 Classifier\_module, 각 큐에 쌓인 데이터 개수를 카운터하기 위한 QueueCounter\_module, 큐의 상태 정보를 OLT에게 알리기 위해 REPORT 메시지를 생성하는 ReportCreat\_module과 같은 주요 모듈들을 포함한다. 하나의 ONU는 3개의 우선순위 큐가 내장되어 있으며 사용자측에서 수신한 데이터패킷의 우선순위를 결정하여 각각의 큐에 저장한다. 그리고 주기적으로 큐의 상태정보는 REPORT 메시지로 만들어져서 OLT에게 전송된다. 그림6은 설계된 ONU의 전체 블록도이며 ONU 모듈은 VHDL 언어로 구현되었다.

본 연구에서는 한 개의 OLT에 두 개의 동일한 동작을 하는 ONU 모듈을 허브를 통하여 연결함으로써 OLT의 대역폭 할당 실험이 이루어졌다.

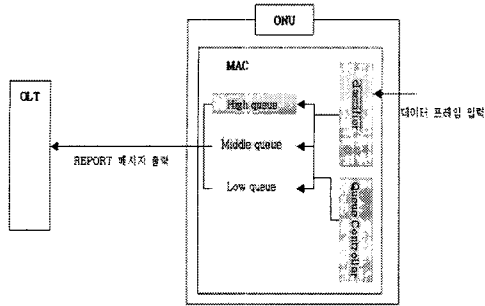


그림 6. ONU의 블록도  
Fig. 6. Block diagram of ONU

3.3 CPU 인터페이스 로직

본 연구에서는 FPGA 칩에 구현된 OLT와 ONU 모듈을 검증하기 위해서 임베디드 리눅스 기반의 검증 프로그램인 디바이스 드라이버와 응용 프로그램을 개발하여 사용했는데, 이를 위해서는 임베디드 리눅스가 탑재된 PXA255 32bit ARM CPU와 연동할 수 있는 인터페이스 로직이 필요하다.

인터페이스 로직은 CPU로부터 받은 데이터, 즉 ONU/OLT 디바이스 드라이버로부터 받은 데이터를 저장하는 레지스터들의 집합이다. 본 연구에서 각 모듈의 입·출력 데이터인 REPORT와 GATE 메시지는 48bits로 정의하였는데, LDS2000 임베디드 시스템에서 PXA255 와 FPGA 칩 사이의 입출력 버스는 최대 32bits이다. 그러므로 48bits 크기의 컨트롤 메시지 한 개를 입력받기 위해서는 디바이스 드라이버에서 8bits씩 나누어 6번에 걸쳐 인터페이스 로직으로 보내주고, 인터페이스 로직에서는 이것을 다시 48bits 데이터로 조립하여 ONU/OLT 모듈에서 사용하게 된다. 그림7은 PXA255 ARM Processor와 Cyclone FPGA 사이에서 REPORT와 GATE 메시지의 입·출력을 위해 필요한 인터페이스 제어 신호들이고 그림8은 Quartus II 에서 생성한 CPU 인터페이스 로직의 블록도이다.

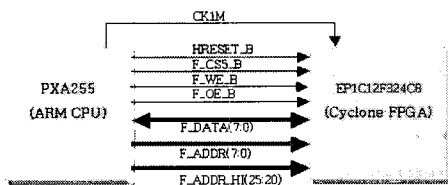


그림 7. 제어신호 인터페이스  
Fig. 7. Interface of Control Signal

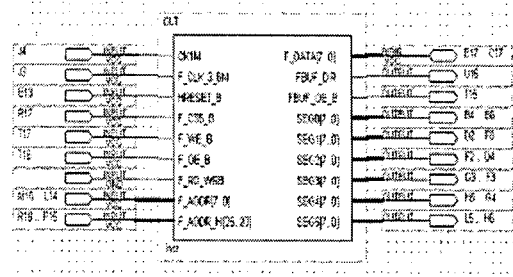


그림 8. CPU 인터페이스 로직의 블록도  
Fig. 8. Block diagram of CPU interface logic

CPU 인터페이스 로직이 포함된 ONU/OLT 모듈은 FPGA 전용 합성툴인 Quartus II를 사용하여 gate-level로 synthesize를 수행하여 programming file을 생성하고 Programming file은 ByteBlaster II를 이용하여 EP1C12F324C8 에 프로그래밍 한다. 합성된 RC-DBA 모듈은 Cyclone EP1C12F324C8 FPGA에서 22%(2,696/12,060)의 logic cell로 배선 및 배치되었으며, Registers는 9%(1,278/12,795)를 사용하였고 Actual fmax는 50.72MHz였다.

Fitter Resource Usage Summary		
	Resource	Usage
1	Logic cells	2,696 / 12,060 ( 22 % )
2	Registers	1,278 / 12,795 ( 9 % )
3	Total LABs	296 / 1,206 ( 24 % )
4	Logic cells in carry chains	128
5	User inserted logic cells	0
6	I/O pins	79 / 249 ( 31 % )
7	-- Clock pins	1 / 2 ( 50 % )
8	Global signals	3
9	M4Ks	0 / 52 ( 0 % )
10	Total memory bits	0 / 239,616 ( 0 % )
11	Total RAM block bits	0 / 239,616 ( 0 % )
12	Global clocks	3 / 8 ( 37 % )
13	Maximum fan-out node	HRESET_B
14	Maximum fan-out	1278
15	Total fan-out	13013
16	Average fan-out	4,69

그림 9. Quartus II의 Compilation Report  
Fig. 9. Complaint Report of Quartus II

#### IV. 임베디드 리눅스 기반의 검증 시스템

특정 장치를 사용하여 원하는 작업을 처리하려면 먼저 장치와 관련된 디바이스 드라이버와 응용 프로그램을 개발하여 원하는 작업을 수행하게 된다. 디바이스 드라이버는 ONU/OLT 장치와의 데이터 입·출력 과정을 수행하고, 응용 프로그램은 사용자가 원하는 처리 작업을 제어하게 된다.

그림10은 OLT 모듈의 디바이스 드라이버와 응용프로그램에 대한 컴파일부터 실행까지의 구동과정을 나타낸다. 임베디드 시스템은 제한적인 특성으로 인하여 프로그램 개발환경을 자체적으로 지원해 줄 수 없기 때문에 일반 리눅스 PC에서 프로그램의 작성 및 컴파일을 수행한다. 그림의 하드디스크 내부는 일반 리눅스 PC에서 디바이스 드라이버 `olt_dd.c`와 응용프로그램 `olt_app.c`를 작성하고 크로스 컴파일러를 통해 임베디드 리눅스용의 오브젝트 파일과 실행파일을 만드는 과정을 나타낸다. 생성된 모듈과 실행파일은 실제 사용하게 될 임베디드 시스템으로 전송하게 된다. FPGA 칩의 OLT 장치를 사용하기 위해서는 먼저 모듈 프로그램(`olt_dd.o`)을 임베디드 리눅스에 `insmod`하여 임베디드 시스템의 커널 메모리에 적재하고, `mknod`를 통하여 OLT 장치를 등록한다. 그리고 사용자는 응용프로그램(`olt_app`)을 통하여 OLT 장치를 테스트할 수 있다.

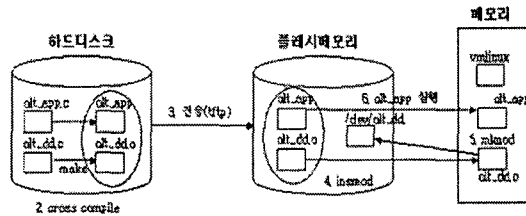


그림 10. OLT 디바이스 드라이버의 구동과정  
Fig. 10. Operation of OLT Device Driver

##### 4.1. 디바이스 드라이버의 구현

디바이스 드라이버는 장치 제어를 목적으로 커널 차원에서 제공되는 기능으로서 제어할 장치의 종류에 따라 블록 디바이스 드라이버, 네트워크 디바이스 드라이버, 문자 디바이스 드라이버로 구분되는데[7], 본 연구에서는 문자 디바이스 드라이버를 통하여 FPGA 보드를 제어하게 된다.

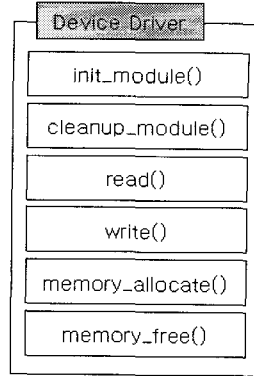


그림 11. 디바이스 드라이버의 주요 함수  
Fig. 11. Major Function of Device Driver

그림11은 구현된 디바이스 드라이버에서 사용된 주요 함수들을 나타낸 그림이다. OLT와 ONU에서 사용하는 입·출력 데이터의 크기는 48bits로 모두 동일하기 때문에 디바이스 드라이버 프로그램도 거의 동일하다. 작성된 디바이스 드라이버의 주요 함수들을 살펴보면 `init_module(void)`과 `cleanup_module(void)`는 모듈의 적재와 삭제를 위한 함수이고, `memory_allocate()`와 `memory_free()`는 ONU/OLT 모듈의 CPU 인터페이스 로직의 레지스터를 이용하기 위한 가상주소를 구하는 함수와 설정한 가상주소영역을 해제하는 함수이다. `write()`에서는 응용프로그램에서 보낸 REPORT 메시지를 `copy_from_user(&data, buf, length)`를 이용하여 디바이스 드라이버 측으로 가져와서 8bits씩 나누어 CPU 인터페이스 로직의 REPORT 레지스터에 할당된 가상주소 영역에 write 하게 된다. `read()` 역시도 해당하는 FPGA의 GATE 레지스터에 해당하는 주소 영역의 데이터를 8bits씩 읽어 와서 디바이스 드라이버 상의 변수에 저장하고 `copy_to_user(buf, &data, sizeof(data))`로 응용 프로그램 측으로 전송한다.

##### 4.2 응용 프로그램의 구현

RC-DBA 알고리즘 모듈을 검증하기 위한 전체적인 테스트 시스템의 구성은 그림 12와 같다. 좌측에는 두 대의 ONU 시스템, 가운데 허브는 ODN, 우측에는 RC-DBA 알고리즘 모듈이 OLT의 역할을 하면서 광케이블 대신 100Mbps UTP 이용하였다. PC2번과 3번에는 OLT측으로 REPORT 메시지를 전송하고 OLT에서 보내는 GATE 메시지를 수신하는 역할을 하는 TCP client 소

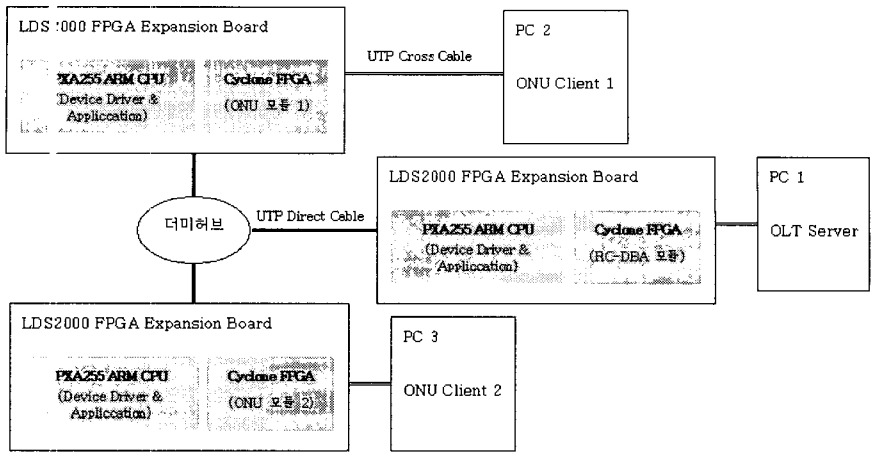


그림 12. 전체 테스트 시스템의 구성도  
Fig. 12. Structure of Testbed System

켓 프로그램이 동작한다. 두 개의 client 프로그램은 전적으로 동일한 동작 흐름을 갖는데, EPON에서는 TDMA 방식을 사용하기 때문에 이를 위해서 두 프로그램은 일정한 시간 차이를 두고 동작함으로써 두 대의 OLT 사이에 충돌을 방지한다.

4.3 검증결과

그림13은 ONU/OLT의 실제 테스트 시스템이다. 임베디드 보드와 FPGA 보드, 두 개의 보드가 하나의 시스템으로 구성된 LDS2000을 사용함으로써 실험 환경이 다소 간결해졌다. 검증용 응용 프로그램은 콘솔 기반의 프로그램으로서 호스트 linux PC에서 minicom을 통하여 동작을 확인할 수 있었다.

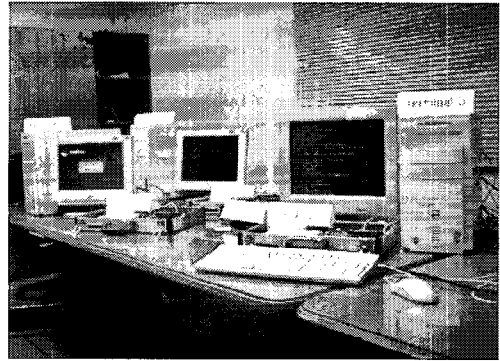


그림 13. ONU/OLT 검증시스템  
Fig. 13. Verification System of ONU/OLT

그림14는 minicom을 통하여 임베디드 시스템의 임베디드 리눅스를 모니터링하고 있다. 임베디드 리눅스 커널에서 RC-DBA 알고리즘 모듈을 사용하기 위하여 olt\_dd.o 모듈을 등록한다. olt\_dd 모듈의 major 번호는 insmod시 정해지지 않았으므로 커널이 임의로 253번을 부여했다. 커널 상에 적재된 olt\_dd.o가 제대로 등록되었는지 lsmod 명령으로 확인할 수 있다. 디바이스 드라이버를 파일로 등록하기 위하여 mknod 명령으로 /dev/olt\_dd로 등록하였다.

```

파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
bash-2.05ba#
bash-2.05ba#
bash-2.05ba# insmod olt_dd.o
Using init 0.0
Loading OLT Device Driver...
OLT Device driver registration OK with major number = 253
bash-2.05ba# lsmod
Module                Size  Used by
olt_dd                2982  0 (unused)
bash-2.05ba# mknod /dev/olt_dd c 253 0
bash-2.05ba# ls -l /dev/olt_dd
crw-rw-rw-  1 0          0      253,  0 /dev/olt_dd
bash-2.05ba# ./olt_app
OLT Open with major/minor (253 / 0)
OLT FPGA detected.

***** main menu *****
* n. write 5 report packets
* r. read 5 grant packets
*

```

그림 14. OLT모듈의 커널등록  
Fig. 14. Kernel Registration of OLT module

OLT의 응용프로그램은 크게 두 부분으로 나뉠 수 있는데 첫째는 ONU의 접속을 감지하고 연결하는 서버부분과 REPORT 메시지나 GATE 메시지의 송·수신을 담당하는 그림15, 16과 같은 응용 프로그램 부분이다. 응용 프로그램이 FPGA의 RC-DBA 알고리즘을 제어하고 있어도 OLT 서버는 여전히 다른 ONU 클라이언트의 접속을 감시하고 있는데, 이는 pthread를 사용했기 때문에 가능하다. 그림15와 같은 REPORT 메시지를 입력했을 때 RC-DBA 모듈은 그림16의 결과를 출력하였다.

GATE 메시지를 ONU측으로 브로드캐스트한 후, 프로그램을 종료하려면 'q' 명령을 사용하고, 응용 프로그램 종료와 함께 OLT 모듈도 커널에서 삭제시켜 주어야 한다. 그림 16에서 olt\_dd 모듈이 성공적으로 삭제되었다.

```

파월(F) 편집(E) 보기(V) 터미널(T) 기기(G) 도움말(H)
***** main menu *****
* w, write 5 report packets *
* r, read 5 grant packets *
* *
* 1, OLT's Scheduler run *
* q, FPGA quit *
*****

select the command number : w

Put 5 Report packets

R_reg0 = 5003 703 200
R_reg1 = 5403 704 200
R_reg2 = 5203 700 200
R_reg3 = 5503 700 0
R_reg4 = 5403 702 0
    
```

그림 15. REPORT 메시지 Write  
Fig. 15. REPORT Message

```

파월(F) 편집(E) 보기(V) 터미널(T) 기기(G) 도움말(H)
***** main menu *****
* w, write 5 report packets *
* r, read 5 grant packets *
* *
* 1, OLT's Scheduler run *
* q, FPGA quit *
*****

select the command number : r

Get 5 Grant packets

Grant0 = 0x5 02 320
Grant1 = 0x15 02 410
Grant2 = 0x25 02 000
Grant3 = 0x35 02 000
Grant4 = 0x45 02 200
    
```

그림 16. GATE 메시지 Read  
Fig. 16. GATE Message

## V. 결 론

세계 각국은 FTTH를 가입자망에 도입하기 위하여 다양한 형태의 PON에 대해 연구개발을 진행하고 있다. 특히 EPON은 LAN의 85% 이상이 이더넷으로 구성되어 있다는 점에서 광가입자망으로는 최적의 네트워크라고 할 수 있다.

DBA 기술은 EPON 망의 성능에 영향을 주는 주요 요인으로서 실제 EPON 개발업체들은 고유의 방식으로 하드웨어, 또는 임베디드 프로세서 기반의 소프트웨어 방식으로 DBA를 구현하고 있다. 본 논문에서는 RC-DBA 알고리즘을 적용한 OLT 내부의 동적대역할당 스케줄러를 HDL언어로 설계하여 Altera사의 Cyclone FPGA 칩에 구현하였다. 그리고 구현된 RC-DBA 알고리즘의 동작검증을 위하여 간단한 ONU 시스템을 개발하고 두 시스템을 TCP통신으로 연결하여 RC-DBA 알고리즘에서 대역폭 할당이 어떻게 이루어지는지 확인해 보았다. 검증시스템은 임베디드 리눅스 기반의 디바이스 드라이버와 응용 프로그램 등으로 제어하였다.

## 참고문헌

- [1] "광 네트워크", IITA 주간기술동향 통권 1264호, 2006.9.20, p45-59
- [2] 유태환, "EPON 기술동향", 텔레콤, 제 20권1호, p36-54, 2004
- [3] 장성호, 장종욱, "EPON망에서 ONU 공정성을 고려한 RC-DBA 알고리즘의 설계와 성능평가", 동의대학교 대학원 박사학위논문, 2005.2
- [4] 802.3 draft문서 - MPCP Message Format [http://www.ieee802.org/3/efm/baseline/hirth\\_1\\_0302.pdf](http://www.ieee802.org/3/efm/baseline/hirth_1_0302.pdf)
- [5] Michael P. McGarry, Martin Maier, and Martin Reisslein, "Ethernet PONs: A Survey of Dynamic Bandwidth Allocation Algorithms", [http://www.fulton.asu.edu/~mre/DBA\\_survey.pdf](http://www.fulton.asu.edu/~mre/DBA_survey.pdf)
- [6] IEEE802.1d Draft Document for 802.1d, May. 2001
- [7] Corebell Co., "The practical use of the Linux Device Driver", p105-140, 2003
- [8] S. Choi and J. Huh, "Dynamic Bandwidth Allocation Algorithm for Multimedia Services over Ethernet PONs", ETRI Journal, Vol. 24, No. 6, pp. 465-468, Dec. 2002



### 저자소개



**장 종 욱(Jong-wook Jang)**

1987~1995년 한국전자통신  
연구소 통신연구단 연구원  
1995년 부산대학교 컴퓨터공  
학과 박사

1999년 미주리주립대 Visiting Scholar(Post.Doc)

1995.3~현재 동의대학교 컴퓨터공학과 교수

※관심분야: 모바일 MAC 프로토콜, 이동성 관리



**강 현 진(Hyun-jin Kang)**

2007년 2월 동의대학교 컴퓨터공학과  
석사

2006.9~ 현재 IKNN 근무

※관심분야: 광통신, 임베디드 시스템