論文 91-28B-5-9

Mini-MAP을 채택한 토큰-패싱 버스 네트워크의 MAC계층 구현에 관한 연구

(A Study on the Implementation of MAC Layer of Token-passing Bus Network Based on Mini-MAP)

姜 文 植* 趙 秉 宣** 朴 玟 用* 李 相 培*

(Moon Sik Kang, Byung Sun Cho, Mignon Park, and Sang Bae Lee)

要 約

본 논문에서는 Mini-MAP을 채택한 토큰-패싱 버스 네트워크의 망접속기와 구동소프트웨어를 구현하였다. 망접속기는 프레임의 송, 수신 및 논리적 링(logical ring)을 유지, 관리하는 MAC(media access control)계층의 기능을, 구동 소프트웨어는 프레임의 송, 수신을 위한 버퍼의 관리 및 에러가 발생하였을 때 이를 상위 계층에 보고하는 등의 기능을 담당한다. 망접속기의 TBC(token bus controller)로는 Motolora의 MC68824를 사용하였으며, CPU는 Intel의 80186을 사용하였다.

제작된 망접속기는 자기진단 테스트를 통해 호스트, 망접속기의 CPU, TBC 사이의 데이타 전송등의 동작을 확인하였으며, 구동 소프트웨어는 각 모듈 테스트를 통해 프레임의 송, 수신시 필요한 여러 기능들의 동작여부를 확인하였다.

Abstract

In this paper, a network interface unit of token-passing bus network and its driving software are implemented based on Mini-MAP. This network interface unit performs the function of MAC layer which is responsible for transmission and reception of frames as well as the management of logical ring. The driving software performs the management of data buffer and the report of errors, if ocoured, to the higher layer. Motorola MC68824 of is used as a TBC (Token Bus Controller) and Intel 80186 as a CPU for network interface unit.

The operation of network interface unit is verified by self-test which checks the functioning of TBC and CPU. In addition each module of driving software is tested to check the functions regarding transmission and reception of frames.

*正會員, 延世大學校 電子工學科
(Dept. of Elec. Eng., Yonsei Univ.)
**正會員, 韓國通信
(Korea Telecom.)
接受日字 : 1990年 12月 30日

I. 서 론

생산규모의 확대로 인한 프로그래머블 디바이스 사이의 정보 공유 및 각 공정들 사이의 자원 공유가 요구되면서 이를 지원할 수 있는 생산 자동화 전용 네트워크에 대한 필요성에 의해 미국의 GM(general

motors)사는 자사의 프로그래머블 디바이스를 접속하기 위한 목적으로 전용 네트워크 프로토콜의 연구를 시작하여, 1985년 LAN(local area network) 의 성격을 갖는 MAP(manufacturing automation Protocol)을 발표하였다.[1][6]

MAP은 기존의 거의 모든 통신 네트워크가 채택하고 있는 OSI(open systems interconnection)7 계층 구조를 따르고 있어서 타 네트워크와의 폭넓은 접속 호환성이 있으며, MAP의 하위 계층은 토큰-패싱버스 방식을 채택하고 있어서 잡음이 많이 존재하는 생산환경에서의 동작을 가능케 하고, 네트워크 스테이션의 추가 및 갱신의 용이성을 제공한다.[2]

그러나 OSI의 7계층 구조를 모두 포함하는 Full-MAP은 설치 비용이 많이 들 뿐만 아니라 공장자동화 네트워크의 가장 중요한 요소중의 하나인 실시간 처리에도 어려움이 있어· 그 대안으로 제시된 것이 Mini-MAP이다. Mini-MAP은 1, 2, 7계층만이 있고 중간 계층이 생략된 구조로써 Full-MAP에 비해 기능은 떨어지지만 설치 비용이 적고 실시간 처리가 용이하다는 장점이 있다. Mini-MAP의 계층 구조에서 제 2 계층인 DLC(data link control)계층의 부계층(sublayer)인 MAC(media access control)계층은 논리적 링(logical ring)의 구성, 유지 및 송, 수신 프레임 관리등의 기능을 담당한다.[4][9]

본 논문에서는 Motorola의 MC68824 TBC(token bus controller)를 사용하여 토큰-패싱 버스 네트워크의 MAC계층의 기능을 담당하는 망접속기를 제작하고 이 접속기의 자기진단 테스트 프로그램과 구동 프로그램을 구현하였다.

Ⅱ. 토큰-패싱 버스 프로토콜

토큰-패싱 버스 프로토콜은 IEEE에 의해 채택된 근거리 통신망 표준중의 하나이다. 이는 스테이션을 연결시키는 물리적 버스인 전송매체를 액세스하는 MAC계층의 기능에 따라 분류된다. MAC계층의 주요기능은 전송매체를 차례로 액세스할 수 있도록 관리하고, 스테이션의 이탈 및 추가 그리고 고장을 회복시키는 논리적 링의 유지이다. MAC계층에서 프로토콜의 특성상 발생되는 장애의 종류는 크게 이중 토큰, 토큰의 분실, 토큰의 전달시 고장 발생, 무응답 스테이션 발생, 스테이션의 이중번지 등으로 분류된다.[4]

논리적 링을 유지한다는 것은 크게 스테이션의 추가 및 이탈 그리고 고장 회복으로 나눈다. 스테이션의 이탈은 자신의 PS에게 NS의 번지를 변화시키도록 함으로써 간단히 이루어질 수가 있다. 그러나 추가의 경우 한개의 스테이션이라면 문제가 없으나 여러 스테이션이 동시에 추가를 요청하여 올경우는 통제 가능한 경쟁(controlled contension)에 의하여 해결을 해야한다. 통제 가능한 경쟁에서는 최상위 번지를 갖는 스테이션이 가장 높은 우선순위가 주어지게 된다.

다른 프로토콜과 비교할 때 토큰-패싱 프로토콜이 갖는 장점은 스테이션을 논리적 링 속으로 추가하거나 이탈할 때 다른 스테이션에 영향을 주지않고 자유로이 이루어질 수가 있다는 것이다.

모든 스테이션은 토큰을 받으면 전송준비가 되어 있는 패킷을 전송하고 Max_Inter_Solicit_Count의 값을 체크한 후 응답 윈도우(response window)를 열어 자 의 다음 스테이션이 되기를 원하는 스테이션에게 논리적 링에 가입할 기회를 제공한다. Max_Inter Solicit_Count의 값은 2^1-2^8범위내에 있으며 링 관리 타이머(ring maintenance timer)와 함께 얼마나 많은 윈도우를 열 것인가를 결정하는데 사용된다.

만약 링 관리 타이머의 값이 S라고 가정하면 토큰을 S번 보내기 전에 최소한 한번은 Solicit_Successsor를 한다는 것이다. 이 값은 최소한 50ms마다 혹은 Solicit_Successor를 할 때마다 갱신되며 각 스테이션에서의 값은 동일하지 않다.

토큰-패싱 방법이 가지고 있는 여러가지 특성 중에서도 몇가지의 특징만을 살펴보면 다음과 같다.[4]

- 네트워크에는 축소된 기능을 갖는(수신기능만 있음)스테이션이 존재할 수도 있으며 적어도 하나의 완전한 기능을 갖는 스테이션만 있으면 된다.
- 스테이션이 전송매체를 사용하는 방식에는 거의 제한이 없다. 특히 어떤 스테이션이든지 특별한 액세스방식(polling/response)을 자기 차례에 사용할 수 있다.
- 각 스테이션이 전송매체의 용량을 균등하게 점유한다는 점에서 공정성이 유지되고 다양한 서비스를 제공할 수 있다.
- 트래픽이 많은 경우에도 액세스 방식의 overhead가 전송매체의 일부분만을 차지한다는 점에서 효율적이다.
- 프로토콜상에서 발생되는 통제가능한 간섭을 구별할 수 있으므로 잡음 측정이 가능하다.

Ⅲ. MAC 계층[4][8]

MAC계층은 전송매체의 액세스를 통제하고 논리적 링을 구성하거나 유지하는 주요 기능외에 상위계층과 하위 계층간 접속 역할로 전송 및 수신 데이타

를 보관하는 기능을 한다. 또한 스테이션을 관리하기 위해 스테이션 관리 entity와 서로 협조하여 MAC의 초기화, MAC에서 사용되는 변수의 값을 읽거나 지정 거리고 MAC에서 발생한 중요 변동사항 및 고장을 통보하며, 그 내부를 구성하신 머신(machine)의 주요기능을 살펴보면 다음과 같다.

IFM(interface machine)은 LLC와 MAC계층간 그리고 스테이션 관리 entity와 MAC계층관 접속과 버퍼 역할을 하고 수신된 MA_Data를 해석하여 전송할 서비스 프리미티브(primitive)를 발생시키고 버퍼로써 요청된 서비스를 차례로 저장된다.

ACM(access control machine)은 MAC계층의 주요부분으로서 토큰을 취급하기 위하여 다른 스테이션의 ACM과 상호 협조하고 LLC계층에게 다른 레벨의 서비스의 질(service quality)을 제공하기 위하여 여러가지의 MAC 액세스 클래스를 관리한다. 그리고 주 임무인 논리적 링의 구성 및 유지와 고장탐지 및 회복을 수행한다. ACM은 MAC계층 내에서 가장 복잡하고 주요한 기능을 수행하며 내부적으로는 11개의 상태를 갖는다.

RxM(receive machine)은 물리 계층에서 온 Atomic_Symbol을 프레임으로 조합하여 IFM과 ACM으로 보내고 Noise_Burst 혹은 Bus_Quiet를 수신한다.

TxM(transmit machine)은 ACM에서온 데이타 프레임을 받아 preamble, SD(start delimiter), FCS(frame check sequence), ED(end delimiter)를 추가하여 MAC PDU(protocol data unit)를 만든다.

RRM(regenerative repeater machine)은 물리 계층으로부터 오는 모든 Atomic_Symbol 열을 받아 반복 전송시키며 head-end remodulator와 같은 특별한 재생기(repeater)가 있는 스테이션에만 있다.

Ⅳ. 망접속기의 구성

1. 망접속기의 하드웨어[9]

토큰-패싱 버스 네트워크의 MAC계층의 기능을 담당하는 망접속기의 TBC로는 Motolora의 MC68824를 사용하였으며, CPU는 칩 선택 제어기(chip select controller), DMA 제어기, 인터럽트 제어기를 내장하여 하드웨어의 부담을 줄일 수 있는 intel의 80186을 사용하였다. 80186 CPU와 MC68824 TBC는 공유 메모리를 통해서 데이타를 송·수신하며 private ROM에는 망접속기 구동 프로그램이 들어가며, private RAM은 인터럽트 벡터테이블, 변수영역과 스택 영역으로 사용된다.

TBC의 제어를 받는 MC68194 CBM(carrier band MODEM)은 5Mbps의 속도로 데이타를 전송하고, 수신되는 데이타를 TBC로 보낸다. 데이타 전송을 위해 80186 CPU는 인터페이스 회로를 통해서 호스트에 접속된다.

(1) CPU-TBC 인터페이스

80186과 TBC사이의 인터페이스는 그림 1과 같다. 버스 제어기에 의해서 CPU와 TBC는 서로 충돌을 일으키지 않고 버스를 공유하여 shared RAM에 억세스할 수 있게된다.

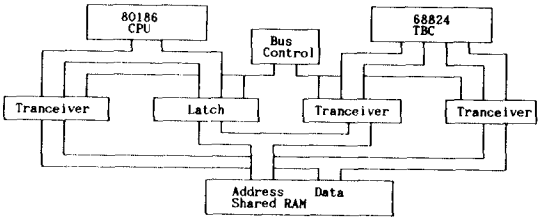


그림 1. CPU-TBC 인터페이스
Fig. 1. CPU-TBC interface.

(2) 호스트와 80186 사이의 인터페이스

호스트와 80186사이의 인터페이스는 그림2와 같다. 그림에서 비교기와 디코더는 호스트와 망접속기 사이의 인터럽트를 위하여 사용되며, 트랜시버와 그 밖의 논리회로는 DMA를 이용한 데이타 전송에 사용된다.

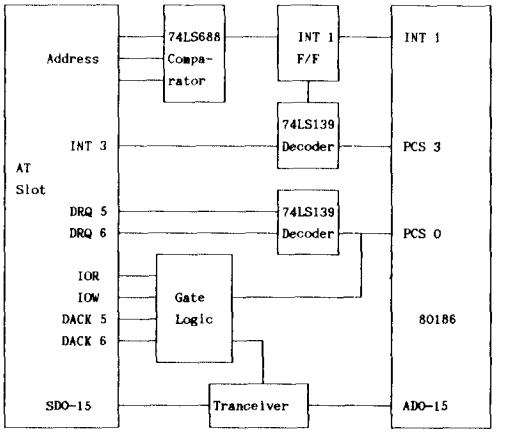
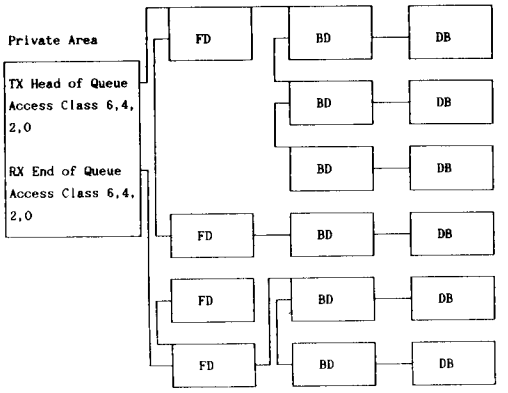


그림 2. 호스트-80186 인터페이스
Fig. 2. Host-80186 interface.

2. 망접속기의 구동 소프트웨어

(1) TBC의 버퍼 구조

TBC의 버퍼 구조는 다음 그림과 같은 fully-linked 구조를 갖는다. TBC의 private area내의 각 클래스별 TX head of queue pointer는 송신 프레임의 선두 FD(frame descriptor)를 가리키며, 각 클래스별 RX end of queue pointer는 수신된 마지막 프레임의 FD를 가리킨다. Head of queue pointer는 호스트에 의해서, End of queue pointer는 TBC에 의해서 갱신된다.



FD : Freme Descriptor, BD : Buffer Descriptor, DB : Data Buffer.

그림 3. TBC의 송, 수신 큐의 구조
Fig. 3. Tranceiver, receiver queue structure of TBC.

(2) 구동 소프트웨어

구동프로그램의 전체 개요는 다음과 같다.

- 초기화
  - 80186 CPU와 TBC의 초기화 : 칩 선택 제어기, DMA 제어기, 인터럽트 제어기의 각 레지스터의 값을 설정하고 TBC의 'INITIALIZE' 명령을 사용하여 초기화 테이블의 각 파라메터의 값으로 private area를 설정한다.
  - Memory와 TBC test
  - 송, 수신을 위한 free FD, BD pool의 초기화
  - 각 flag의 초기화
- 프레임의 전송
  - 호스트에 의해 송신 프레임이 발생했을 경우, 호스트는 망접속기로 인터럽트 1을 발생시키고 망접속기는 INT1_FLAG를 1로 설정한다.
  - INT1_FLAG가 1로 되면 망접속기는 priority, 프레임의 갯수, 각 프레임의 길이, 각 프레임의 MAC destination address 등으로 이루어 프레임 해더를 호스트로부터 가져와, Free FD, BD Pool로 부터 송신에 필요한 만큼의 FD와 BD를 만든 후 DMA 0를 구동시켜 프레임의 데이타를 각 BD의 데이타 버퍼에 배열한다.
  - 선택된 priority의 송신 큐에 이미 배열되어 있던 프레임중 전송이 완료된 프레임의 FD와 BD를 free pool로 환원하고 전송이 실패한 프레임이 있으면 이를 호스트에 보고한다.
  - 선택된 priority의 송신 큐의 상태를 판별한다.
  - INT1_FLAG을 0으로 초기화한다.
  - 각 클래스의 송신 큐에 대해 망접속기의 CPU는, TBC의 송신 여부를 체크하기 위하여 첫번째 FD에 대한 포인터와 새로운 FD의 추가를 위해 마지막 FD에 대한 포인터를 유지해야 한다.
- 프레임의 수신
  - TBC의 private area에 있는 각 클래스별 RX end of queue pointer가 가르키는 수신 큐의 마지막 FD의 confirmation/indication word 내의 NRV(next receive pointer valid)비트는 새로운 프레임이 수신되었을때 1로 된다. 따라서 NRV가 1로 되면 망접속기는 DMA 1을 구동시켜 수신 프레임을 호스트로 전송한 다음 호스트로 인터럽트 3을 걸어 프레임의 수신을 알린다.
  - 수신 프레임의 FD내에 있는 수신 상태 워드(receive status word)를 호스트로 전송하여 수신 프레임의 에러 여부를 호스트에 알린다.
  - 호스트로 전송이 끝난 수신 프레임의 FD와 BD의 각 control word와 포인터를 재설정한 다음 이를 free pool로 환원한다.
  - 각 클래스의 수신 큐에 대해 망접속기의 CPU는, 수신된 프레임을 호스트로 전송하기 위하여 수신 큐의 첫번째 FD에 대한 포인터를 유지해야 한다.

V. 모듈테스트 및 결과 고찰

MAP 망접속기의 테스트는 다음과 같은 하드웨어의 자기 진단 테스터와 버퍼 관리프로그램의 모듈테스트를 통해 망접속기의 동작여부를 검증하였다.

1. 자기 진단 테스트

TBC를 테스트하기 위해 TBC에는 다음과 같은 4가지의 테스트를 위한 커맨드가 준비되어 있어 이

를 통해 TBC의 작동을 테스트할 수 있다. 아래의 테스트에서 사용된 초기화 테이블(initialization table)은 그림 4 와 같다.

```
2000 :  0000 FFFF 2100 0000 05FF 0000 0000 03FF
2010 :  0000 3FFF 0000 03FF 0000 5FFF 0000 0000
2020 :  0000 0000 0020 FC00 4000 FFFF 2180 0000
2030 :  C000 0000 0000 0000 C000 0000 0000 0000
2040 :  C000 0000 0000 0000 FFFF 21EC 0000 0000
2050 :  0000 0000 0000 0000 FFFF 2220 FFFF 2244
2060 :  0000 0000 0000 FFF2 0000 0000 0020 0007
2070 :  3F00 FF00 0000 0000 0FAA 8000 0000 0000
2080 :  0000 0000 0000 0000 0000 0000 0000 0000
2090 :  0000 0000 0000 0000 0000 0000 0000 0000
```

그림 4. 초기화 테이블
Fig. 4. Initialization table.

2. 호스트 인터페이스 테스트(host interface test)

이 테스트에서는 공유메모리의 버퍼에서 TBC의 FIFO로, 다시 버퍼로의 경로를 체크한다. 테스트 방법은 호스트가 34바이트의 데이타를 임의로 준비하고, 테스트 완료 후 동일한 데이타가 연속하여 TBC에 의해 버퍼에 쓰여지게 된다. 제작된 망접속기의 테스트에서는 그림 5 와 같은 결과를 얻었다. 그림에서 보듯이 임의로 주어진 34바이트의 데이타가 연속하여 옮겨져 있음을 알 수 있다.

```
2180 :  0001 0203 0405 0607 0809 0A0B 0C0D 0E0F
2190 :  0001 0203 0405 0607 0809 0A0B 0C0D 0E0F
21A0 :  0001 0001 0203 0405 0607 0809 0A0B 0C0D
21B0 :  0E0F 0001 0203 0405 0607 0809 0A0B 0C0D
21C0 :  0E0F 0001 0000 0000 0000 0000 0000 0000
```

그림 5. 호스트 인터페이스 테스트의 결과
Fig. 5. A result of host interface test.

3. 쌍방향 루프 백 테스트(full duplex loop back test)

이 테스트는 TBC의 송수기와 수신기가 동시에 동작(full duplex)하는가를 체크한다. 이 테스트에는 internal loop back과 모뎀을 통한 external loop back 이 있으며, 이 실험에서는 internal loop back 모드를 사용하여 TBC의 송,수신기가 동시에 동작함을 확인하였다.

4. 전송 테스트(transmitter test)

이 테스트는 메모리의 송신 버퍼에서 TBC의 송신기로, 다시 TBC의 수신기로의 경로를 테스트한다. 이 테스트 역시 internal loop back과 external loop back이 있다. 테스트 방법은 실제의 프레임을 전송하고 이를 수신기가 되받는다. 이때 송신프레임의 FD와 초기화 테이블의 CPA(command parameter area)를 조사함으로써 CRC 에러와 underrun 에러, 그리고 송신 프레임의 confirm 여부를 체크한다. 이 테스트에서는 TBC가 In_Ring 상태에 있는 것처럼 동작하므로 High_Priority_Token_Hold_Timer를 'FFFFH'로 세트하여 최대의 전송시간을 보장해 준다.

이 실험에서는 internal loop back 모드를 사용하였으며 전송 프레임의 한 FD를 보이면 그림 6 과 같다. FD의 첫번째 워드가 confirmation 워드이며 이의 첫번째 비트가 CFD 비트이다. 그림에서 보면 프레임의 전송 후 CFD 비트가 1로 세트되어 있어 TBC가 프레임의 전송에 성공했음을 알 수 있다. 본 실험에서는 초기화 테이블의 CPA 조사하여 CRC 에러, underrun 에러가 발생하지 않음도 확인하였다.

```
2180 :  A000 0000 0000 FFFF 0000 0000 FFFF 21A4
2190 :  0022 0000 0000 0000 0000 0062 F0FF A0FF
21A0 :  0000 0000 0000 FFFF 21B6 8000 0022 0000
```

그림 6. 전송 테스트 후의 FD
Fig. 6. FD after transmitter test.

5. 수신 테스트(receiver test)

이 테스트에서는 수신 버퍼에 TBC의 수신기 사이의 경로를 체크한다. 테스트 방법은 호스트에 의해 준비된 데이타 패턴을 TBC의 송신기가 전송하고 이를 수신기가 받아서, 수신된 프레임의 여러 여부를 체크한다. 프레임은 SD(start delimeter)와 FCS(frame ckeck sequence) 사이에 전송되며 수신 프레임의 에러 여부는 호스트에 의해 미리 준비된 수신 버퍼의 FD의 수신 상태 워드(receive status word)에 기록된다. FD의 두번째 워드인 수신상태 워드는 그림 7 과 같으며 프레임의 각 에러에 대해 해당비트가 1로 된다.

테스트 완료후, 한 수신프레임의 FD는 그림 8 과

CRCE-CRC error occured in the frame received
EBIT-The E bit in the END Delimeter is set
FOVE-FIFO Overflow
FTL-Frame too long(the received frame is longer than 8K bytes)
NOBUF-Not enough buffers in buffer pool
*-Reserved

그림 7. 수신 상태 워드
Fig. 7. Received status word.

```
2220 :  0000 0000 0000 FFFF 0000 0000 FFFF 2244
2230 :  0021 0000 0000 0000 0000 0062 F0F0 F0F0
2240 :  0000 0000 0000 0000 0000 0000 0000 0000
```

그림 8. 수신 테스트 후의 FD
Fig. 8. FD after receiver test.

같다. 수신상태워드가 0으로 되어 있어 수신 프레임에 에러가 없음을 알 수 있다.

6. 구동 소프트웨어의 모듈테스트

망접속기의 구동 프로그램의 주요루틴은 다음과 같다.

- 송, 수신 buffer의 FD, BD pool의 초기화 루틴
- 망접속기와 호스트 사이의 DMA루틴
- 송신 프레임의 FD, BD를 만드는 루틴
- 송신 큐의 큐 상태 판별 루틴
- 송, 수신이 완료된 프레임의 FD, BD를 free FD, BD로 환원하는 루틴

위의 각각 루틴은 실행후 buffer와 프로그램에서 사용하는 각 address pointer를 조사하여 바르게 실행됨을 확인하였으며, 그중 송신프레임의 FD, BD를 만드는 루틴을 실행한 결과를 보인다.

FD, BD pool에 대한 세그먼트 address는 1D96H로, data buffer address는 0000H로 설정하였다. 송신 buffer를 초기화한 후에 이 루틴에 의해 생성된 3개의 FD는 그림 9와 같다.

그림 9에서 FD의 각 포인터, MAC address, frame data length, frame control, 그리고 각 제어워드 등이 바르게 설정되어 있음을 알 수 있다. 위와 같은 FD에 대해 BD는 그림10에 나타나 있다.

```
0000 : 0000 0000 8000 FFFF 0024 0000 FFFF 00B4
0010 : 0034 0000 0000 FFFF 0000 00C2 A0FF D0FF
0020 : 0000 0000 0000 0000 8000 FFFF 0048 0000
0030 : FFFF 00D8 000C 0000 0000 FFFF 0000 00C2
0040 : B0FF D0FF 0000 0000 0000 0000 0000 FFFF
0050 : 006C 0000 FFFF 00EA 0078 0000 0000 FFFF
0060 : 0000 00C2 C0FF D0FF 0000 0000 0000 0000
```

그림 9. 테스트 루틴에 의한 FD
Fig. 9. FD according to test routine.

```
00B0 : 0000 0000 0000 FFFF 0000 0000 0032 0000
00C0 : 0000 FFFF 00C6 0000 FFFF 0032 8000 0002
00D0 : 0000 0000 FFFF 00D8 0000 FFFF 0064 8000
00E0 : 000C 0000 0000 FFFF 00EA 0000 FFFF 0096
00F0 : 0000 0032 0000 0000 FFFF 00FC 0000 FFFF
0100 : 00C8 0000 0032 0000 0000 FFFF 010E 0000
0110 : FFFF 00FA 8000 0014 0000 0000 FFFF 0120
```

그림10. 테스트 루틴에 의한 BD
Fig. 10. BD according to test routine.

그림10에서와 같이 주어진 3개의 프레임에 필요한 5개의 BD가 생성되었으며, BD에 대해 각 포인터, 제어워드, 버퍼길이 등이 바르게 설정되어 있음을 확인할 수 있다.

V. 결 론

본 논문에서는 Mini-MAP의 계층구조중 2계층의 부계층인 MAC계층의 기능을 담당하는 망접속기와 이의 구동 소프트웨어를 제작하였다. 망접속기의 TBC는 motorola의 MC68824를 사용하였으며, CPU는 타이머, 인터럽트 콘트롤러, DMA 콘트롤러등을 내장하여 하드웨어를 간소화할 수 있는 intel의 80186을 사용하였다. CPU와 TBC는 공유 메모리를 통하여 데이타를 송, 수신하며 망접속기의 private RAM은 인터럽트 벡터 테이블, 변수영역, 스택영역등으로 사용된다.

망접속기의 ROM에는 프레임의 송, 수신 및 버퍼의 관리를 담당하며, 프레임의 송, 수신시 에러가 발생하면 이를 호스트에 보고하는등의 기능을 갖는 구동 소프트웨어를 장착하였다.

제작된 망접속기는 메모리 테스트, 호스트 인터페

이스 테스트, 쌍방향 루프백 테스트, 송, 수신 테스트등의 자기진단 테스트를 통하여 바르게 동작함을 확인하였으며, 구동 소프트웨어 또한 각 기능별 모듈테스트로부터 동작여부를 확인하였다.

參 考 文 獻

[1] Vinoent C. Jones, "MAP/TOP networking: a foundation for computer integrated manufacturing," New York: McGraw-Hill, 1987.

[2] L.J. MoGuffin, L.O. Reid and S.R. Spark, "MAP/TOP in CIM destributed computing," *IEEE Network,* vol. 2, no. 3, pp. 23-30, May. 1988.

[3] William Stallings, *Handbook of Computer Communications Standards* vol. 2, Macmillian Publishing Company, New York, 1987.

[4] IEEE, *Token-Passing Bus Access Method and Physical Layer Specification,* John Wiley & Sons, Inc., 1984.

[5] Alfred C. Weaver and David W. Butler, "A fsult-tolerant network protocol for real-time communication," *IEEE Transactions on Industrial Electronics,* vol. IE-33, no. 3, August, 1986.

[6] Alfred C. Weaver and Catherine F. Summers, "The IEEE Token bus-A performance bound on GM MAP," *IEEE Transactionson Industrial Electronics,* vol. 35, no. 1, February, 1988.

[7] R.A. Dirvin, A.R. Miller, "The MC68824 token bus controller VLSI for the factory LAN," *IEEE Micro,* p. 51-55, June, 1986.

[8] ANSI/IEEE Std. 802.2-1985, Logical Link Control.

[9] Motorola, *MC68824 Token Bus Controller User's Manual,* 1989.

[10] GM Task Force, *Manufacturing Automation Protocol Specification version* S.O. GM, 1987.

[11] IEEE, *IEEE Standards for Local, Area Networks: Logical Link Control,* John Wiley & Sons, Inc., 1984.

[12] Asok Ray, "Networking for Computer-Integrated Manufacturing," *IEEE Network,* vol. 2, no. 3, May, 1988.

著 者 紹 介

趙 秉 宣(正會員)
1965年 9月 30日生. 1988年 3月 연세대학교 전자공학과 졸업.1990年 8月 연세대학교 전자공학과 대학원 졸업. 석사학위 취득. 1991年 1月~현재 한국통신 전임연구원. 주관심분야는 컴퓨터 네트워크 등임.

●

李 相 培 (正會員) 第27卷 第3號 參照
현재 연세대학교 전자공학과 교수

姜 文 植 (正會員) 第27卷 第2號 參照
현재 연세대학교 전자공학과 박사과정

●

朴 玟 用 (正會員) 第27卷 第3號 參照
현재 연세대학교 전자공학과 부교수