

MAP 적용을 위한 토큰 패싱 버스 프로토콜의 구현

*이창원 이강익 신기명 김용득

아주대학교 공과대학 전자공학과

The implementation of token passing bus protocol for MAP application

*Chang-Won Lee Kang-Ik Lee Gi-Myung Shin Yong-Deuk Kim

Dept. of Electronics Eng. AJOU Univ.

ABSTRACT

This paper deals with the implementaion of the IEEE 802.4 token passing bus protocol which is the world standard for LAN protocol. we are design station adaptor using MC68824 for data link layer, MC68194 for physical layer, and implemented by IEEE 802.4 and 802.2 type 3. We made three station adaptor and its performance is experimented.

1. 서론

IEEE 802.4 프로토콜은 여러 독자적인 스테이션이 공유하는 하나의 전송 매체를 점해진 순서에 따라 사용하도록 설계되어 있다. 이 프로토콜은 정보의 실시간 처리를 요하는 공장 환경의 통신에 적합하다. 공장 환경에서의 통신은 다양한 자동기기, 제어기기들 사이에 짧은 매세지들이 빈번하게 전송되고, 엄격한 응답 시간을 요한다. 또한 버스형태의 망 구성 방식은 공장의 생산 라인 형태에 잘 적용될 수 있으며 Layout의 변경 확장에 적응성이 높다. 이러한 이유로 토큰 패싱 버스 방식은 GM 사의 MAP 규격안의 physical, data link 계층의 프로토콜로 채택되었다.

본 연구에서는 IEEE 802.4 토큰 버스 프로토콜을 호스트들 IBM-PC로한 접속기를 설계하여 구현하고, Application으로는 간단한 파일 전송과 네트워크 상태를 모니터 할 수있는 운용 프로그램을 올려 실험하고, 성능 분석된 자료들과 비교, 분석하여 이후 MAP으로의 접근에 발판이 되고자 하였다.

2. 토큰 패싱 버스 프로토콜

토큰 패싱 버스 프로토콜은 IEEE 에서 채택한 LAN 표준 프로토콜중의 하나로 여러 스테이션이 버스의 형태로 하나의 라인에 물려있고 논리적으로 링을 구성하여 통신 서비스

를 행한다. 전체 프로토콜은 정상 상태의 동작과 링 초기화 과정과 링을 구성한 스테이션들의 관리에 대한 동작으로 이루어진다. 정상상태의 동작은 각 스테이션들이 논리적으로 링을 구성하고 있으면서 토큰이라는 제어 프레임이 어드레스가 감소하는 순서에 따라 다음 스테이션으로 넘겨준다. 토큰을 받은 스테이션은 모든 데이터를 전송한 경우나 전송할 데이터가 없는 경우, 스테이션의 타이머가 off 된 경우 토큰을 다음 스테이션으로 넘겨주게된다. 정상상태 동작은 데이터 전송 절차와 토큰 전송 절차를 번갈아 가면서 하게 된다. 이외에 초기에 링을 구성하고 유지하는 것과 여러 보정에 대한 복잡한 기능을 수행한다.

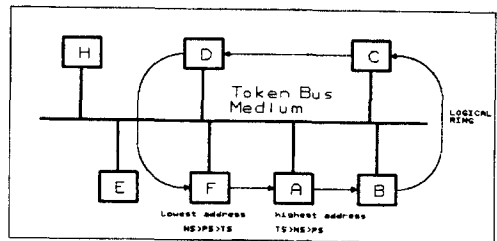


그림 1. 토큰 버스상에 구성된 링

3. 접속기의 하드웨어 구조

본 논문에서 설계한 토큰 버스 스테이션 접속기는 구조상 크게 두부분으로 나누어진다. 이것은 MAP의 계층에 따른 구분과 일치하는데 응용계층은 IBM-PC에 소프트웨어로 구현되며, 물리 계층과 데이터링크 계층은 네트워크 접속기의 보드상에 전용 LSI와 프로세서로 구현되었다. 네트워크 접속기는 다시 물리계층의 모뎀(MODEM)부와 데이터 링크계층으로 구현되었다.

모뎀부분은 주로 아날로그 소자로 구성되며 데이터 링크 계층과는 표준 접속방식에 따라 접속할 수 있어야 한다. 본 논문에서는 5Mbps의 전송속도를 갖는 위상응집 FSK변조

방식의 모델을 설계하였다. 모델은 MC68194 캐리어밴드 모델 소자를 중심으로 설계하였다.

데이터 링크계층은 두개의 부속계층(LLC와 MAC계층)으로 분리되는데 MAC계층은 토큰버스 제어기(MC68824)에 의해 하드웨어로 구현하였으며 LLC계층과 MAC계층의 관리를 위한 로컬 프로세서로 i80186을 사용하였다.

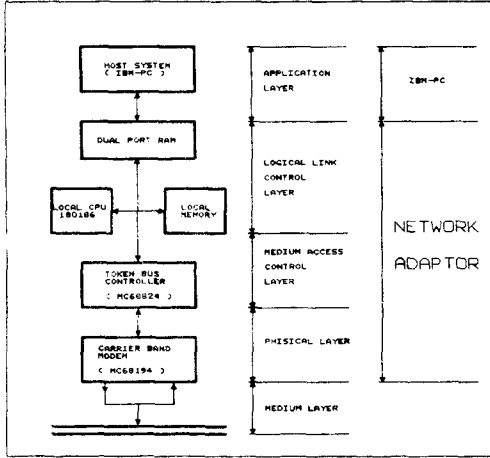


그림 2. 토큰 버스 스테이션의 구조

네트워크접속기는 IBM-PC slot에 내장할 수 있도록 설계 되었으며, 다음과 같은 기능을 갖도록 설계하였다.

- IEEE 802.4 PHY 규격의 5Mbps 캐리어밴드 모델 회로
- IEEE 802.4 MAC 규격의 토큰 버스 관리 기능
- IEEE 802.2 LLC type 1 규격의 데이터 링크 기능
- 계층 1,2 관리 기능
- IBM-PC BUS 와 접속 기능

그림3.에 네트워크접속기의 구조를 보았다. IBM-PC 와의 정보 전송 효율을 높이기 위해 접속 회로에 듀얼 포트 메모리가 사용 되었으며 CPU는 인텔의 80186 16 bit 프로세서가 사용 되었고, 토큰관리를 위해 모토로라의 TBC(Token Bus Controller:MC68824) 가 사용되었다. 그리고 싱글 칩 모델을 사용하여 모델부의 회로를 간단히 하였다.

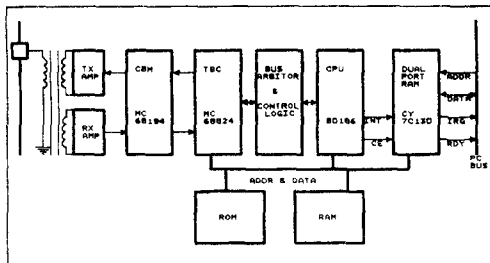


그림 3. 스테이션 접속기의 하드웨어 블록 구조

1) 호스트 접속 회로

네트워크접속기와 호스트 시스템의 접속 방식은 두 시스템 간의 정보 전송 속도에 큰 영향을 주게 되는데, 이는 결국 LAN의 성능에까지 영향을 주게한다. 일반적으로 DMA 접속 방식, 인터럽트에 의한 핸드 셰이크 방식등이 검토되는데 DMA 방식은 정보 전송율은 좋으나 두 시스템 버스 분리와 버스 제어권 관리가 복잡 해지고, 인터럽트방식은 정보 전송 속도가 낮아서 정보 전송량이 많아지면 전체 성능이 저하된다.

본 연구에서는 1K바이트의 듀얼 포트 메모리를 사용하여 두 시스템버스의 분리와 DMA에 의한 정보 전송의 고속성 및 인터럽트에 의한 시스템간의 정보 교환이 가능했다.

2) CPU 와 TBC 간의 접속

CPU와 TBC 간의 정보 교환은 TBC의 내부 레지스터를 통하는 형태와, 공유 메모리를 통하는 형태로 구별된다. CPU와 TBC 는 같은 버스를 사용 하므로, 버스 제어권 중계 회로를 두어 정보 충돌을 배제 하였다. 데이터의 교환은 공유 메모리가 사용되고 동작제어 명령은 레지스터를 통해 전달 된다. CPU가 TBC의 동작을 명령하면 TBC 는 독립적으로 데이터 송수신을 수행하게 되는데 이때 내부 FIFO 상태에 따라 버스 제어권을 CPU에 요구하고 공유 메모리에서 데이터를 취하거나 혹은 수신한 데이터를 CPU가 설정한 환경에 맞게 공유 메모리상에 적재한다. 그림 4.은 TBC와 CPU 간의 버스 중계회로와 이의 시간동작을 보여주고 있다. TBC에서 출력한 버스요구신호는 CPU의 HOLD신호로 입력되고 CPU는 명령의 수행을 종료하면서 HOLDA 신호를 출력하여 TBC 에 버스사용권을 넘긴다. TBC의 버스사용이 끝나면 TBC는 BGACK신호를 이용하여 버스사용권을 다시 CPU로 이전한다.

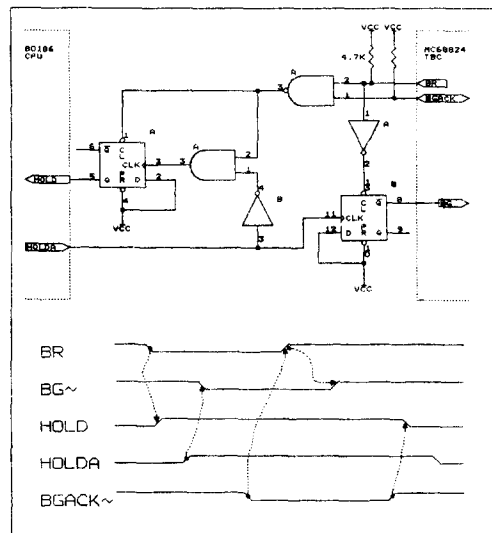


그림 4. 버스 중계회로의 동작

4. 접속기의 소프트웨어 구조

접속기의 소프트웨어 구조는 크게 TBC를 운영하는 운영 프로그램과 호스트와의 인터페이스를 담당하는 부분으로 나뉜다. 스테이션 인터페이스는 dual port RAM 최 상위의 바이트(03ffh)를 액세스하여 필요할때 상대측으로 인터럽트를 걸어 host와의 정보 교환을 담당한다. TBC는 마이크로 프로세서의 관리로 토큰 버스 매체의 액세스를 제공하며 노드의 추가, 삭제 및 에러 보정을 위한 정보들을 제공해 준다. TBC와 마이크로 프로세서와의 통신을 기본적으로 공유 메모리를 통하여 커맨드 구조를 사용하여 TBC의 동작을 제어한다. 그림 5에 접속기의 소프트웨어 전체 구조를 보여준다.

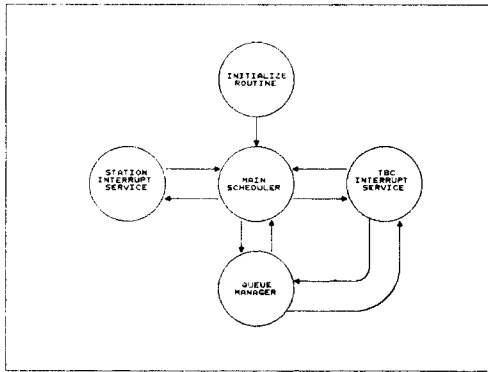


그림 5. 접속기의 소프트웨어 흐름

프레임 전송을 위해서 TBC는 frame descriptor, buffer descriptor, data buffer로 구성된 buffer structure를 구성하고있으며 이들의 초기화와 link, 유지 절차를 프로세서에서 제어해 주어야한다. 각 송,수신당 하나의 FD와 적어도 하나 이상의 BD,DB가 필요하게된다.

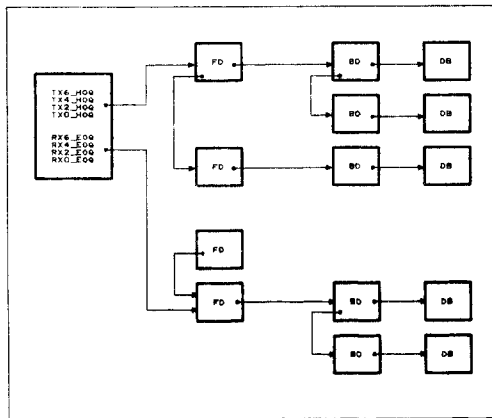


그림 6. TBC의 buffer structure

Queue manager 루틴에서는 8개의 TX,RX FD와 12개의 TX,RX BD를 link시켜 pool을만들고, 각각의 head, tail 포인터를 유지하여 사용되어진FD, BD를 pool에 다시 link시키

는 기능을 수행한다. 본연구에서는 priority 기능을 구현하기 위해 4개의 TX_dummy_FD와 RX_dummy_FD를 두어 pool의 FD를 사용하는 access class가 바뀔 경우에 포인터가 잘못 지정되는 경우를 막고있다.

그림 7~10까지 하나의 class6 프레임을 송,수신하는 경우의 Queue manager 루틴에 의한 buffer structure 구조의 link 과정을 보였다.

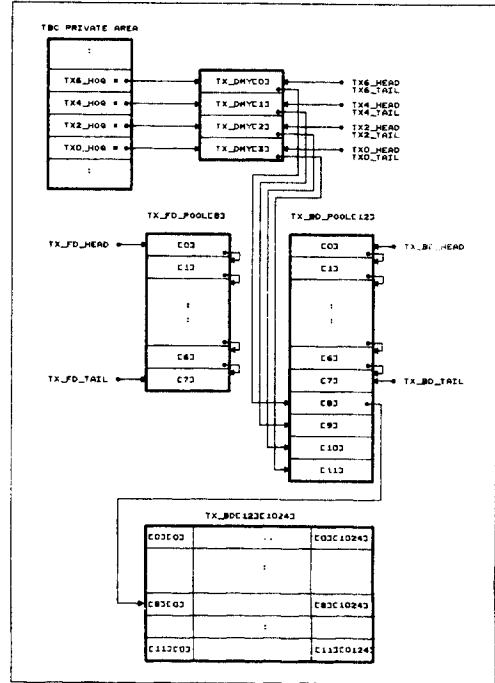


그림7. TX_queue의 초기화

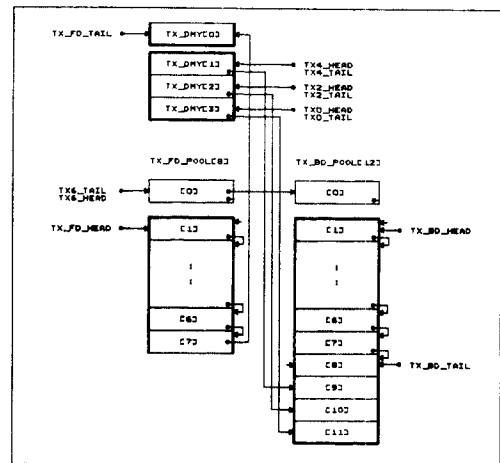


그림8. 하나의 class6 프레임을 보내고 confirm을 받은 경우의 buffer structure

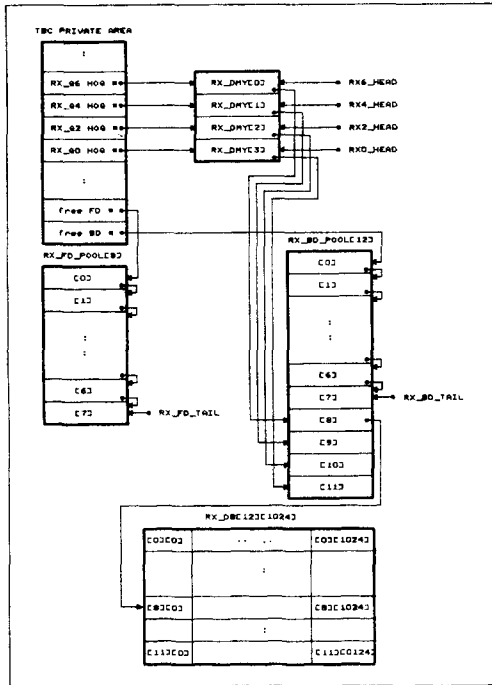


그림9. RXqueue의 초기화

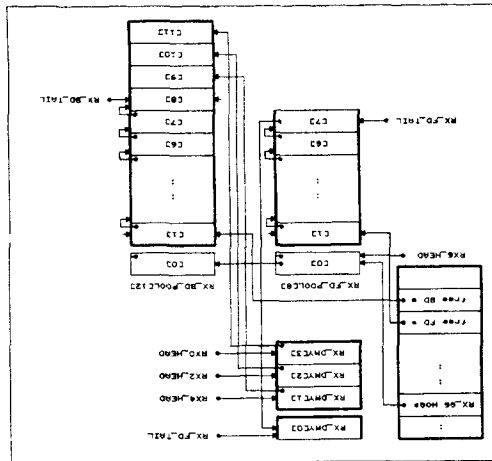


그림10. 하나의 class6 프레임 수신한경우의 buffer structure

5. 측정 및 실험

실험은 세대의 스테이션 접속기를 제작하였으며 host인 IBM-PC에 Application으로 간단한 파일 전송과 네트워크 상태를 모니터 할 수 있는 운용 프로그램을 올려 네트워크 초기화 과정과 프레임 송,수신을 실험하고,한대의 스테이션을 모니터 상태로 두어 토큰 버스 프로토콜의 동작에 대한 파라미터들의 변화를 측정하였다.

일반적으로 낮은 traffic에서는 각 access class의 성능에 많은 변화를 주지 않았지만 traffic이 커질 수록 낮은 access class의 target rotation time을 증가시키는것이 성능을 높일 수있었고, token hold time은 될수있는대로 큰편이 바람직하다.

6. 결론

본 연구에서는 IEEE LAN 표준 프로토콜의 하나인 토큰 패싱 버스 프로토콜을 구현한 스테이션 접속기를 개발하였다. MAC 계층 구현에 사용된 모토롤라사의 토큰 버스 컨트롤러는 복잡한 프로토콜을 소프트웨어로 구현할 시의 overhead를 줄일 수 있어 접속기의 공유 메모리 탕을 줄일 수 있었다. 특히 송,수신에대한 4 레벨의 우선순위부여기능과 RWR 기능을 제공하고 있어 이와함께 LLC type 3를 구현한다면 이후 MAP 네트워크의 하위계층에도 적합하게 운용될 수 있을것이다.

7. 참고 문헌

- [1] SME, *HAP Application Overview*, SME, 1986
- [2] GM MAP Task Force, *Manufacturing Automation Protocol V3.0*, GM, 1987.5
- [3] IEEE, *IEEE Draft Standard 802.4*, 1983
- [4] IEEE, *Logical link Control, ANSI/IEEE std. 802.2 for Lan*, 1985
- [5] Motorola, *MC68824 Token Bus Controller User's manual*, 1987
- [6] Motorola, *MC68194 Carrierband MODEM Advanced Information*, 1989
- [7] William Stallings, *Local Networks*, Macmillan Publishing CO., 1987
- [8] 김동규, "컴퓨터 통신 네트워크", 상조사, 1987
- [9] 김 용득, "MAP/EPA 용 네트워크 접속기 개발에 관한 연구", 전자 공학회 논문집, 12권 1호, pp.204~207, 1989.7
- [10] 임 춘규, 송 재규, 유선TV 방송 기술, 가남사, 1989