

---

# SPI-4.2 프로토콜을 사용한 PHY-LINK 계층간의 데이터 전송 성능평가

박노식\* · 손승일\* · 최익성\*\* · 이범철\*\*

## The Performance Evaluation for PHY-LINK Data Transfer using SPI-4.2

Noh-Sik Park, Seung-Il Sonh, Ick-Sung Choi, Bhum-Cheol Lee

### 요 약

시스템 패킷 인터페이스 4레벨 2단계(System Packet Interface Level4 Phase 2)는 10Gbps 이더넷 응용 뿐만아니라, OC-192 대역폭의 ATM 및 POS를 통한 패킷 또는 셀 전송을 위한 물리계층과 링크계층 소자간의 인터페이스이다.

본 논문에서는 시스템 패킷 인터페이스 4레벨 2단계(SPI-4.2)에 대한 연구와 C언어를 이용한 인터페이스 모듈의 성능평가를 실시하였다. SPI-4.2 인터페이스 모듈은 512워드의 FIFO를 사용할 경우 랜덤 유니폼 트래픽에서는 97%까지, 버스트 길이 32를 갖는 버스트 트래픽에서는 94% 까지의 offered load에 대해 적응이 가능하다. 그리고 14 바이트 미만의 작은 크기 패킷을 대규모로 수신할 경우, 오버헤드로 인한 성능 저하가 발생한다는 것을 확인하였다. SPI-4.2 인터페이스 모듈은 기가비트/테라비트 라우터, 광학 크로스바 스위치 및 SONET/SDH 기반의 전송 시스템에서 라인카드로 사용할 경우 적합할 것으로 사료된다.

### ABSTRACT

System Packet Interface Level 4 Phase(SPI-4.2) is an interface for packet and cell transfer between a physical layer(PHY) device and a link layer device, for aggregate bandwidths of OC-192 ATM and Packet Over Sonet/SDH(POS), as well as 10Gbps Ethernet applications. In this paper, we performs the research for SPI-4.2. Also we analyze the performance of SPI-4.2 interface module after modeling using C programming language. This paper shows that SPI-4.2 interface module with 512-word FIFO depth is able to be adapted for the offered loads to 97% in random uniform traffic and 94% in bursty traffic with bursty length 32. SPI-4.2 interface module can experience a performance degradation due to heavy overhead when it massively receives small size packets less than 14-byte. SPI-4.2 interface module is suited for line cards in gigabit/terabit routers, and optical cross-connect switches, and SONET/SDH-based transmission systems.

### 키워드

SPI-4.2, OC-192 ATM, 10Gbps Ethernet, Interface Protocol

### 1. 서 론

인터넷의 급속한 발달과 폭발적인 확산에 따라, 인터넷 관련 기술은 빠르게 발전되고 있다. 이러한

인터넷의 확산으로 더욱더 빠른 네트워크망이 요구되고 있다. 네트워크망의 발전과정을 보면, 기존의 전화선망, 전용선망을 거쳐 이제는 빛의 속도로 데이터를 전송할 수 있는 광통신망이 요구되고, 실제 많은 네트워크망이 광통신망으로 바뀌고 있다. 네트워크망의 발전 속도를 현재 구축된 인터페이스 디바이스들은 이를 충족시키지 못하고, 네트워크망 내부에서 병목현상을 일으키고 있다. 이를 해결하기 위해 고속의 대역폭을 지원하는 인터페이스 기술이 발표되고 있다[1]-[5]. 시스템 패킷 인터페이스 4레벨 2단계는 물리층 소자와 링크층 소자간의 10Gb/s 이상의 대역폭을 위한 인터페이스로 광 인터페이스 포럼(Optical Interface Forum)에서 권장하는 규격이다[3].

SPI-4.2 프로토콜 코어나 이를 내장한 네트워크 프로세서 제품을 살펴보면, Paxonet 커뮤니케이션사의 CC401 코어, Xilinx사의 SPI-4.2 코어, 인텔사의 IXP2800 네트워크 프로세서, Altera사의 Stratix GX, Lattice 반도체사의 ORSPI4 등이 있으며, 향후에도 SPI-4.2 프로토콜을 지원하는 제품이 활발히 출시될 것으로 예견되고 있다[6]-[11].

본 연구에서는 OC-192 ATM과 POS (Packet Over SONET/SDH), 그리고 10Gb/s 이더넷을 통한 패킷 또는 셀 전송을 위한 물리계층소자와 링크계층소자 간의 인터페이스인 시스템 패킷 인터페이스(SPI-4.2)에 대한 연구와 SPI-4.2 인터페이스 모듈을 C언어를 이용하여 성능평가와 최적화를 실시하였고, 향후 본 연구에서 도출된 SPI-4.2 모듈을 추후 VHDL 언어를 이용하여 설계하여 구현하고자 한다.

## II. SPI-4.2의 개요

SPI-4.2는 물리층 소자와 링크층 소자간의 데이터 전달을 위한 시스템 패킷 인터페이스이다. 그림 1은 SPI-4.2의 개념도를 보여주고 있다. 한편, SFI (SERDES Framer Interface)는 PHY 계층인 SONET/SDH 프레임러와 고속 Parallel-to-Serial/Serial-to-Parallel(SERDES) 로직과의 전기적인 인터페이스를 정의하고 있다[5].

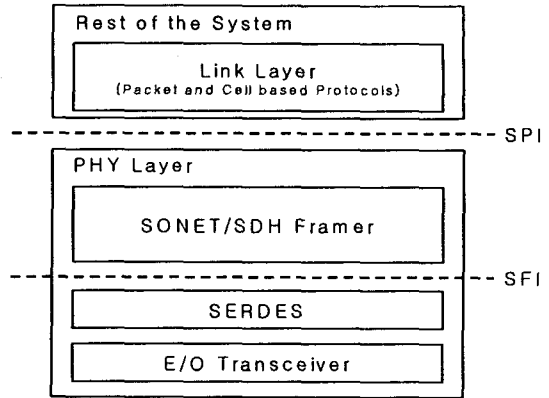


그림 1. SPI-4.2의 개념도  
Fig. 1 Overview of SPI-4.2

SPI-4.2는 데이터와 제어 및 상태 신호들의 전달 방향에 따라 "Transmit"과 "Receive"로 구분된다. 그림 2는 SPI-4.2의 인터페이스 신호를 보여주고 있다.

양방향 모두에 대해 FIFO 상태 신호는 데이터와 별도로 전송되며, 이러한 구조 때문에 송수신부를 서로 격리시킬 수 있고 양방향성이 아닌 단방향의 특성을 가져야 하는 응용분야에도 SPI-4.2를 사용할 수 있다.

16비트의 데이터 버스를 가지고 있으며 각 데이터 동기를 위하여 TDCLK신호를 각 모듈의 수신단에 전송하게 된다. 데이터를 전송하기 위해서 SPI-4.2모듈은 2비트의 수신단 FIFO의 상태정보를 수신하고 수신된 상태정보를 확인하여 데이터를 전송하게 된다. 상태신호의 동기를 위하여 TSCLK 신호를 입력받게 된다[3].

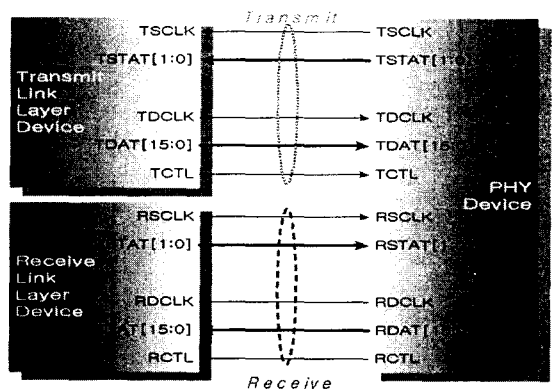


그림 2. SPI-4.2의 인터페이스 신호  
Fig. 2 Interface signals for SPI-4.2

데이터의 전송은 제어가 가능한 최대 길이로 연속(Burst)해서 전달될 수 있으며 EOP(End of Packet) 신호에 의해 전달이 중지된다. 매 전달과 관련된 정보들은 16비트 제어워드 형태로 전달된다. 그림 3은 데이터 흐름에 ATM 셀과 가변길이의 패킷이 매핑되는 것을 보여주고 있다[3].

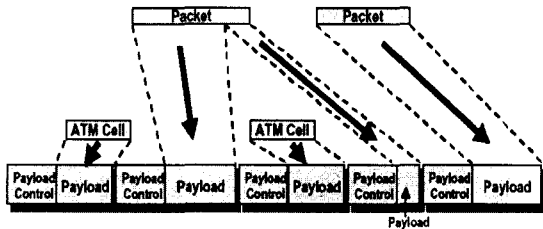


그림 3. 페이로드 스트림 상으로 패킷과 ATM 셀의 매핑

Fig. 3 Mapping of packets and ATM cells onto payload stream

그림 4는 SPI-4.2의 소스로 입력되는 데이터 포맷을 보여주고 있다. 64비트의 데이터와 14비트의 부가 정보를 가지고 있다. Sop는 패킷의 시작을, Eop는 패킷의 끝을, P는 현재 데이터의 에러 정보를, VByte는 데이터의 유효 데이터의 바이트 수를 나타내고, Port Address는 목적지 포트주소를 가지고 있다[3].

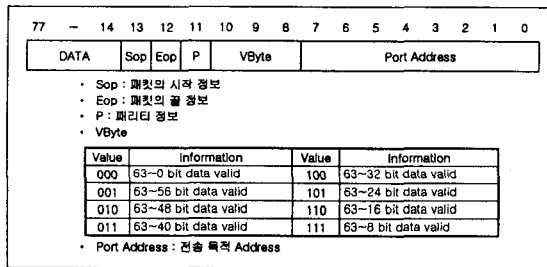


그림 4. 입력데이터 형식  
Fig. 4 Input data format

그림 5는 소스에서 전송되고, 싱크에서 수신하는 컨트롤 워드 형식을 보여주고 있다. 컨트롤 워드는 16비트이다. Tp는 컨트롤 워드가 Idle 컨트롤 인지 페이로드 컨트롤인지를, Eop는 마지막 패킷을, Sop는 패킷의 시작을, Port Address는 목적지 주소를, DIP(Diagonal Interleaved Parity)-4는 에러 정보를 나타낸다[3].

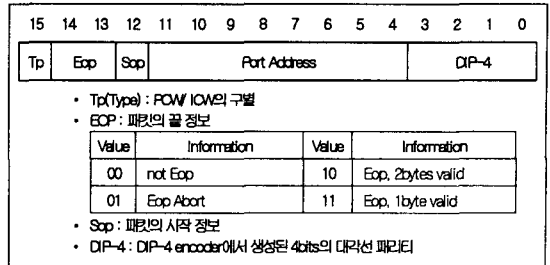


그림 5. 컨트롤 워드 형식  
Fig. 5 Control word format

SPI-4.2 인터페이스는 광통신에 적용하기 위한 인터페이스로 정확한 데이터 전송을 위해 데이터 라인에서 발생할 수 있는 비트 도달시간을 조정하여 올바른 신호를 수신할 수 있는 기능을 제공하여 데이터 전송 타임을 조정할 수 있는 기회를 제공한다. 이와 같은 처리를 트레이닝 시퀀스라 하며, 트레이닝 시퀀스는 별도로 설정된 주기와 연속적인 에러가 발생하였을 때 수행하게 된다.

### III. 시뮬레이션 모델

SPI-4.2 모듈에 포함되어야 할 파라미터 정보의 효율적인 적용과 내부 FIFO의 크기값에 대한 도출과 효율적인 하드웨어 설계를 위하여 C언어를 이용하여 설계모듈을 성능평가 하였다. 그림 6은 성능평가를 위한 시스템 블록도를 보여주고 있다.

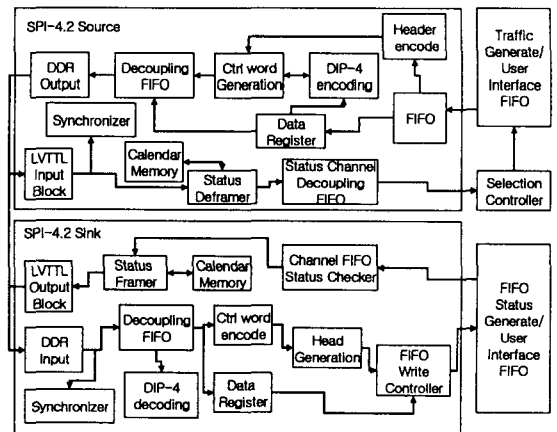


그림 6. 성능평가 시스템 블록도  
Fig. 6 Block diagram of performance evaluation system

성능 평가를 위한 시스템 블록도는 다양한 패킷 크기를 갖는 데이터의 생성과 Offered Load에 의해 트래픽이 생성되는 Traffic Generator 모듈, Traffic Generator에서 생성된 데이터를 PHY 계층에서 요구하는 데이터 형태로 바꿔주고, 제어정보를 삽입하는 Source 모듈과 수신된 데이터를 Data Link 계층에서 요구하는 데이터 형태로 바꿔주고, 헤더정보를 만들어주는 Sink 모듈로 구성되어 있다.

3.1 송/수신부의 동기

데이터전송의 시작과 특정주기에 도달하였을 때 SPI모듈은 송신단과 수신단의 동기와 비트도달시간을 조정하기위하여 트레이닝 시퀀스를 시행한다. 송신부에서는 데이터 패스를 통해 트레이닝 패턴을 n회 전송하면 수신부에서는 이 트레이닝 패턴을 n회 수신 후 수신단의 동기와 비트도달시간을 조정하게 된다. 수신단의 동기가 이루어진 후 수신단은 전송단에 FIFO 상태 전송 데이터 패스를 통하여 프레임 패턴을 a번 전송부에 전송하면 전송부에서는 이 프레임 패턴을 a번 수신하여 수신단의 동기를 확인하고 전송부와 수신부의 동기를 맞춘다. 전송부와 수신부의 동기 신호를 확인한 후 데이터전송을 시작한다. 그림 7은 전송부와 수신부의 동기를 맞추는 작업을 순서도로 보여주고 있다.

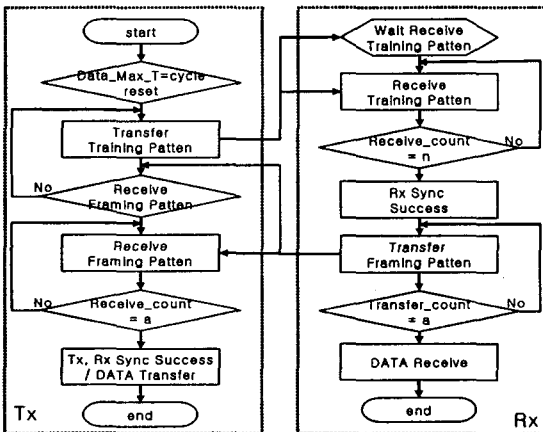


그림 7. 전송부와 수신부의 동기 작업의 순서도  
Fig. 7 Flowchart for synchronization of sending and receiving sides

3.2 전송부의 설계

전송부는 트래픽 생성기에 의해 생성된 78비트

의 헤더를 포함한 데이터를 소스로 입력하여 헤더 정보를 분석 후 물리계층으로 전송할 컨트롤 워드를 생성하고, 생성된 컨트롤 워드와 데이터정보를 이용하여 DIP-4정보를 생성하여 내부 디커플링 FIFO에 32비트의 데이터의 크기로 저장한 후 디커플링 FIFO에서 DDR 인터페이스를 통하여 데이터를 전송한다. 그림 8은 전송부의 패킷 처리 순서도를 보여주고 있다.

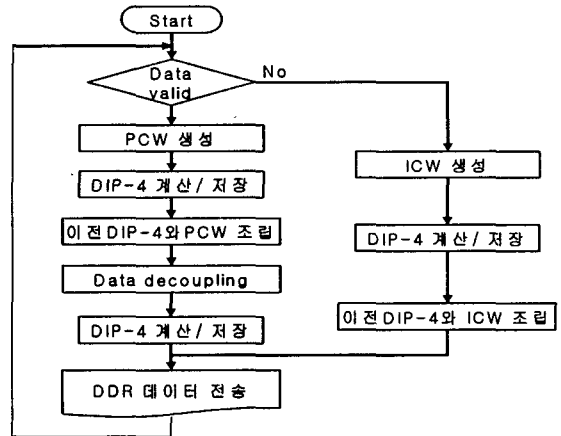


그림 8. 전송부의 패킷 처리 순서도  
Fig. 8 Flowchart of packet processing(sending side)

3.3 수신부의 설계

수신부에서는 DDR(Double Data Rate) 인터페이스를 통해 들어온 데이터를 커플링 FIFO에 16비트 데이터를 32비트로 조합하여 저장한다. 전송부의 역순으로 컨트롤 워드를 분석하여, 데이터를 재조립하여 링크 계층으로 전송한다. FIFO상태 생성기는 링크계층의 FIFO 상태를 생성해주는 모듈로서 FIFO 상태의 생성은 가득 찬 상태와 비어있는 상태를 이용하여 생성하게 하였다. 데이터의 전송은 FIFO 상태정보가 싱크모듈에서 수신후 데이터를 전송하게 된다. FIFO 상태정보 전송에 있어서 소스모듈과 싱크모듈이 공통으로 공유하는 Port 처리 순서 정보를 가지고 있는 카렌더 메모리를 접근하여 해당 FIFO 상태 정보를 수신하여 데이터 전송을 제어한다[7]. 그림 9는 수신부의 패킷 처리 순서도를 보여주고 있다.

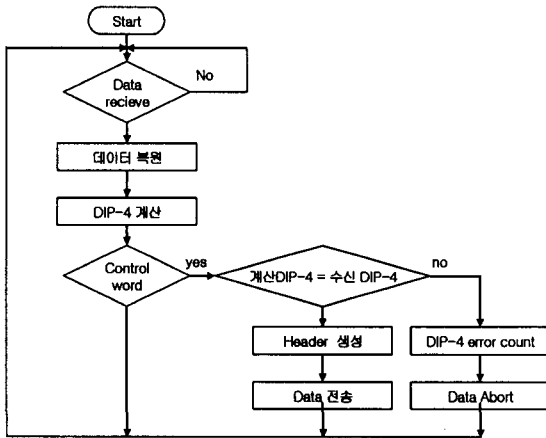


그림 9. 수신부의 패킷 처리 순서도  
Fig. 9 Flowchart of packet processing(receiving side)

### 3.4 FIFO 상태신호 전송부 설계

FIFO 상태 정보는 수신부에서 주기적으로 전송부에 전달되는 신호로서 SPI-4.2 인터페이스 모듈에 의해 수신될 모듈의 FIFO 상태정보를 생성하여 전송부에 보내주게 된다. 그림 10은 수신부의 FIFO 상태정보 전송을 위한 순서도를 보여주고 있다.

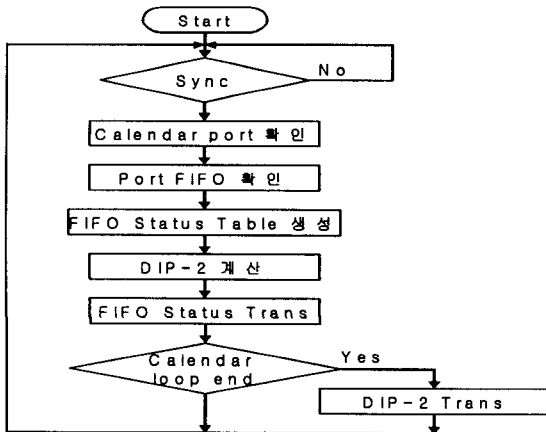


그림 10. 수신부의 FIFO 상태 정보 전송 순서도  
Fig. 10 Flowchart of FIFO status transfer(receiving side)

### 3.5 FIFO 상태신호 수신부 설계

수신부에서 주기적으로 보내지는 FIFO 상태 정보를 확인하여 해당 포트의 데이터를 전송하게 된다. 그림 11은 수신된 FIFO 상태 정보를 업데이트 하고 채널 정보를 알려주기 위한 순서도를 보여주

고 있다.

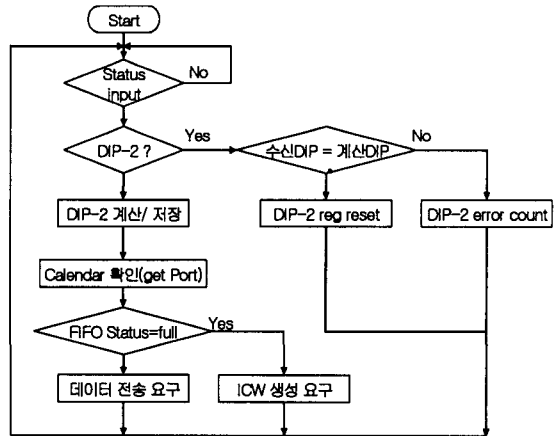


그림 11. 전송부의 FIFO 상태 정보 확인 후 데이터 전송을 위한 순서도

Fig. 11 Flowchart of data transfer after FIFO status verification(sending side)

## IV. 성능 평가

C언어를 이용해 설계된 SPI-4.2 모듈의 내부 FIFO 적용도를 판별하기 위하여 입력되는 트래픽에 Offered Load를 주어 각 Offered Load에 의한 FIFO 크기 변화를 확인하였다. 입력되는 트래픽은 8개의 다른 포트 주소를 할당하여 시뮬레이션 하였는데, 6개의 입력 포트는 1Gb/s의 대역폭을, 그리고 2개의 입력 포트는 2Gb/s의 대역폭을 할당하여 총 10Gb/s의 대역폭을 유지하여 시뮬레이션 하였다. 그리고 패킷크기의 변화에 따른 FIFO의 변화량을 확인하였다.

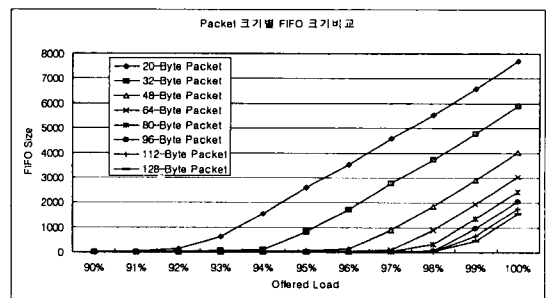


그림 12. 패킷크기별 FIFO 크기변화  
Fig. 12 FIFO sizes according to various packet sizes

확인하고자 하는 FIFO는 입력단에 패킷의 처리를 기다리는 FIFO로서 내부인터페이스 모듈의 컨트롤 워드의 생성, 트레이닝 시퀀스에 의해 대기 패킷이 생겨 FIFO의 크기를 변화시키게 되는데 이를 확인하였다. 그림 12는 패킷 크기별 FIFO 크기변화를 보여주고 있다. 그림 12에서 보이는 것과 같이 패킷 크기가 작아질수록 FIFO사용량이 많음을 확인할 수 있다. 패킷크기가 아주 작을 때는 많은 양의 컨트롤 정보가 생성되어 입력단의 FIFO 사용량이 급증하는 것을 확인하였다.

그림 13은 패킷 크기가 10바이트일 때의 FIFO 사용량을 보여주고 있다.

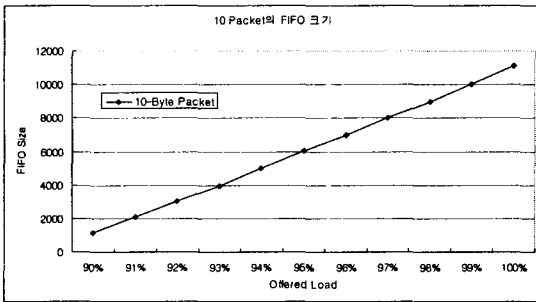


그림 13. 패킷크기 10일 때의 FIFO 크기변화  
Fig. 13 FIFO sizes when packet size is 10-byte

그림 14는 4개의 패킷 크기가 서로 다른 셀을 입력하여 얻어진 결과이다. 다음 결과에서 볼 수 있듯이 패킷의 크기가 작은 셀이 들어있는 경우의 FIFO 크기를 많이 차지함을 확인할 수 있다.

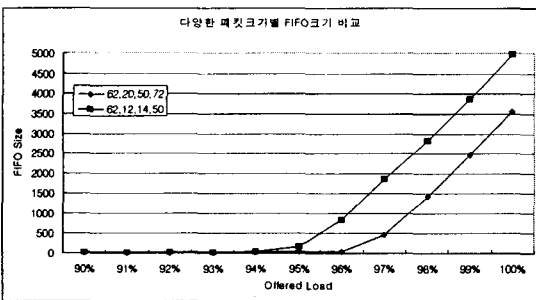


그림 14. 다양한 패킷 크기별 FIFO 크기 비교  
Fig. 14 FIFO sizes according to various packet sizes

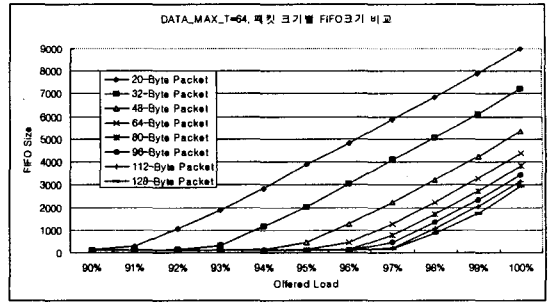


그림 15. DATA\_MAX\_T=64일때의 패킷 크기별 FIFO 크기  
Fig. 15 FIFO sizes according to packet sizes(DATA\_MAX\_T=64)

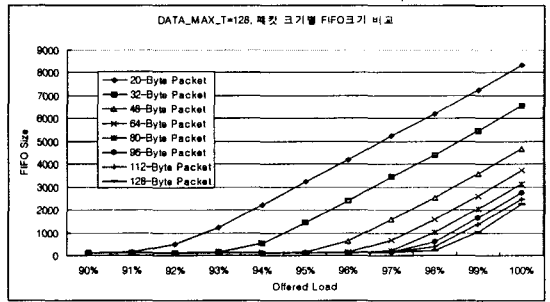


그림 16. DATA\_MAX\_T=128일때의 패킷 크기별 FIFO 크기  
Fig. 16 FIFO sizes according to packet sizes(DATA\_MAX\_T=128)

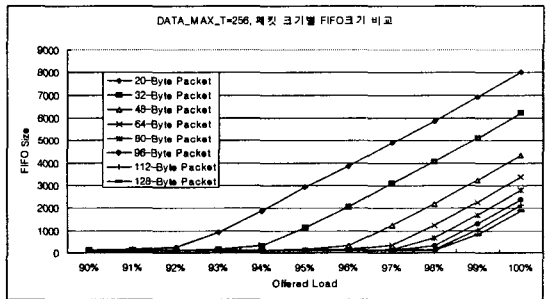


그림 17. DATA\_MAX\_T=256일때의 패킷 크기별 FIFO 크기  
Fig. 17 FIFO sizes according to packet sizes(DATA\_MAX\_T=256)

트레이닝 시퀀스는 미리 정의된 DATA\_MAX\_T값

에 의해 DATA\_MAX\_T×256 Cycle이 되었을 때 주기적으로 수행하게 되어있다. 각각 DATA\_MAX\_T값의 변화에 따른 FIFO 크기의 변화를 측정하였다. 그림 15, 16, 17은 DATA\_MAX\_T 값을 64, 128, 256으로 하였을 때의 FIFO 크기 변화를 보여주고 있다.

그림 18은 패킷 크기가 64바이트일 때 DATA\_MAX\_T별 FIFO 크기를 보여주고 있다.

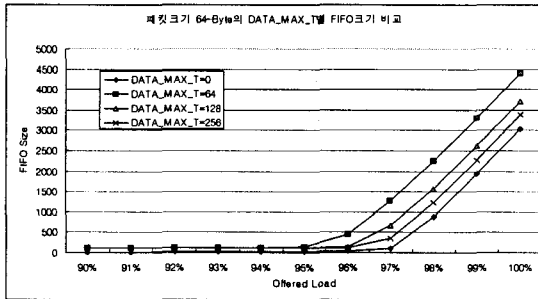


그림 18. 64바이트 패킷일 때의 DATA\_MAX\_T별 FIFO 크기 비교

Fig. 18 FIFO sizes according to various DATA\_MAX\_T values (packet size = 64-byte)

그림 19는 4개의 서로 다른 패킷 크기를 입력하였을 때 DATA\_MAX\_T별 FIFO 크기를 보여주고 있다.

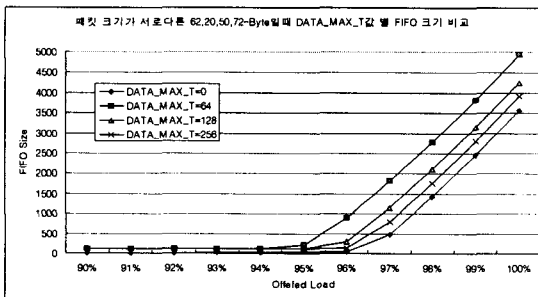


그림 19. 62,20,50,72바이트 4개의 패킷 종류가 입력될 때 DATA\_MAX\_T별 FIFO 크기

Fig. 19 FIFO sizes according to DATA\_MAX\_T values when the types of packet are 62, 20, 50 and 72-byte)

SPI-4.2는 연속적으로 들어오는 데이터 버스트의 일정한 크기가 지났을 경우 자동적으로 PCW를 생성하여 일정한 간격으로 데이터 버스트의 사이에

입력한다. SPI-4.2 성능평가 프로그램에 데이터 버스트 크기가 각각 32바이트, 64바이트일때 PCW를 삽입하게 하여 성능평가를 하였다. 이때에 DATA\_MAX\_T값은 256으로 설정하였다. 그림 20은 버스트 길이가 32바이트일 때 32바이트를 초과하는 패킷크기를 가질때 다음과 같은 결과를 확인하였다.

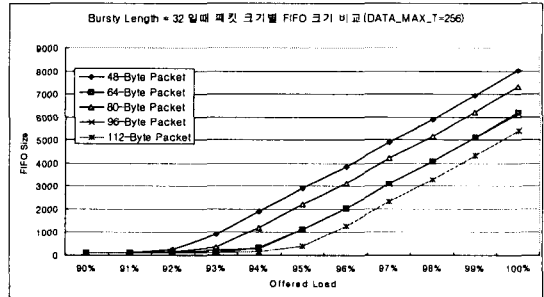


그림 20. Burst Length = 32바이트일때 패킷 크기별 FIFO 크기

Fig. 20 FIFO sizes according to various packet sizes(Bursty Length = 32)

그림 21은 버스트 길이가 64바이트일 때 64바이트를 초과하는 패킷크기를 가질때 다음과 같은 결과를 확인하였다.

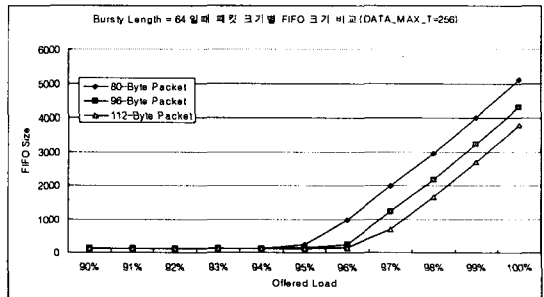


그림 21. Burst Length = 64바이트일때 패킷 크기별 FIFO 크기

Fig. 21 FIFO sizes according to various packet sizes(Bursty Length = 64)

그림 22는 버스트 길이가 32, 64바이트일 때 서로 다른 패킷크기를 가지는 데이터 패킷이 입력되었을 때 FIFO 크기를 확인하였다.

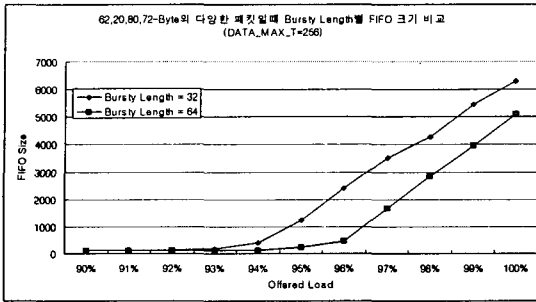


그림 22. 62,20,80,72바이트의 다양한 패킷일 때 버스트 길이별 FIFO 크기

Fig. 22 FIFO sizes according to bursty length values when the types of packet are 62, 20, 80 and 72-byte)

### V. 결 론

SPI-4.2는 10Gb/s 이상의 대역폭을 위한 패킷 인터페이스를 지원한다. 정의된 인터페이스는 라인당 최소 622MHZ 이상의 전송률을 가지는 16비트의 데이터 패스로 구성되며, OC-192 SONET/SDH를 통한 셀 또는 패킷의 전송, 그리고 10Gb/s 이더넷을 지원한다.

본 논문에서는 시스템 패킷 인터페이스 4레벨 2 단계(SPI-4.2)에 대한 연구와 모듈을 C언어로 구현하여 패킷 크기의 변화에 따른 SPI-4.2의 성능 측정을 하였다. SPI-4.2 인터페이스 모듈은 512워드의 FIFO를 사용할 경우 랜덤 유니폼 트래픽에서는 97%까지, 버스트 길이 32를 갖는 버스트 트래픽에서는 94%까지의 offered load에 대해 적응이 가능하다는 것을 확인하였다. 하지만 14바이트 미만의 작은 패킷을 대량으로 처리할 때 SPI-4.2 모듈은 데이터 패스를 충분히 활용할 수 없기 때문에 이로 인해 FIFO에 쌓이는 패킷이 많이 발생하게 된다는 것을 확인하였다. 추가적으로 SPI-4.2에 작은 크기의 패킷이 들어왔을 때 적응이 가능한 SPI-4.2에 대해 연구가 필요함을 확인하였다.

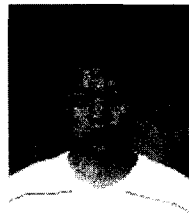
### 참고 문헌

[1] "The Voice of the Future : Next Generation Networks", ATM Form, Jul. 2002  
 [2] "Common Switch Interface Specification-L1", ATM Form, Aug. 2001

[3] "System Packet Interface Level 4 (SPI-4) Phase 2 : OC-192 System Interface for Physical and Link Layer Devices.", OIF, Jan. 2001  
 [4] "Streaming Interface(NPSI) Implementation Agreement," NPF, Oct. 2002  
 [5] "OIF Electrical Interfaces", OIF, 2001  
 [6] "CoreEI SPI-4 Phase 2 Interface Core(CC 401)", Paxonet Communications, Inc., May 2003  
 [7] "SPI-4.2(PL4) Core V6.1" Xilinx Inc., Feb. 2004  
 [8] "POS-PHY Level 4 MegaCore Optimization for the Intel IXP2800 Network Processor", White Paper, April 2003  
 [9] "SPI-4.2 Interoperability with the Intel IXF 1110 in Stratix GX devices", Altera Corp., May 2003  
 [10] "SPI-4.2 Interoperability with PMC-Sierra X ENON Family in Stratix GX devices", Altera Corp., May 2003  
 [11] "ORCA ORSPI4 - Dual SPI4 Interface and High-Speed SERDES FPSC", Lattice Semiconductor Corp., Jan. 2004

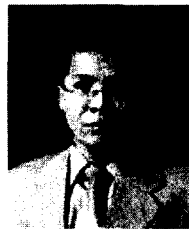
### 저자소개

#### 박노식(Noh-Sik Park)



2003년 호남대학교 컴퓨터공학과(학사)  
 2003년 한신대학교 정보과학대학원(석사과정)  
 ※관심분야 : ATM 통신, 네트워크

#### 손승일(Seung-Il Sonh)



1989년 연세대학교 전자공학과(학사)  
 1991년 연세대학교 대학원 전자공학과(석사)  
 1998년 연세대학교 대학원 전자공학과(박사)

1998~2002년 호남대학교 컴퓨터공학과 조교수  
 2002년~현재 한신대학교 정보통신학과 부교수  
 ※관심분야 : ATM 통신 및 보안, ASIC 설계





최익성(Ick-Sung Choi)

1992년 서강대학교 전자공학과  
(학사)  
1994년 서강대학교 대학원 전자  
공학과(석사)  
1999년 서강대학교 대학원 전자  
공학과(박사)

1999년~현재 한국전자통신연구원 스위칭기술팀  
선임연구원 재직 중.

※관심분야 : 설계자동화, 저전력 회로 설계 및  
합성, computer 구조 및 설계, VLSI testing,  
switching system design.



이범철(Bhum-Cheol Lee)

1997년 연세대학원 전자공학과  
졸업(박사)  
1983년 - 1995년 한국전자통신연  
구원 입사. Interfaces, Switching,  
Link, and Network Synchroniz-  
ation Engineer.

1995년~현재 한국전자통신연구원 스위칭기술팀장  
재직 중.

※관심분야 : Bit Synchronization, High-speed  
I/O, Multi-layer Ethernet Switch and Switch  
Fabric based on QoS.