

# CSIX 모듈의 FPGA 구현 및 검증

## FPGA Implementation and Verification of CSIX Module

김형준\*  
Hyoung-Jun Kim

손승일\*\*  
Seung-Il Sonh

강민구\*\*\*  
Min-Goo Kang

### 요약

본 논문에서 설계한 CSIX 모듈은 ATM, IP, MPLS, Ethernet과 같은 데이터 통신의 응용을 위해 트래픽 매니저와 스위칭 패브릭 사이의 표준화된 인터페이스 규격인 CSIX-L1(Common Switch Interface - Level1)을 따른다. 이 모듈은 전달하고자 하는 데이터를 CSIX 정보 유닛인 CFrame으로 생성하고 수신측에서는 수신한 CFrame에서 데이터를 추출하는데 사용된다. CSIX 모듈은 32, 64, 96, 128-bit interface를 지원하며, 가변길이의 CFrame 및 Idle CFrame을 생성하고, Padding byte를 생성하며, Vertical parity를 지원한다. Xilinx ISE 4.1i를 사용하여 전체적인 작업이 이루어졌으며, 설계된 모듈에 대한 테스트를 수행한 후, Xilinx FPGA XCV1000EH6240C 칩에 다운로드하여 기능을 검증하였다. 합성된 CSIX모듈은 27MHz에서 동작한다.

### Abstract

CSIX-L1 is the Common Switch Interface that defines a physical interface for transferring information between a traffic manager (Network Processor) and a switching fabric in ATM, IP, MPLS, Ethernet and data communication areas. In Tx, data to be transmitted is generated in CFrame which is the base information unit and in Rx, original data is extracted from the received CFrame. CSIX-L1 supports the 32, 64, 96, and 128-bit interface and generates a variable length CFrame and Idle CFrame. Also CSIX-L1 appends Padding byte and supports 16-bit Vertical parity. CSIX-L1 is designed using Xilinx 4.1i. After functional and timing simulations are completed, CSIX-L1 module is downloaded in Xilinx FPGA XCV1000EH6240C and verified. The synthesized CSIX module operates at 27MHz.

## 1. 서론

오늘날 ATM 스위치 패브릭 생산 업체들이 발표한 스위치 패브릭은 트래픽 매니저와 스위치 패브릭 사이에 공통된 인터페이스 규격을 적용하여 설계하는 것이 아니라 자신들이 정의한 인터페이스 방식에 따라 데이터 및 제어정보를 전송하고 있다. 따라서 타사의 스위치 패브릭을 사용할 경우에는 호환성에 있어서 문제가 발생한다.

따라서 트래픽 매니저와 스위치 패브릭 사이에 공통된 인터페이스 규격을 사용함으로써 특정 회사의 제품에 종속되지 않고 사용자가 원하는 스위치 패브릭을 사용하려는 경향이 대두되고 있다. 이러한 경향의 일환으로 최근 스위치 인터페이스와 관련하여 표준화가 진행되고 있다. CSIX(Common Switch Interface)라 불리는 이 표준화는 1차 표준이 2000년 8월에 발표되었고, 2001년에는 CSIX-L1을 지원하는 2.5Gbps급 네트워크 프로세서(Traffic Manager)가 출시되었다. 현재 스위치 관련 많은 업체들이 이 표준에 맞는 스위치를 개발하고 있다[1].

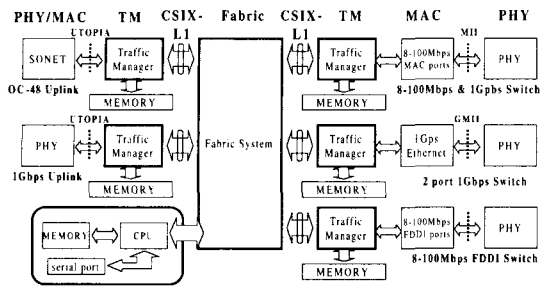
CSIX-L1의 경우 OC-48(2.488Gbps)급에 가장 적합한 인터페이스 규격을 제공하며, OC-192 (9.6Gbps)급의 인터페이스에도 적용이 가능하다[2]. 그러나 CSIX-L1의 경우 최대 32Gbps까지의 동작을 지원하지하므로 40Gbps급 인터페이스인 OC-768의 경우 적용이 불가능하다. 따라서 이에 대한 새로운 CSIX

- 1) 본 연구는 과학기술부·산업자원부의 “시스템 IC 기반기술 개발 사업”의 지원에 의해 이루어졌음.
- 2) 일부 연구는 IDEC의 CAD툴 지원에 의해 이루어졌음.

\* 정희원 : 호남대학교 컴퓨터공학과 석사과정  
junari@hanmail.net

\*\* 정희원 : 한신대학교 정보통신학과 조교수  
saisonh@hanshin.ac.kr

\*\*\* 종신회원 : 한신대학교 정보통신학과 부교수  
kangmg@hanshin.ac.kr



(그림 1) CSIX를 사용하는 라우터의 구조

규격이 향후 정의되어야 할 것으로 예측된다.

그림 1은 스위치와 라인카드의 TM(Traffic Manager) 사이에 CSIX 표준 인터페이스를 사용하는 라우터의 구조를 보인 것이다.

본 논문에서는 CSIX의 데이터 유닛인 CFrame의 기본 구조와 CSIX의 인터페이스에 대해 알아보고, CSIX 모듈을 설계한다. 설계되는 CSIX 모듈은 VHDL 언어를 사용하여 기술되었으며[3], 기술된 내용은 Xilinx Foundation 툴을 사용하여 합성한 후, FPGA로 구현하여 타이밍 시뮬레이션을 수행하였다[4,5]. 구현한 FPGA를 PC를 통해 검증하기 위해 테스트 보드를 설계하였다. 설계한 테스트 보드는 PCI 인터페이스를 통해 PC에서 소프트웨어로 검증하도록 하였는데, PCI 인터페이스 칩으로는 PLX9050 칩이 사용되었으며, 테스트 프로그램은 MFC를 사용하였다[6,7].

## 2. CFrame의 구조

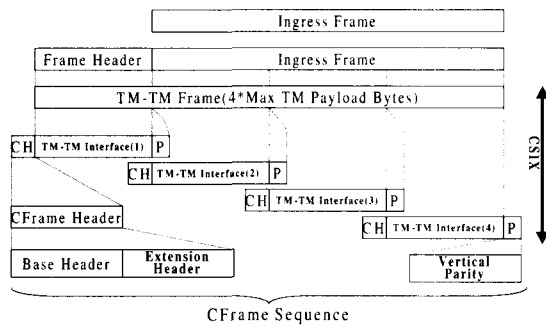
CFrame(CSIX Frame)은 TM들과 스위치 패브릭 사이에 전송되는 정보 유닛으로 그림 2와 같이 하나의 CFrame은 2~6bytes의 Header, 페이로드, 그리고 16bits의 Vertical Parity 필드로 구성된다.

CFrame Header는 CSIX 패브릭 인터페이스에서 TM의 동작 제어에 필요한 정보필드를 포함한다.

CFrame의 Header는 Base Header와 Extension Header로 분할 될 수 있다. 페이로드는 길이가 가변적이고 TM의 입력으로부터 TM의 출력으로 CSIX 패브릭에 의해 전달된다. 페이로드의 포맷은 현재 CSIX-L1

CFRAME COMPONENT	Base Header	Extension Header	Payload	Vertical Parity
LENGTH	2 bytes	0-4 bytes	Maximum allowable length is 256 bytes	2 bytes
COMMENTS		Number of bytes determined by CFrame and address type	Features of the traffic manager and fabric determine actual maximum length.	Field is required ; use of field is optional

(그림 2) CFrame의 구조



(그림 3) CFrame Sequence의 예

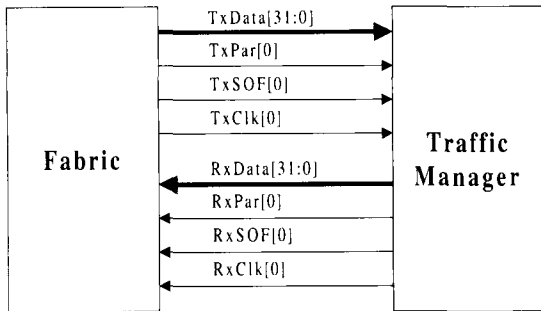
에서는 정의되어 있지 않으나, TM간의 인터페이스로 TM Header와 TM Payload같은 영역을 포함하며, 향후 CSIX-L2에 의해 정의될 예정이다.

Vertical Parity는 CSIX 페이로드의 에러 검출을 위해 사용되는 필드이다.

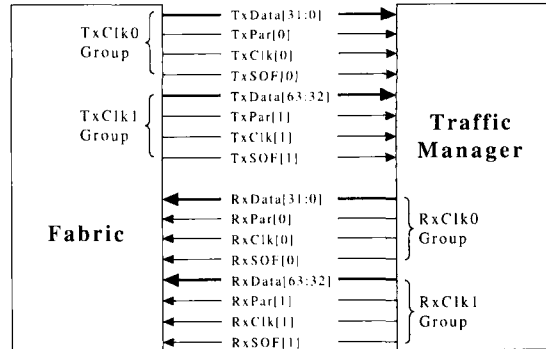
CFrame 페이로드의 크기는 최대 256bytes이고, 생성된 CFrame의 최대 크기는 264bytes(최대 6bytes의 Header, 최대 256bytes의 페이로드, 그리고 2bytes의 Vertical Parity)이다. CFrame의 크기(패딩포함)는 CWord의 크기에 따라 분할 될 수 있는데, CWord는 하나의 버스 사이클에서 전송된 데이터로 32-bit 인터페이스를 갖는 CSIX의 경우 CWord는 32-bit 이다.

TM-TM 인터페이스 메시지 길이가 최대 CFrame 페이로드보다 작거나 같을 때, 하나의 CFrame은 입력으로부터 출력 TM까지 정보를 전송하는데 사용될 수 있다.

TM-TM 인터페이스 메시지 길이가 최대 CFrame 페이로드 보다 클 때, 다중 CFrame들은 입력으로부터 출력 TM까지 반드시 순서적으로 전송되어야 한다. CFrame들의 정리된 집합은 CFrame Sequence로써 정리된다. 그림 3에서는 CFrame Sequence의 예를 볼 수 있다.



(그림 4) 32-bit 인터페이스



(그림 5) 64-bit 인터페이스

### 3. CSIX의 인터페이스

CSIX-L1은  $n \times 32\text{-bit}$ ( $n=1,2,3,4$ )의 데이터 패스를 이용할 수 있으며 동작 주파수는 최대 250MHz를 규정하고 있다. CSIX의 인터페이스 시그널들을 보면 Tx그룹 신호들은 스위치 패브릭으로부터 TM까지 정보를 전달하고, Rx그룹 신호들은 TM으로부터 스위치 패브릭까지 정보를 전달하는데 사용된다.

그림 4는 32-bit 인터페이스를 갖는 CSIX의 인터페이스 신호들을 나타낸 것이다.

TxData는 스위치 패브릭이 TM으로 전달하는 데이터를 실는 버스이다. TxPar는 스위치 패브릭에서 TM으로 전달하는 32-bit 데이터의 Odd parity (horizontal parity)를 나타낸다. TxClk은 스위치 패브릭이 TxData의 전송 동기화를 위해 TM에게 제공하는 데이터 전송 동기화 클럭이다. 하나의 32-bit TxData 그룹에 하나에 TxClk이 요구된다. TxSOF은 스위치 패브릭이 전송하는 CFrame의 처음 데이터를 나타낸다. 각각의 TxClk 클럭 그룹에 대해 하나의 TxSOF 신호를 요구한다.

RxData는 TM가 스위치 패브릭으로 전달하는 데이터를 실는 버스이다. RxPar는 TM에서 스위치 패브릭으로 전달하는 32-bit 데이터의 Odd parity (horizontal parity)를 나타낸다. RxClk은 TM이 TxData의 전송 동기화를 위해 스위치 패브릭에게 제공하는 데이터 전송 동기화 클럭이다. 하나의 32-bit TxData 그룹에 하나에 RxClk이 요구된다.

RxSOF은 TM이 전송하는 CFrame의 처음 데이

터를 나타낸다. 각각의 RxClk 클럭 그룹에 대해 하나의 RxSOF 신호를 요구한다.

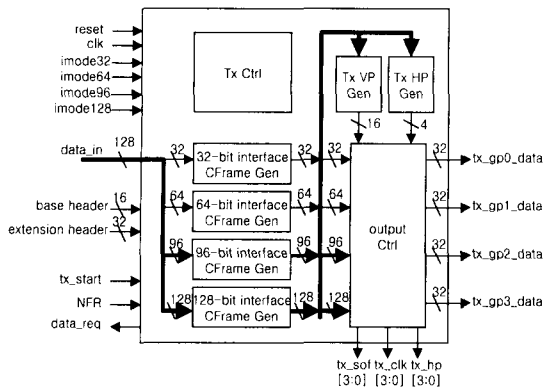
32-bit 인터페이스 이외의 인터페이스(64,96, 128-bit)는 송수신단의 데이터 패스를 32-bit단위로 그룹을 나누어 각각의 그룹에 Par, Clk, SOF 신호들이 추가된 형태이다. 그림 5에 64-bit 인터페이스를 도시하였다.

설계된 CSIX 모듈의 송수신단은 각각 스위치 패브릭에서는 Tx그룹이 전송단이 되고 Rx그룹은 수신단이 된다. 그리고 TM에서는 Rx그룹이 전송단이 되며, Tx그룹은 수신단이 된다.

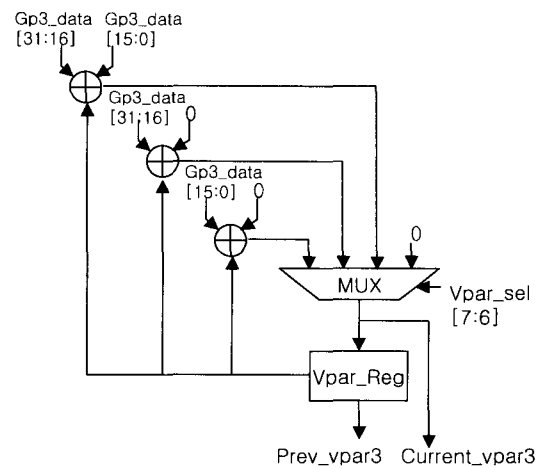
### 4. CSIX 모듈의 설계

CSIX 모듈은 크게 전송단과 수신단으로 나눌 수 있다. 그 중에 전송단은 전송하고자하는 데이터를 입력받아 설정된 인터페이스 모드에 맞게 CFrame을 생성한 후, 출력단을 통해 데이터를 전송하게 된다. 전송단의 구성은 그림 6과 같다.

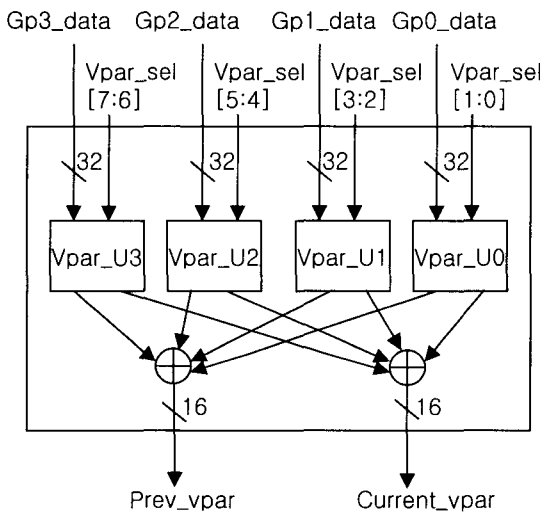
먼저 32-bit interface CFrame Gen 블록은 32-bit interface로 CSIX 모듈이 작동할 때, 32-bit interface에 맞게 CFrame을 생성해주는 블록으로 imode32가 high일 때, 작동한다. 이와 같이 64-bit interface CFrame Gen 블록은 64-bit interface일 때 작동하며, 96-bit interface CFrame Gen 블록은 96-bit interface일 때, 128-bit interface CFrame Gen 블록은 128-bit interface일 때 각각 작동한다. Tx VP Gen 블록은



(그림 6) 전송단의 구조



(그림 8) Vpar\_U3 블록의 구조

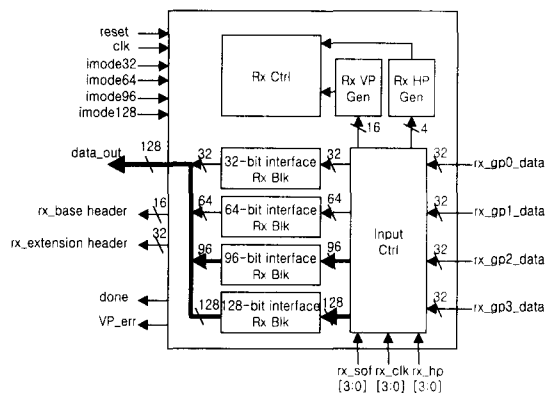


(그림 7) Tx VP Gen 블록의 구조

Vertical Parity를 생성하는 블록으로 그림 7과 같은 구조를 가지고 있으며, 네 개의 인터페이스를 모두 지원한다. 그림 7의 Vpar\_U3 블록은 그림 8에 도시 하였으며, Vpar\_U0, Vpar\_U1, Vpar\_U2 블록 모두 Vpar\_U3와 같은 구조로 되어 있다.

Tx HP Gen 블록은 인터페이스 모드에 따라 현재 클럭 사이클에 전달되는 데이터의 Horizontal Parity를 생성하는 블록으로 네 개의 인터페이스 모드를 모두 지원한다.

Tx Ctrl 블록에서 전체 전송단을 제어하며, Output Ctrl 블록에서는 CSIX 모듈의 출력 신호들을 관리한다. 데이터 입력은 32-bit interface 모드일 때, tx\_



(그림 9) 수신단의 구조

csix\_data[127:96]을 통해서 입력된다. 64-bit interface 모드일 때는 tx\_csix\_data[127:64], 96-bit interface 모드일 때는 tx\_csix\_data[127:32], 128-bit interface 모드일 때는 tx\_csix\_data[127:0]을 통해 입력된다. 그리고 Base Header와 Extension Header는 각각 base\_header[15:0]와 ex\_header[31:0]을 통해 전달된다. 이렇게 해서 헤더와 데이터를 입력받으면, CSIX모듈은 설정된 인터페이스 모드에 따라 CFrame을 생성하여 출력한다.

CSIX 모듈의 수신단은 상대측에서 전송된 CFrame에서 데이터를 추출하고, 데이터의 오류 여부를 알려준다. 그림 9는 수신단의 구조를 나타낸다.

Rx VP Gen 블록은 수신한 CFrame의 Vertical

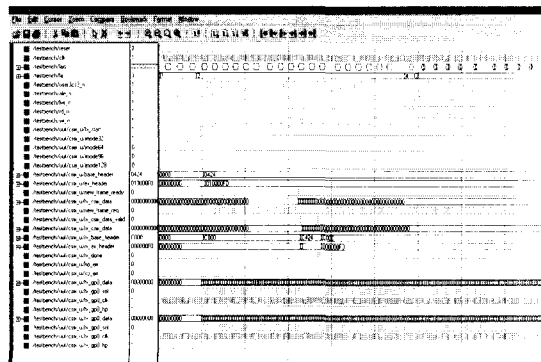
Parity를 계산하여 인터페이스 모드 신호에 의해 동작 중인 Rx Blk로 전달한다. 그러면 Rx Blk는 수신한 CFrame의 Vertical Parity와 Rx VP Gen 블록에서 전달된 Vertical Parity를 비교하여 Vertical Parity Error 여부를 판단한다. Rx VP Gen 블록은 전송단의 Tx VP Gen 블록과 같은 구조를 가지고 있다.

Rx Blk는 네 개의 인터페이스 모드를 각각 지원하는 블록들로 나뉘어 있다. 인터페이스 모드에 따라 해당 Rx Blk가 동작하는데, 수신한 CFrame에서 Base Header와 Extension Header를 레지스터에 저장하고, Payload는 rx\_csix\_data\_valid신호와 함께 rx\_csix\_data를 통해 출력된다. 이때 32-bit interface이면, rx\_csix\_data[127:96]에서 32-bit 데이터가 출력되며, 64-bit interface일 때는 rx\_csix\_data[127:64]에서 64-bit 데이터가 출력된다. 그리고 96-bit interface일 때는 rx\_csix\_data[127:32]를 통해 96-bit 데이터가 출력되며, 128-bit interface일 경우에는 rx\_csix\_data [127:0]을 통해 128-bit 데이터가 출력된다. 만약 수신한 CFrame에서 Horizontal Parity Error가 발생하거나 Vertical Parity Error가 발생하면, Rx Blk에서 hp\_err이나 vp\_err를 high로 출력한다.

Rx Ctrl 블록에서는 전체 수신단의 제어를 담당한다.

### 5. CSIX 모듈의 검증 및 성능 분석

본 논문에서는 설계한 모듈들은 FPGA에 구현

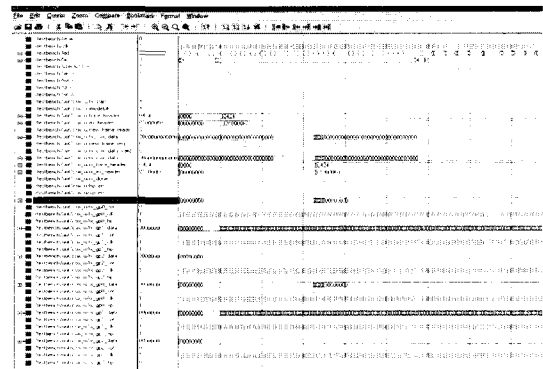


(그림 10) 32-bit interface 타이밍 시뮬레이션

하여 회로를 검증하였다.

Xilinx vertexe 타겟 라이러리를 사용하여 합성하고 P&R을 수행한 결과에 대한 post layout 시뮬레이션 즉, 타이밍 시뮬레이션을 수행한 결과는 그림 10과 그림 11에 나타내었다. P&R을 완료한 이후에 각 모듈의 타이밍 시뮬레이션을 수행하기 위해 본 논문에서는 동작 시뮬레이션을 수행할 때와 동일한 벡터를 사용하여 테스트를 수행하였다. 따라서 동작 시뮬레이션을 통해 얻은 결과와 타이밍 시뮬레이션을 수행하여 얻은 결과가 동일함을 확인함으로써 검증을 수행하도록 하였다.

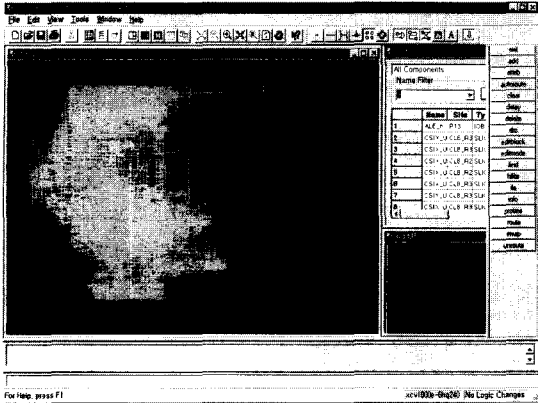
구현한 CSIX 모듈에 대한 결과를 표 1에 나타내었다. 결과는 각각 FPGA로 직접 구현하였을 때와 ASIC으로 구현하였을 경우 예상결과를 비교하여 나열하였다. 사용한 FPGA 유형은 XCV1000



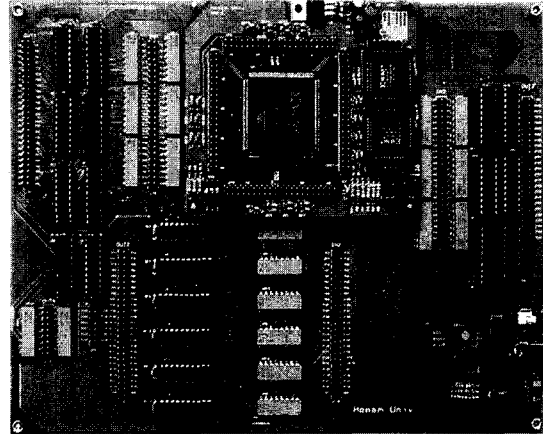
(그림 11) 64-bit interface 타이밍 시뮬레이션

(표 1) CSIX 모듈의 사양 및 성능 분석

FPGA	Target Library	vertexe(System Library V.1.37)		
	FPGA Type	XCV1000E6H240C		
	# of External I/O	240 핀		
	# of gates	333,539(include test module)		
ASIC	Frequency	27MHz		
	Frequency	200MHz		
	Performance	32-bit	6.4Gbps	
		64-bit	12.8Gbps	
		96-bit	19.2Gbps	
128-bit		25.6Gbps		



(그림 12) CSIX 모듈의 레이아웃



(그림 13) 테스트 보드

EHQ240C이며, 외부 입출력은 240핀이다. 그리고 합성 결과 게이트 수는 333,539개로 이는 테스트 모듈을 포함한 것이다. 만약에 ASIC 칩으로 구현 하였을 때 권고안의 성능인 200MHz를 만족할 수 있을 것으로 예견된다. 이 때 인터페이스 모드에 따라 각각 6.4Gbps(32-bit interface), 12.8Gbps(64-bit interface), 19.2Gbps (96-bit interface), 25.6Gbps(128-bit interface)로 데이터의 송수신이 가능할 것이다.

그림 12는 FPGA로 구현한 CSIX 모듈의 레이아웃을 나타내고 있다.

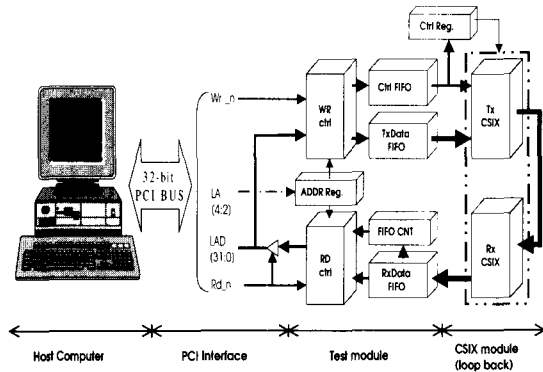
## 6. 보드레벨 테스트

본 논문에서는 시뮬레이션뿐만 아니라 테스트 보드를 설계하여 FPGA로 구현한 CSIX 모듈을 PCI 카드를 통해 PC의 소프트웨어를 통한 검증을 수행하였다. 그림 13은 테스트 보드를 나타내었다.

그림 14는 PCI인터페이스를 통해 구현한 CSIX 모듈을 검증할 수 있도록 구현한 데모 시스템을 나타내고 있다.

오른쪽의 블록도는 FPGA에 구현된 것으로 설계한 CSIX모듈과 테스트 모듈로 구성된다.

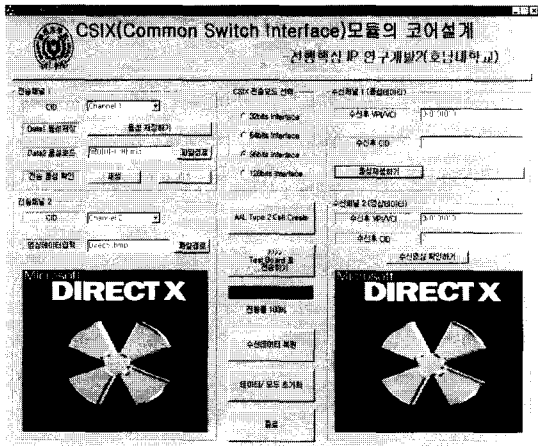
테스트 모듈에서 WR\_n, LA, LAD, RD\_n 등은 PCI 카드(PLX9050칩 사용)와 테스트 보드사이에 인터페이스 신호들이다. LAD를 통해 들어오는 데이터는 LA의 값에 따라 CSIX Ctrl과 TxFIFO로 전



(그림 14) 데모 시스템

달되게 된다. CSIX Ctrl은 PCI 카드를 통해 들어오는 CSIX 모듈의 컨트롤 정보를 가지고, CSIX 모듈의 제어 신호(imode32, imode64, imode96, imode128, new\_frame\_ready)들을 생성한다. 그리고 TxFIFO에는 실제 CSIX 모듈을 통해 전송하고자 하는 데이터들이 저장된다.

CSIX Ctrl에서 출력되는 제어 신호에 따라 CSIX 모듈에서 생성된 CFrame은 Tx 그룹 신호선들을 통해 출력하게 되는데, 이 Tx 그룹의 신호를 CSIX 모듈 자신의 Rx 그룹 신호로 연결하여 CFrame을 수신하게 된다. 수신된 CFrame은 RxFIFO에 저장된다. PC에서는 테스트 프로그램을 통해 전송한 데이터와 수신한 데이터를 서로 비교함으로써 검증을 수행하였다. 그림 15는 설계한 테스트 프로그램을



(그림 15) 테스트 프로그램

보이고 있다.

테스트 프로그램의 좌측은 전송할 데이터를 입력하는 부분으로 음성 파일과 이미지 파일을 선택하도록 되어 있다. 입력한 음성 파일은 재생 버튼을 통해 확인 할 수 있으며, 이미지 파일은 파일 오픈과 동시에 하단 이미지 창에 이미지가 나타나게 된다. 파일을 선택한 후 가운데 상위에 있는 CSIX 전송 모드에서 전송하고자 하는 CSIX Interface 모드를 선택한다. 전송 버튼을 누르기 전에 그 위에 있는 ATM Cell 생성 버튼을 누른다. 그러면, 오픈된 두 개의 파일 내용을 ATM Cell로 가공하여 ATM Cell 단위로 전송을 할 수 있다. 전송 버튼을 누르면, 전송한 데이터는 PCI버스를 통해 테스트 보드로 전달된다.

전달된 전송 데이터(ATM Cell)은 테스트 모듈을 통해 CSIX의 전송단으로 전달되며, CSIX 전송단에서는 전달된 데이터를 CSIX의 정보 유닛인 CFrame으로 가공한 후 수신단 CSIX 모듈에 전달한다. 수신단에서는 수신한 CFrame에서 실제 데이터(ATM Cell)을 추출하여 테스트 모듈을 통해 PC의 테스트 프로그램으로 전달한다.

테스트 프로그램에서 수신한 데이터는 ATM Cell 형태이므로 “수신데이터 복원” 버튼을 눌러 원래 음성 정보와 이미지 정보로 복원한다. 그 다음 우측에 있는 “음성재생하기” 버튼 및 “수신영

상확인하기” 버튼을 누르면 결과를 확인 할 수 있다. 그리고 좌측의 음성 및 이미지를 비교하여 오류 여부를 확인할 수 있다.

## 7. 결 론

본 논문에서는 스위치 패브릭과 네트워크 프로세서의 인터페이스 표준으로 자리잡고 있는 CSIX 모듈을 설계하였다. 설계한 CSIX 모듈은 4개의 인터페이스(32, 64, 96, 128-bit)를 지원한다. 그리고 Vertical Parity 계산 및 검증 기능, Horizontal Parity 계산 및 검증 기능, 패딩 바이트 부착 기능을 제공한다.

전체적인 모듈에 대해 시뮬레이션을 완료하고 합성 및 P&R을 완료하였으며 타이밍 시뮬레이션을 완료하였다.

FPGA로 구현을 위해 사용한 Xilinx 칩은 XCV 1000E HQ240C이며, 입출력 핀은 240 핀이다. CSIX 모듈의 게이트 수는 333,539개이며, 동작 주파수는 CSIX 모듈은 27MHz에서 동작하였다.

또한 보드 레벨에서 검증을 위하여 테스트 보드를 제작하였으며, 이를 통해 PC에서 설계한 CSIX 모듈간에 PCI버스를 통해 사용자 데이터 파일을 전송하고 수신하여 각 CSIX 모듈에 대한 검증을 수행하고 있다.

본 논문에서는 또한 CSIX 모듈에 대한 Soft IP를 설계하여 ATM, Ethernet 및 SONET 등과 같은 네트워크 및 통신 분야에서 효율적으로 이용될 수 있을 것으로 예상된다.

## 참 고 문 헌

- [1] 이형호, 김봉완, 안병준, “테라비트 라우터 기술”, Telecommunication Review. 제11권 2호, pp. 237~247, Mar. 2001.
- [2] “CSIX-L1 : Common Switch Interface Specification-L1”, www.csix.org Aug. 2000.
- [3] 박현철, “HDL을 이용한 반도체 회로 설계”, 한성출판사, 2001년 6월.

- [4] David Van Den Bout 원저, 김만복 번역, "FPGA DESIGN 이론 및 실습", 홍릉과학출판사, 2000년 9월.
- [5] Janick Bergeron, "WRITING TESTBENCHES Functional Verification of HDL Models" KLUWER ACADEMIC PUBLISHERS.
- [6] "PCI 9050-1 Data Book", www.plxtech.com, April 17, 1997.
- [7] 차태호, 전춘기, "비주얼C++에 의한 인터페이스 응용", 북두출판사, 2001년 3월.
- [8] 진종암, 변성혁, 안병준, 이형호, "테라비트 라우터 기술 동향," 대한전자공학회지, 제28권, 제9호, pp. 50-59, 2001년 9월.
- [9] M. Karol, M. Hluchyj, and S. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch." IEEE Tr. Commun., Vol.COM-35, no.12, pp. 1347~1356, 1987.
- [10] 손장우, "Input Queueing Switch", <http://netmanias.com/contents/whitepaper/son>, Mar. 2001.
- [11] M. G. Hluchyj and M. J. Karol, "Queueing in High-performance Packet Switching." IEEE JSAC, vol. 6, no.9, pp. 1587~1597, Dec. 1988.



◎ 저자 소개 ◎



**김 형 준**

2001년 호남대학교 컴퓨터공학과 졸업(학사)  
2001년~현재 : 호남대학교 컴퓨터공학과 석사과정  
관심분야 : ATM 통신, 네트워크, etc.  
E-mail : junari@hanmail.net



**손 승 일**

1989년 연세대학교 전자공학과 졸업(학사)  
1991년 연세대학교 대학원 전자공학과 졸업(석사)  
1998년 연세대학교 대학원 전자공학과 졸업(박사)  
1998년~2002년 호남대학교 컴퓨터공학과 조교수  
2002년~현재 : 한신대학교 정보통신학과 조교수  
관심분야 : ATM 통신 및 보안, ASIC 설계, etc.  
E-mail : saisonh@hanshin.ac.kr



**강 민 구**

1986년 연세대학교 전자공학과 졸업(학사)  
1989년 연세대학교 대학원 전자공학과 졸업(석사)  
1994년 연세대학교 대학원 전자공학과 졸업(박사)  
1985년~1987년 삼성전자 연구원  
1997년~1998년 일본 오사카대학 객원연구원  
1994년~2000년 호남대학교 정보통신공학과 조교수  
2000년~현재 : 한신대학교 정보통신학과 부교수  
관심분야 : 이동통신, 무선인터넷, 암호이론, etc.  
E-mail : kangmg@hanshin.ac.kr