

차세대 통신 플랫폼을 위한 입출력 컨트롤러 설계 및 검증

현유진*, 이정현**, 오현석*, 성광수***

*영남대학교 전자공학과 집적회로연구실, **한전 KDN (주)

**영남대학교 전자정보공학부

e-mail:braham@yumail.ac.kr

Design of I/O Controller for Future Communication Platform

Eugin Hyun*, Jung-Hyon Lee**, Hyun-Seok Oh*, Kwang-Su Seong**

*VLSI Lab, Dept of Electronic Engineering, Yeungnam University

**Korea Electric-power Data Network Co. LTD

***Dept of Electrical & Computer Science, Yeungnam University

요 약

본 논문에서는 차세대 통신 플랫폼을 위한 PCI Express의 전송계층과 데이터 연결계층의 모든 기능을 지원하는 PCI Express 컨트롤러를 설계하였다. 설계되어진 컨트롤러는 재전송 매커니즘을 효과적으로 지원하기 위해 제안되어진 송신버퍼 구조를 가지고 있다. 이 버퍼 구조는 전송 버퍼와 재전송 버퍼를 한 개의 버퍼로 통합하여 재전송 버퍼의 공간을 유동적으로 할당할 수 있는 방법이다. 또한 설계되어진 컨트롤러의 송신단 전송계층은 제안되어진 버퍼 구조 효과적으로 지원하도록 설계되어 졌다. 설계 되어진 컨트롤러의 각 블록을 효과적으로 관리하기 위해 80C51 마이크로프로세서를 내장하여 PCI Express 프로토콜을 제공하는 프로그램을 코딩하여 포팅하였다. 또한 설계되어진 컨트롤러의 검증을 위해, Host Bridge, 로컬 마스터 디바이스, 로컬 슬레이브 디바이스를 버스 동작 모델로 구성된 테스트 벤치도 제안하였다. 또한 실제 PCI Express 프로토콜 상에서 발생할 수 있는 모든 경우를 발생 하도록 하기 위해, 각 버스 동작 모델을 위한 어셈블러 명령어들을 정의 하였다.

1. 서론

오늘날 통신 기술과 컴퓨터 기술은 하나로 통합되고 있다. 인터넷과 네트워크 사용의 폭발적인 증가로 인해 일반적인 PC는 물론이고 노트북 그리고 PDA에 이르기까지 유무선 통신 기술이 시스템의 한 부분으로 내장되고 있다^[1]. 반대로 통신 시스템 역시 효과적인 시스템 제어를 위해 컴퓨터 기술을 내장한다^[1]. 하지만 지금까지 통신 및 임베디드 솔루션은 칩 간 혹은 시스템간의 연결에 있어 다양한 인터페이스를 사용한다. 비록 컴퓨터 시스템은 PCI와 같은 입출력 표준안을 인터페이스로 지원해왔다. 결국 이는 통신 시스템과 컴퓨터 시스템의 상호 연결의 문제점을 야기 시켜 코딩 및 물리적 설계 까지도 복잡하게 만들게 됨으로 인해 시스템 전체의 가격을 증가시킨다. 따라서 통신 및 컴퓨터 시스템의 통합 발전을 위해서는 보다 간단한 인터페이스 표준안이 필요하게 되었다^{[1][2]}. 게다가 오늘날의 통신 및 임베디드 시스템에는 비디오 및 오디오의

streaming data의 실시간 처리가 요구됨에 따라 높은 데이터 전송률이 요구된다. 하지만 여러 디바이스가 하나의 버스를 공유하는 병렬버스 구조로 되어있는 기존의 PCI 인터페이스는 이러한 요구를 충족시킬 수 없다^{[1][2]}. 이에 PCI SIG에서는 차세대 컴퓨터 및 통신 시스템을 위한 새로운 입출력 표준안으로 PCI Express를 소개하였다. PCI Express는 point-to-point 방식을 이용한 직렬 전송 방식으로 현재 2.5Gpbs의 전송 속도를 가진다. 또한 기존 PCI 장치에 사용되는 OS와 디바이스 드라이버 S/W를 그대로 사용할 수 있어 차세대 컴퓨터 및 통신 시스템의 입출력 표준안으로 자리 잡을 것이다^{[4][5]}.

그림 1은 PCI Express를 이용한 통신 시스템의 구성도 이다. PCI Express 시스템은 Host Bridge, Switch, 그리고 여러개의 Endpoint로 구성된다. Endpoint는 PCI Express에 연결된 모든 입출력 디바이스로서 Gb 이더넷 카드, 그래픽 컨트롤러, PCI 2.2

및 PCI-X 브리지, 라인 카드 등을 예로 들 수 있다.

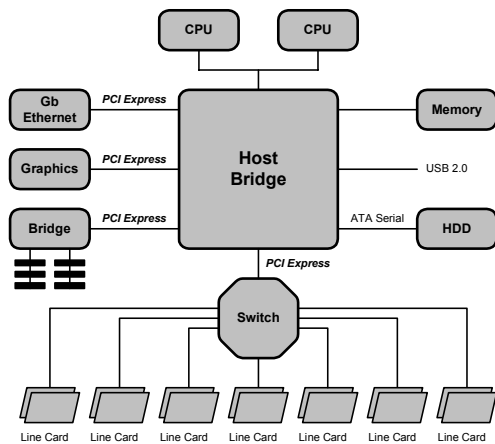


그림 1. PCI Express를 이용한 네트워킹 통신 시스템 구성도
본 논문에서는 PCI Express Endpoint 컨트롤러를 설계하였고 검증을 위한 테스트 벤치도 제안하였다.

2. PCI Express 개요

PCI Express는 그림 2(a)와 같이 전송계층, 데이터 연결계층, 물리 계층으로 구성된 계층 구조를 가진다. 디바이스 코어로부터 요청되어진 데이터는 송신단의 전송계층에 의해 패킷으로 생성되는데 헤더, 데이터, 그리고 ECRC(End to End CRC)로 구성되며 이를 TLP(Transaction Layer Packet)라 한다. 데이터 연결계층에서는 신뢰성을 확보하기 위해 전송하는 각 TLP의 일련번호를 순차적으로 첨부한다. 그리고 헤더, 데이터, ECRC, 일련번호를 이용하여 계산한 LCRC도 TLP에 붙게 된다. 물리 계층은 데이터 연결계층으로부터 받은 TLP에 시작과 끝을 알리는 프레임 정보를 붙여 직렬 정보로 변환한 다음 외부로 전송한다.

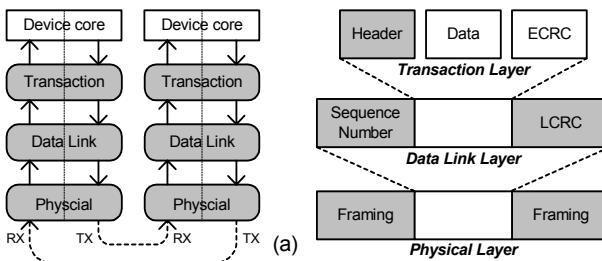


Figure 2. PCI Express의 계층 구조 및 패킷 구조

PCI Express의 송신단은 TLP를 전송하기 전에 수신 디바이스가 전송하려는 데이터를 받을 수 있는 버퍼 공간이 충분한지를 먼저 확인한다^{[4][5]}. 따라서 이러한 흐름제어(Flow control)를 지원하기 위해 수신 디바이스의 데이터 연결계층은 자신의 수신버퍼 크기를 FC DLLP(Data Link Layer Packet)를 이용하여 주기

적으로 링크의 반대편 디바이스에 알려주어야 한다. 여기서 DLLP는 데이터 연결계층에 의해 생성되어지는 패킷이다. 이러한 흐름제어를 지원하기 위해서는 수신단은 *Posted Request*, *Non-Posted Request*, 완성 요청(*Completion Request*)을 위한 각각의 수신버퍼를 가지고 있어야 한다^{[4][5]}. *Posted Request*는 메모리 쓰기 명령어와 메시지 전송 명령어이다. *Non-Posted Request*는 모든 메모리 읽기 명령어와 PCI 디바이스 내부 컨피규레이션(Configuration) 레지스터에 접근하기 위한 명령어이다. 메모리 읽기 명령어를 수신한 디바이스는 데이터를 준비하여 이를 원래 요청한 디바이스에 전송하게 되는데 이를 완성 요청이라 한다.

링크 반대편의 수신 디바이스가 송신 디바이스로부터 TLP를 수신한 경우, 수신 디바이스의 데이터 연결계층은 수신한 TLP의 일련번호와 LCRC의 오류를 검사한다. 만약 에러가 발생하지 않았다면 정상적으로 TLP를 수신하였음을 원래의 송신 디바이스에 알리기 위한 승인 절차로 패킷을 전송하게 되는데 이를 ACK(Acknowledge) DLLP라고 한다. 수신단은 한 개의 TLP를 받을 때 마다 ACK DLLP를 전송하는게 아니라 일정 시간을 기다렸다가 가장 마지막으로 수신된 TLP의 일련번호를 붙여 전송하게 된다. 만약 수신한 TLP에 오류가 있는 경우 NACK(Negative ACK) DLLP를 전송하게 되는데 이를 받은 송신 디바이스의 데이터 연결계층은 상대 디바이스에 의해 승인되지 않은 모든 TLP들을 다시 전송하게 되는데 이를 재전송 메커니즘이라 한다^{[4][5]}. 또한 송신 디바이스가 TLP를 전송한 후 일정 시간이 지나도 ACK DLLP를 받지 못한다면 이때도 재전송을 해야 한다.

이러한 재전송 메커니즘을 지원하기 위해 데이터 연결계층은 송신버퍼와 별도로 재전송 버퍼를 그림 3과 같이 두어야 한다. 전송 버퍼에 있는 TLP는 물리 계층으로 전송되어진 후 재전송 버퍼에 백업되어진다. 만약 재전송을 해야 할 상황이 발생한다면 재전송 버퍼에 있는 모든 TLP는 다시 전송되어질 것이다. 그러나 상대 디바이스로부터 ACK DLLP를 받는 경우 재전송 버퍼에 있는 해당 TLP들은 모두 폐기되어진다.

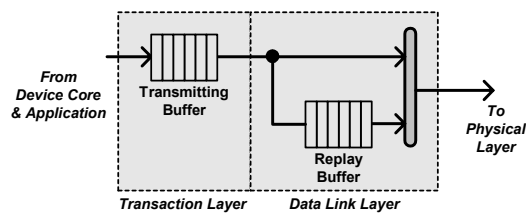


그림 3. 재전송 기법을 지원하는 개념적인 버퍼 구조

3. 설계되어진 컨트롤러

본 논문에서는 PCI Express 전송계층과 데이터 연결계층의 기능을 지원하는 컨트롤러를 그림 4와 같이 설계하였다. APCE라고 명칭을 정한 이 컨트롤러는 송신단 전송계층(TxTL), 송신버퍼(TxBf), 송신단 데이터 연결계층(TxDL), 수신단 전송계층(RxTL), 수신버퍼(RxBf), 수신단 데이터 연결계층(RxDL), 마이크로프로세서(uP), 컨피규레이션 레지스터로 구성 되어 있다.

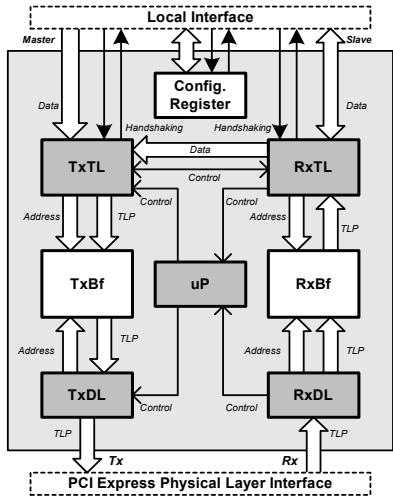


그림 4. APCE의 구성도

송신단의 데이터 연결계층은 재전송 메커니즘을 지원하기 위해 그림 4와 같이 재전송 버퍼를 가지고 있어야한다. 그러나 이는 간단한 구조를 가짐에도 불구하고 두개의 버퍼가 분리 되어 있음으로 인해 전송 효율 및 버퍼 사용 효율이 떨어지게 된다^[6]. 그래서 단지 하나의 버퍼를 이용하여 재전송 버퍼 공간을 유동적으로 사용할 수 있는 방법을 그림 5와 같이 제안하였다^[6]. 전송계층은 WSP(Write Start Point)가 가리키는 공간에 새로운 패킷을 저장하면 된다. 데이터 연결계층은 TSP(Transmit Start Point)가 가리키는 공간에 저장되어있는 TLP를 물리계층으로 전송하면 된다. 그리고 RSP(Replay Start Pointer)부터 TSP-1까지 저장된 패킷들은 재전송 되어질 TLP들이다.

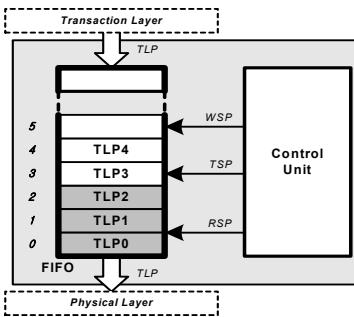


그림 5. 제안된 송신 버퍼의 구조

데이터 연결계층은 32비트 CRC와 16비트 일련번호를, 물리계층은 8비트 시작 프레임과 끝 프레임을 전송할 TLP에 첨부하여야 한다. 만약 재전송 상황이 발생하면 데이터 연결계층은 재전송할 TLP에 일련번호를 다시 붙여야 하기 때문에 아직 승인되지 않은 TLP들의 모든 일련번호를 따로 저장하고 있어야 함으로 인해 설계상의 어려움이 있다. 그래서 APCE의 전송계층은 그림 6과 같이 LCRC, 일련번호, 시작 프레임, 그리고 끝 프레임을 모두 이용하여 TLP를 생성하여 송신 버퍼에 저장하도록 하였다. 또한 이 패킷은 32비트로 정렬되어 있어 버려지는 버퍼 공간을 없앨 수 있다.

Address	31		0
0	Header		Start Frame
1	1st Data		Header
...		1st Data
...	Last Data	
N-2	LCRC		Last Data
N-1	End Frame		LCRC

그림 6. TxTL에 의해 생성되어 TxBf에 저장되는 패킷의 형태

모든 PCI Express 수신 디바이스는 흐름제어를 지원하기 위해 그림 7과 같이 각 명령어 별로 모두 3가지 FIFO 버퍼를 가지고 있어야 한다. 하지만 이 경우 수신된 전체 패킷의 순서를 정확히 알 수가 없어 전송계층이 수신버퍼에 저장되어진 패킷을 순서대로 처리하는데 어려움이 있다. 따라서 APCE에서는 그림 7(b)와 같이 물리적으로는 FIFO 버퍼를 한개 만을 사용하고 3개의 명령어를 위한 공간을 추상적으로 각각 할당하였다. 그리고 각 명령어별로 카운터를 두어 각 명령어의 패킷이 한 개의 버퍼에 얼마나 차있는지를 항상 계산하여 이를 이용하여 흐름제어를 할 수 있다.

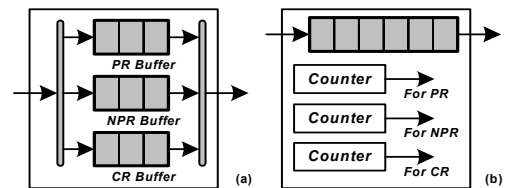


그림 6. (a) 개념적인 수신버퍼, (b) 제안되어진 수신버퍼

APCE는 각 블록을 효과적으로 제어하기 위해 80C51 마이크로프로세서를 내장하였다. 그리고 RxTL에 의해 수신되어진 메시지의 체크, 메시지를 생성 후 TxTL에 요청, RxTL로 수신된 ACK와 NACK DLLP의 처리, 수신된 TLP에 대한 ACK 혹은 NACK DLLP 전송을 TxDL에 요청, 재전송 지시, 흐름 제어 관리를 위한 프로그램을 코딩 후 porting하였다.

4. 검증 환경

설계되어진 APCE를 검증하기 위한 테스트 벤치는 그림 8와 같다. Host Bridge, 로컬 마스터, 로컬 슬레이브는 C 언어를 이용하여 행위 모델로 설계 되었다.

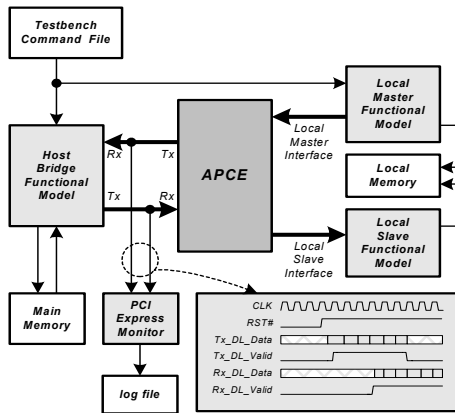


그림 8. 테스트 환경

또한 APCE의 효과적인 검증을 위해 어셈블리 명령어를 정의 하였다. 이 명령어는 크게 두 가지 타입으로 나누어진다. 첫 번째 타입은 데이터 전송을 명령하기 위한 것으로 APCE의 Host Bridge의 송신단과 로컬 마스터에 의해 사용되어진다. 두 번째 타입은 각 인터페이스에서 데이터 전송 시 발생할 수 있는 모든 상황을 설정하기 위한 파라미터를 설정하기 위한 명령어이다. 파라미터의 몇몇 예제가 아래에 소개되어 있다.

- Host Bridge와 로컬 마스터가 전송하려는 패킷을 생성하는데 필요한 정보.
- 로컬 인터페이스에서 데이터 전송 시 발생하는 대기 시간(wait phase)
- Host Bridge가 APCE로부터 TLP를 수신한 후 NACK DLLP를 송신할 확률.
- Host Bridge가 APCE로부터 요청되어진 메모리 읽기 명령어를 수신한 후 정상적인 완성요청을 전송할 확률.
- Host Bridge가 LCRC나 일련번호 에러를 가지고 TLP를 APCE에 전송할 확률 등등.

5. 시뮬레이션 결과

그림 9는 메모리 읽기 명령어를 수행한 시뮬레이션 결과이다. 시스템이 초기화 된 후 APCE와 Host Bridge는 흐름제어를 위해 수신버퍼 정보를 교환한다 (a). 먼저 Host Bridge가 APCE가 정상적으로 동작하는데 필요한 파라미터를 세팅하기 위해 컨피규레이션 쓰기 명령어를 시작한다(b). 이 명령어를 수신한 APCE는 얼마 후 이에 상응하는 완성 요청을 실행한다(c). 얼마 후 로컬 마스터는 로컬 인터페이스를 통해 메모리 쓰기 명령어를 시작하고(d) APCE는 TLP를 생성한 다음 이를 Host Bridge에 전송한다(e). Host Bridge는 얼마 후 APCE에 ACK DLLP를 전송한다.

6. 결론

본 논문에서는 PCI Express의 전송계층과 데이터 연결계층의 모든 기능을 지원하는 PCI Express Endpoint 컨트롤러인 APCE를 설계하였다. 또한 설계되어진 컨트롤러의 검증을 위해 테스트 벤치도 제안하였고 실제 PCI Express 프로토콜 상에서 발생할 수 있는 모든 경우를 발생 하도록 하기 위해 각 버스 동작 모델에 필요한 어셈블리 명령어들을 정의 하였다.

[1] Intel whitepaper, "Advanced Switching for the PCI Express Architecture", www.intel.com, 2002
 [2] Intel whitepaper, "Creating a PCI Express Interconnect", www.intel.com, 2002
 [3] http:// www.pcisig.com
 [4] PCI SIG, PCI Express Base Specifications Revision 1.0a, PCI SIG, 2003.
 [5] Ravi Budruk, Don Anderson, and Tom Shanley, PCI Express System Architecture, MindShare, 200
 [6] Eugin Hyun, Kwang-Su Seong, "The effective buffer architecture for data link layer of PCI express", ITCC2004, Vol. 1, p,p 809-813, April 2004.

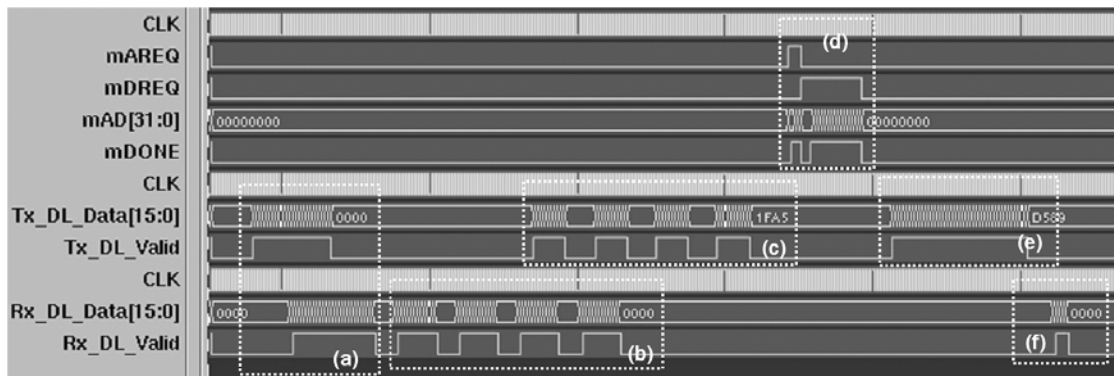


그림 9. 결과 파형