

## 멀티미디어 프로세서의 PCI 컨트롤러 디자인 및 검증

이준희(李俊熙), 남상준(南相俊), 김병운(金炳雲), 임연호(林堯濤),  
권영수(權榮洙), 경종민(慶宗旻)  
한국과학기술원 전기 및 전자공학과  
전화 : (042) 869-4402 / 팩스 : (042) 869-4040

### Design and Verification of PCI Controller in a Multimedia Processor

Jun-Hee Lee, Sang-Joon Nam, Byoung-Woon Kim, Yeon-Ho Im,  
Young-Su Kwon and Chong-Min Kyung  
Department of Electrical Engineering, KAIST  
E-mail : munjigi@vslab.kaist.ac.kr

#### Abstract

This paper presents a PCI (Peripheral Component Interconnect) controller embedded in a multimedia processor, called FLOVA (FLOating point VLIW Architecture), targeting for 3D graphics applications. Fast I/O interfaces are essential for multimedia processors which usually handle large amount of multimedia data. Therefore, in FLOVA, PCI bus is adopted for I/O interface due to fast burst transaction. However, there are several problems in implementation and verification to use burst transaction of PCI. It is difficult to handle data transaction between two units which have two different operating frequency. FLOVA has more higher operating frequency about 100MHz than that of PCI local bus and it makes lower utilization of FLOVA bus. Also, traditional simulation is not sufficient for verification of PCI functionality. In this paper, we propose buffering schemes to implement the PCI controller with wide bandwidth and high bus utilization. Also, this paper shows how to verify the PCI controller using real PCI bus environments before its fabrication.

#### I. 서론

멀티미디어의 발전은 멀티미디어 기능을 가진 많은

칩셋을 나오게 했다. 이러한 멀티미디어 칩셋은 일반적으로 PC의 add-on board 형태로 동작을 하게 되는데 이 때 시스템 성능을 결정하는 가장 큰 요인 중에 하나가 호스트 프로세서와 멀티미디어 프로세서간의 데이터가 얼마나 빠르게 전달 할 수 있는나 하는 것이다[1]. 빠른 데이터 전송을 목적으로 제안된 것 중 하나가 기존 ISA (Industry Standard Architecture) 방법과는 달리 데이터 버스 폭을 늘리고 동작 주파수를 증가시켜 주변장치들이 데이터 전달을 하는데 가지는 병목을 최소화 하기 위해 제안된 것이 PCI 버스 프로토콜이다[2][8][9].

PCI 버스는 멀티미디어 시스템에서 AGP (Advanced Graphic Port)와 같이 가장 많이 쓰이는 버스 프로토콜의 하나로 멀티미디어 프로세서의 경우 호스트 프로세서와 데이터를 고속으로 전송하기 위해 PCI 버스 컨트롤러를 내장하게 된다.

이러한 PCI 컨트롤러는 데이터 전송을 빨리 할 목적으로 만들어지기 때문에 내부 구조에 따라 성능에 크게 영향을 줄 수 있다[3]. 또한 서로 다른 두 버스의 프로토콜을 연결하고 두 버스간의 동작 주파수가 다름으로 인해 발생하는 병목을 최소화 할 수 있도록 디자인하여야 한다.

이 논문에서는 멀티미디어 프로세서인 FLOVA에 대해서 II장에서 소개를 하고 병목을 없애기 위한 구조에 대해 III장에서 다루어지며, IV장에서는 이를 검증하는 방법을 제시한다.

#### II. FLOVA의 개요

PCI 컨트롤러를 탑재하고 있는 멀티미디어 프로세서

인 FLOVA(FLOating-point VLIW Architecture)는 VLIW(Very Long Instruction Word) 구조로 제작되었다[5]. FLOVA는 사이클 당 4개의 명령어를 수행하며 주요 기능 유닛으로는 분기/컨트롤 유닛, 3개의 정수 ALUs와 한 개의 정수 곱셈기와 쉬프트 유닛, 두 개의 로드/스토어 유닛, IEEE 754를 지원하는 3개의 부동 소수점 연산기가 있다. 더불어 Intel MMX 명령어와 같은 sub-word 연산을 처리 할 수 있는 기능을 포함하고 있다[4]. 또한 8KB의 명령어 캐쉬와 두 개로 구성된 16KB의 내부 데이터 메모리를 가지고 있다. 그 외 DMA 기능과 타이머, PCI 컨트롤러를 내장하고 있다.

동작 주파수는 100MHz이며 64개의 32-bit 레지스터로 구성이 되어 있다. 이들 레지스터는 32-bit 또는 64-bit으로 읽고 쓰기가 가능하며 8개의 read-ports와 4개의 write-ports로 읽고 쓰기가 이루어지게 된다.

FLOVA의 부동 소수점 곱셈 연산기는 SIMD(Single Instruction Multi Data)의 구조와 cross mode 연산을 가능하게 함으로 3D그래픽에서 부동 소수점 연산이 많은 기하연산에 적합한 구조로 되어 있다[5]. 특히 기하연산이 vertex 단위로 구성된 부동 소수점 연산이 반복으로 이루어지고 있는데 FLOVA에서 지원하는 'for'명령어는 VLIW구조의 단점인 branch가 일어날 때 발생하는 penalty를 없앨 수 있다[6].

그림-1은 FLOVA의 구조를 나타내고 있다.

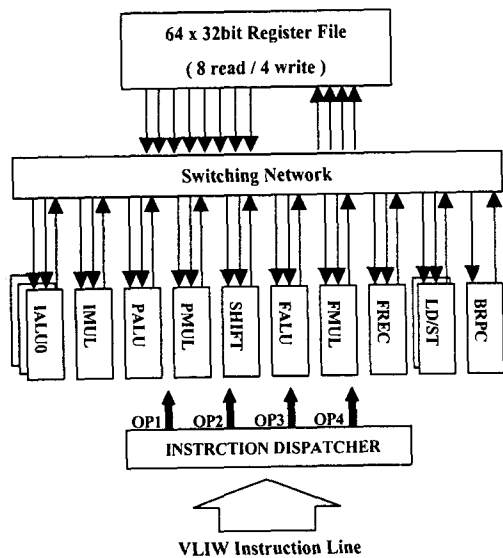


그림-1 FLOVA의 구조

### III. PCI의 기능과 구조

PCI 규정[9]에서 지원하는 최대 데이터 전송속도는 burst mode로 전송을 할 경우 132Mbytes/sec이다. 그러므로 기본적으로 제공하고 있는 단일 데이터 read/write뿐만 아니라 burst mode 데이터 read/write도 제공을 하고 있다. 또한 FLOVA가 멀티미디어 프로세서로 PC에 add-on으로 동작되어야 하기 때문에 외부 PC에 의해 초기화가 되는 PNP(Plug and Play) 기능 또한 제공을 하고 있다.

제작된 PCI는 slave mode로만 동작이 되며 FLOVA가 데이터를 외부로 전송하고자 할 경우에는 인터럽트를 이용하여 master에 데이터전송을 요청하게 된다. PCI 컨트롤러를 이용하여 FLOVA 내부에 접근 할 수 있는 메모리 영역은 명령어 캐쉬를 제외한 내부메모리 영역 전역과 외부 메모리 영역 전역을 접근 할 수 있다. 이러한 메모리 영역은 직접 접근을 하지 않고 FLOVA 내부의 버스 arbitration 유닛에 의해 중재를 받게 된다.

그림-2는 PCI 버스와 FLOVA와의 연결된 핀을 보여 주고 있다.

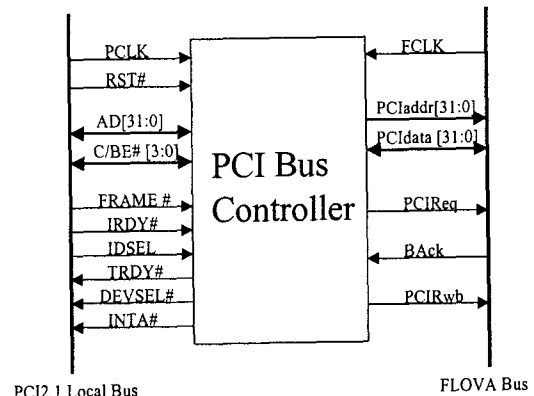


그림-2 PCI 컨트롤러의 핀 연결

#### 컨트롤러

PCI의 컨트롤러는 기본적으로 내부 블록을 제어하는 기능을 수행하게 된다. 발생되는 대부분의 제어는 PCI 버스 클럭(33MHz)에 움직이는 블록과 FLOVA의 내부 클럭(100MHz)에 의해 움직이는 블록을 제어하는 2개의 컨트롤러 블록으로 구성이 되어 있다. 일반적으로 한 개의 클럭을 사용하는 디자인과는 달리 PCI 컨트롤러는 두 개의 클럭을 사용함으로써 서로 다른 주파수에 의해 동작되는 블록간의 프로토콜 문제와 hold/setup time에 주의를 하여야 한다. 뿐만 아니라

PCI 버스 클락은 33MHz로 고정되어 있는 반면 FLOVA의 버스 클락은 100MHz 이내의 모든 주파수로 동작이 가능하게 된다. 이러한 경우 33MHz 보다 더 빠른 주파수로 동작하는 경우와 이보다 느린 주파수로 동작하는 경우에 대해 서로 두 블록간에 병목이 되는 부분이 달라지므로 컨트롤러의 역할이 달라지게 된다. 예를 들어 FLOVA의 클락이 33MHz 이상으로 동작을 하게 되면 컨트롤러는 PCI 버스 쪽에서 오는 데이터가 병목이 되어 PCI 버스의 준비신호인 TRDY# 신호를 없애게 된다. 반면 FLOVA가 PCI 버스 클락보다 느릴 경우 FLOVA 신호 중 PCIREq를 없애게 된다. 이처럼 컨트롤러는 데이터가 이동하는데 대해 병목을 없앨 수 있도록 조절을 하게 되며 주파수가 서로 다른 두 블록사이의 컨트롤러간에는 Req나 Ack와 같은 기본적인 신호를 이용하여 handshaking을 하고 있다.

### 버퍼

서로 다른 주파수간에 데이터를 전송하는데 있어 느린 주파수로 동작되는 블록의 병목을 없애기 위해 버퍼링을 하여야 한다. 이러한 버퍼는 두 개로 구성이 되며 read시와 write시 서로 다른 경로의 버퍼를 지나게 된다[7]. 서로 다른 경로의 디자인이 된 이유는 write와 read의 경로가 같을 경우 두 개의 버퍼 클락이 사용되는데 이는 매우 위험한 디자인이 된다. 이를 방지하기 위해 버퍼에 write 클락과 read 클락을 다르게 하였다. 이러한 데이터의 경로는 write시에는 PCI 블록에서 PCI 버스 프로토콜에 맞추어 데이터를 버퍼에 저장한 다음 컨트롤 신호에 의해 FLOVA 블록에서 FLOVA 클락에 맞추어 데이터를 읽어 갈 수 있도록 하였다. read는 이와 반대로 동작을 하게 된다.

버퍼의 사이즈는 read 버퍼와 write 버퍼 모두 32비트 16개로 구성이 되어 있다. 16개의 버퍼 크기는 burst 데이터 전송의 효율을 높이면서 FLOVA의 전체 면적에 영향을 주지 않는 크기로 결정하였다. 표-1은 버퍼 사이즈에 따른 1Kbytes를 전송하는데 걸리는 시간을 비교한 것이다.

Buffer Size	Burst Write	Burst Read
4 depth	42.3usec	49.0usec
8 depth	24.6usec	27.4usec
16 depth	16.5usec	18.7usec

표-1 1Kbytes를 전송하는데 걸리는 시간

### 어드레스 생성부

초기화 단계에서 할당받은 어드레스를 FLOVA의 어드레스로 변환하기 위한 블록이다. PCI 버스는 한 개의 버스로 어드레스와 데이터를 보내는 time-multiplexed 어드레스/데이터 버스를 사용하고 있다. 이러한 방식은 burst mode의 경우 한 개의 어드레스와 여러 개의 데이터가 전송이 되는데 각각의 데이터에 대하여 어드레스를 카운트해야 한다. 마찬가지로 burst read의 경우 연속적으로 읽혀질 데이터에 대해 어드레스를 생성하여 read 버퍼에 먼저 읽어 들이므로 read시 발생하는 penalty를 줄였다.

### Configuration 레지스터

PCI 버스의 특징 중 하나가 어드레스를 동적으로 할당받을 수 있다는 것이다. 이러한 과정은 파워가 켜지는 순간 PNP에 의해서 이루어진다. configuration 레지스터는 디바이스가 정상 동작을 할 수 있도록 기본적인 정보를 가지고 있다.

## IV. PCI의 기능 검증

PCI 컨트롤러의 기능 검증은 FLOVA내부에 탑재되기 이전에 반드시 있어야 한다. 다른 기능블록과 다르게 PCI의 검증은 실제 PC의 신호들을 기반으로 하고 있기 때문에 시뮬레이션으로만 검증이 어렵다. 또한 PCI 컨트롤러는 PC에 연결되어 있는 PCI 버스 프로토콜과 FLOVA 버스의 프로토콜을 연결하고 있기 때문에 단지 시뮬레이션과 같은 가상 환경에서는 실제 발생하는 문제에 대해 충분히 고려하기가 어렵다. 뿐만 아니라 앞에서 언급했듯이 PCI 컨트롤러는 FLOVA 내부에 탑재되는 블록이기 때문에 반드시 칩이 만들어지기 이전에 검증이 되어야 하는 어려움이 있다.

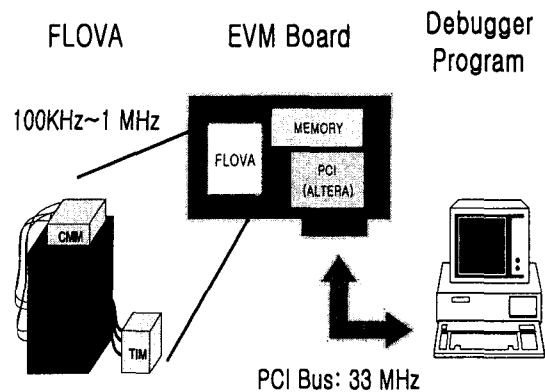


그림-3 PCI board의 검증모델

이러한 어려움을 해결하고자 그림-3에서 보는 것과 같이 에뮬레이터를 이용하여 실제 칩이 있는 것처럼 에뮬레이션하고 내부에 탑재되어야 할 PCI 컨트롤러는 FPGA를 이용하여 구현을 하였다. 내부에 탑재되는 PCI 컨트롤러를 외부로 나오게 되면서 발생하는 커넥션을 그대로 유지하게 되지만 실제 칩에서 동작되는 것과는 동작 주파수가 달라지게 된다. 이 때 FLOVA가 동작하는 실제 주파수는 100MHz이지만 에뮬레이션을 할 때는 이보다 느린 100KHz~1MHz로 동작을 하게 된다.

에뮬레이션을 위해 사용된 장비는 3백만 게이트를 에뮬레이션 할 수 있는 QUICKTURN사의 M3000을 이용하였으며[10], 이는 초당 40만개의 FLOVA 명령어를 수행할 수 있다. FPGA로 구현이 된 PCI 컨트롤러는 ALTERA사의 FLEX10K50을 이용하여 구현하였다. 그 외 에뮬레이션을 수행할 수 있도록 명령어를 다운로드 받는 외부 메모리를 가지고 있다.

에뮬레이션을 이용하여 실제 환경에서 검증은 하는 경우 시뮬레이션에 비해 많은 종류의 경우에 대해 수행 가능하므로 고장 검출율을 높일 수 있다.

## V. 결론

멀티미디어 프로세서의 경우 호스트 프로세서와 커뮤니케이션을 할 수 있는 PCI 버스 컨트롤러와 같이 빠른 버스 컨트롤러가 없다면 멀티미디어 프로세서가 가지고 있는 기능을 발휘하기가 어렵다. 그러므로 앞으로 PCI 버스 컨트롤러와 같은 버스 컨트롤러의 제작은 멀티미디어 프로세서 디자인의 경우 필연적이게 된다. 이 때 서로 다른 프로토콜의 두 버스를 쉽게 연결할 수 있고 동작 주파수가 다르므로 생기는 병목을 없앨 수 있도록 디자인해야 한다. 이러한 방법으로 제안한 것이 write시와 read시 경로를 다르게 함으로 상호 주파수 차이로 발생하는 문제를 해결할 수 있다. 또한 병목을 없애기 위해 사용되는 버퍼의 크기는 멀티미디어 프로세서가 사용되는 목적에 따라 달라질 수 있지만 일반적으로 버퍼 사이즈가 증가함에 따라 burst mode 전송의 경우 시간이 더 짧게 걸림을 알았다. 그러나 버퍼 사이즈가 증가함에 따라 PCI 컨트롤러가 차지하는 면적은 선형적으로 증가하게 되는데 이는 또 다른 디자인의 문제를 발생시킬 수 있으므로 한번에 전송해야 하는 데이터의 양과 PCI 버스 컨트롤러가 차지할 수 있는 면적을 충분히 고려하여 디자인하여야 한다. 그리고 무엇보다 중요한 것은 반드시 PCI 버스 컨트롤러는 실제 환경에서 검증을 하여야 한다는 것이다. 이는 많은 어려움을 가지고 있는데 무엇보다도 실

제 칩 내부에서 동작하는 것과 같은 모델로 에뮬레이션을 해야한다.

## 참고문헌

- [1] Steger, H, "Graphical object-oriented multimedia application development: Technology and market trends," COMPCON 'Technologies for the Information Superhighway' Digest of Papers, pp.299, 1996.
- [2] T. Shanley and D. Anderson, "PCI System Architecture," Richardson, TX : MindShare Press, 1993.
- [3] D.B.Jung, Y.W.Hwang and J.K.Choi, "A Structure of Multimedia Head-End Platform using PCI Environments," International Conference on Signal Processing Applications and Technology, 1997.
- [4] S.J.Nam, Y.S.Kwon, Y.H.Im, K.K.Kang, and C.M.Kyung, "DIVA: Dual-Issue VLIW Architecture with Media Instructions for Image Processing," IEEE Transaction on Consumer Electronics, Vol.45, No.1, pp.192-202, Feb. 1999.
- [5] S.J.Nam et.al, "VLIW Geometry Processor for 3D Graphics Acceleration," International Symposium on Low-Power and High-Speed Chips (COOL Chips), pp.107-120, Apr. 1999.
- [6] Chia-Lin Yang; Sano, B.; Lebeck, A.R., "Exploiting Instruction Level Parallelism in Geometry Processing for Three Dimensional Graphics Applications," Proc. 31st Annual ACM/IEEE International Symposium on Microarchitecture, pp.14-24, 1998.
- [7] Philips Electronics North America Corporation, TriMedia TM1000 Preliminary Data Book, 1997
- [8] <http://www.techfest.com/hardware/bus/pci.htm>
- [9] PCI Special Interest Group, PCI Local Bus Specification, Rev. 2.0, 1993.
- [10] Quickturn Design Systems, Inc., System Realizer Users Guide Version 5.0.