

## Xilinx FPGA용 PCI express 구현 및 성능 분석

이진\*

### Implementation and Performance Evaluation of PCI express on Xilinx FPGA

Jin Lee\*

Department of Information and Communication, Pyeongtaek University, Gyeonggi-do 17869 Korea

#### 요약

하드웨어 가속기를 사용하여 다양한 실시간 계산을 하는 여러 공학/과학 분야에서 많은 경우에 FPGA와 호스트 컴퓨터를 PCI express(PCIe)로 연결하는 시스템 구성이 요구된다. 하지만, 초당 수 기가바이트의 데이터를 주고 받는 고속 인터페이스인 PCIe의 구현은 하드웨어 가속기 개발의 가장 큰 어려움 중에 하나이다. 상용 제품과 논문을 통해서 여러 PCIe IP 솔루션을 찾을 수 있지만, 고가의 비용을 지불해서 구매하거나, 별도의 시간과 노력을 투자해서 PCIe를 구현해야 한다. 따라서, Xilinx사의 FPGA를 기반의 하드웨어 가속기를 구현할 때는 Xilinx사에서 무료로 제공하는 XDMA PCIe IP를 사용하는 것이 개발 기간 및 비용 단축을 위한 최선의 선택이 될 수 있다. 이러한 이유로 본 논문에서는 Xilinx사의 PCIe IP의 성능 평가를 위해 Zynq-7000 FPGA 개발보드와 Windows 10 호스트 컴퓨터로 평가 시스템을 구성하고, PCIe IP의 구성 파라미터에 의한 전송 속도 성능 변화에 대해 평가 분석한다.

#### ABSTRACT

Recently, speeding up real time calculation using the specialized hardware accelerator is often used in the various engineering and science area, and the accelerators are required to include PCI express interconnection between FPGA and a host computer. The implementation of the high speed PCIe for the multi-giga bytes per second transmission is one of the most difficult issue in the development of the accelerators. There are several commercialized IP solutions and research results in the literature, but these solutions are required extra cost and design period to analyze the detailed implementation method. For the hardware accelerator on Xilinx FPGA, utilizing Xilinx's XDMA PCIe IP, which is provided without extra charge, can be the best solution in terms of the development period and cost. Consequently, this paper presents the evaluation system on Zynq-7000 FPGA and Windows 10 host computer, and analyze the performance of the PCIe IP with various configuration parameters.

**키워드** : PCI express, FPGA, DMA, 성능

**Key word** : PCI express, FPGA, DMA, Performance

Received 12 November 2018, Revised 15 November 2018, Accepted 20 November 2018

\* Corresponding Author Jin Lee (E-mail:mygenie79@ptu.ac.kr, Tel:+82-31-659-8288)

Department of Information and Communication, Pyeongtaek University, Gyeonggi-do 17869 Korea

Open Access <http://doi.org/10.6109/jkiice.2018.22.12.1667>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서론

정보통신 분야를 비롯한 수많은 과학/공학 실험에서 실시간 데이터 수집, 프로세싱, 관측을 위해 초당 수 기가바이트의 데이터 처리가 필요해지면서 특수한 하드웨어를 사용하는 경우가 많아졌다. 대부분의 경우 복잡하게 프로세서를 연결하는 것보다 FPGA (Field Programmable Gate Array)를 사용해서 하드웨어 가속기를 설계한다[1-9].

다양한 경우에 대해 실험하고 분석하기 위해서 FPGA 기반 하드웨어 가속기는 호스트 컴퓨터와 연동되어 동작해야 하고, 이 경우 호스트 PC와 FPGA 보드 간의 인터페이스에서 병목현상(bottleneck)이 발생한다.

특히 PC의 확장형 카드로 사용되는 PCI express (PCIe) 기술은 고속 (high-bandwidth)의 확장 가능하고 (Scalable), 신뢰성 있는 (reliable) 기기간 연결 프로토콜이기 때문에, 자연스럽게 PCIe가 많은 응용분야에서 FPGA 하드웨어 가속기와 호스트 PC간 고속 연결에 사용되고 있다.

PCIe를 사용한 데이터 전송은 메모리 기반으로 이루어지기 때문에 CPU가 일반 메모리를 접근할 때 사용하는 DMA (Direct Memory Access)를 사용해서 구현한다. DMA를 포함하는 PCIe 링크를 FPGA에 사용할 수 있는 솔루션이 여러 개 존재한다. 상용 제품으로 Hitechglobal, PLDA, CAST, Northwest, Xillybus, MPRACE와 같은 제품이 존재하고, 비상용으로 PCIe의 물리 계층은 FPGA의 고속 인터페이스 IP를 사용하면서 상위 계층과 DMA를 연동해서 구현한 EPEE, ffLink, DyRACT와 같은 연구 결과물도 존재한다[2-4]. 하지만, 상용 제품은 추가 비용이 필요하고 논문으로 발표된 연구 결과물을 기반으로 실제 PCIe IP를 개발하는 것은 많은 시간과 투자가 필요하다.

따라서, 많은 개발자들이 Xilinx나 Altera와 같은 FPGA 업체에서 제공하는 PCIe IP와 드라이버를 활용하여 하드웨어 가속기를 설계하게 된다.

Xilinx사는 Virtex5에서부터 고속 SERDES 인터페이스를 탑재하여, PCIe를 지원 하는 IP를 제공하기 시작하였다. 10년 이상의 버전 업이 있었고, 그동안 호스트 PC의 OS 및 CPU 등의 하드웨어 장치도 큰 성능 향상이 있었으므로 최신의 Xilinx IP의 정확한 성능 평가가 필요하다. 본 논문에서는 Xilinx사에서 제공하는 DMA가 탑재된 PCIe IP 최신 버전 [10]을 Windows 10 64비트 OS가 탑재된 컴퓨터에 실장하여 동작을 검증하고 성능을 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 PCIe의 기본 개념 및 이론적 최대 성능에 대해서 알아본다. 3장에서 PCIe를 FPGA에서 구현한 관련 연구들 및 Xilinx사의 PCIe IP에 대해서 설명한다. 4장에서 Zynq-7000 FPGA가 탑재된 ZC706 개발 보드[11]에 PCIe를 구현하여 성능 분석한 결과를 설명하고 5장의 결론으로 끝을 맺는다.

본 논문의 구성은 다음과 같다. 2장에서는 PCIe의 기본 개념 및 이론적 최대 성능에 대해서 알아본다. 3장에서 PCIe를 FPGA에서 구현한 관련 연구들 및 Xilinx사의 PCIe IP에 대해서 설명한다. 4장에서 Zynq-7000 FPGA가 탑재된 ZC706 개발 보드[11]에 PCIe를 구현하여 성능 분석한 결과를 설명하고 5장의 결론으로 끝을 맺는다.

## II. PCI Express

### 2.1. PCI Express 기본 개념

PCIe는 x1에서 x32까지 확장 가능한 다수의 레인 (lane)으로 구성된 물리 계층을 기반으로 고속 데이터 전송을 제공하는 점대점(point-to-point) 기기간 연결 프로토콜이며, 그림 1과 같이 트랜잭션 계층 (TL : Transaction Layer)와 데이터 링크 계층 (DL : Data Link), 물리 계층 (PL : Physical Layer)의 3개의 계층 (Layer)으로 구성되어 있다 [12].

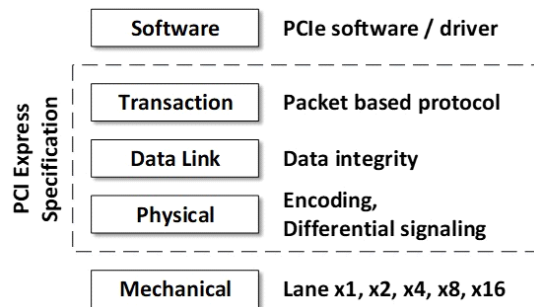


Fig. 1 The layers of PCI Express architecture.

PCIe는 Generation(Gen) 1부터 Gen 4까지 발표 되었고, Gen 5가 개발되는 중이지만, 아직까지 상용제품은 Gen3와 Gen2가 주를 이룬다.

물리계층은 전송 속도와 레인의 수 인코딩 방식으로 정리할 수 있다. 전송 속도는 싱글 레인 (x1)일 때의 인코딩된 데이터를 전송 단위로 하여 Gbps단위 대신 GT/s(Giga Transfer per Second)를 사용한다. PCIe는 세

대 별로 2.5GT/s, 5GT/S, 8GT/s 그리고 16GT/s의 속도를 갖는다. 인코딩 방식은 Gen 2까지는 8b/10b을 사용하고 Gen 3부터는 128b/130b 방식을 사용한다.

데이터 링크 계층에서는 신뢰성 있는 통신을 위해서 CRC 체크섬을 추가하여 전송의 성공여부를 확인하고, 링크의 플로우 제어 (flow control)을 담당한다.

트랜잭션 계층에서는 실제 사용자 데이터를 포함하는 트랜잭션 계층 패킷 (TLP, Transaction Layer Packet) 을 사용하여 기기 사이에 패킷 통신이 이루어지게 한다.

송신 기기에서는 그림 2와 같이 사용자 데이터를 페이로드(payload)로 하는 TLP를 구성한다. TLP에는 트랜잭션 타입, 수신자 주소, 전송 사이즈에 관련된 정보가 포함된다. 이후 데이터 링크에서 CRC 체크섬과 시퀀스 ID를 추가하고 물리 계층에서 멀티 레인에 맞는 순서 정렬 및 인코딩을 하고 고속 직렬 차동 모드 방식으로 전송을 하게 된다. 수신 기기에서는 역 순으로 데이터를 복원하는 과정을 거친다.

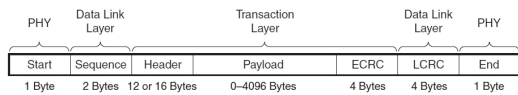


Fig. 2 Transaction layer packet (TLP).

PCIe를 사용하는 시스템의 토폴로지 (topology)는 그림 3과 같다. 주 장치에 해당하는 루트 콤플렉스 (Root Complex, RC) 기기와 종 장치에 해당하는 엔드포인트 (Endpoint, EP) 그리고 스위치 (Switch)로 구성된다. FPGA 기반의 하드웨어 가속기들은 대부분 스위치와 연결된 EP로 구현된다.

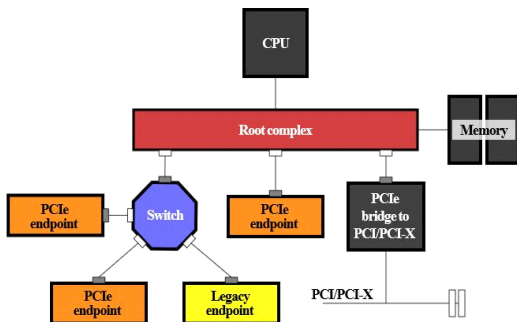


Fig. 3 Topology of PCI Express system[12].

### 2.2. PCI Express의 이론적 성능

PCIe의 이론적 최대 대역폭(Bandwidth)는 전송속도와 레인의 수 그리고 인코딩 방식에 따라 식 (1)과 같이 표현될 수 있다.

$$B = D \cdot N \cdot E \tag{1}$$

$B$ 는 최대 대역폭을 나타내고, 단위는 bps이다.  $D$ 는 전송속도로 Gen1은 2.5GT/s, Gen2는 5GT/s, Gen3는 8GT/s 그리고 Gen4는 16GT/s이다.  $N$ 은 레인의 수를 나타낸다.  $E$ 는 물리 계층 인코딩의 전송 효율을 나타낸다. Gen1과 Gen2는 8/10이고, Gen3 이상은 128/130이다. 식 (1)에서 알 수 있듯이 PCIe의 물리 계층의 인코딩 방식에 따라 Gen2까지는 전송속도의 20%까지 손실이 생긴다. Gen3이상은 128b/130b 인코딩을 사용하여 이에 대한 손실을 대폭 줄일 수 있다.

그림 2의 TLP 구조에서 알 수 있듯이 데이터 링크와 트랜잭션 계층에서는 시퀀스 ID, CRC 체크섬과 헤더와 같은 오버헤드가 추가 된다. 이 오버헤드는 20에서 28바이트까지 가변될 수 있다. 따라서 PCIe의 이론적 최대 전송속도(throughput)은 식 (2)와 같이 정리할 수 있다.

$$T = \frac{P_L}{P_L + O_V} \cdot B \tag{2}$$

$P_L$ 은 최대 페이로드 사이즈 (MPS, Maximum Payload Size)이고,  $O_V$ 는 패킷의 오버헤드를 나타낸다. MPS에 따라 전송속도가 변할 수 있다. 예를 들어 MPS가 256바이트이고 오버헤드가 20바이트인 PCIe Gen2의 x4 레인 연결의 이론적 최대 전송속도는 1840MBytes/s 이다. 하지만 MPS가 128바이트라면 최대 throughput은 1720MBytes/s로 떨어지게 된다. MPS는 그림 2에서 알 수 있듯이 4,096바이트가 최대 값이다.

PCIe 프로토콜은 양쪽 기기간의 신뢰성 있는 통신을 위해서 전송 확인 또는 플로우 제어와 같이 트래픽 (traffic) 특징에 따라 변하는 프로토콜 오버헤드가 추가 되기 때문에, 단순한 패킷의 오버헤드만 고려한 식 (2)의 전송속도와 실제 성능은 다소 차이가 있다 [13].

## III. FPGA용 PCI Express

PCIe를 사용한 데이터 전송은 메모리 기반으로 이루어지기 때문에 CPU가 일반 메모리를 접근할 때 사용하

는 DMA (Direct Memory Access)를 사용해서 구현한다. FPGA 업체 및 3rd 파티에서 상용으로 판매하는 IP도 있고, 여러 논문에서 제안된 방법도 있지만 기본적인 구조는 유사하다.

FPGA 내부의 Gigabit SERDES 인터페이스 로직을 이용한 물리 계층을 구성한다. 디지털 로직으로 PCIe의 데이터링크와 트랜잭션 계층을 구현하는데, CPU의 개입 없이 TLP 페이로드의 처리를 하기 위해 DMA가 사용된다 [1-7].

PCIe를 사용하기 위해서는 FPGA에 구현되는 endpoint 하드웨어 뿐만 아니라 호스트 PC에서 사용되는 PCIe 드라이버도 같이 개발되어야 한다 [4, 7].

### 3.1. 관련 연구

상용 제품으로 판매되는 PCIe 솔루션으로는 Hitechglobal, PLDA, CAST, Northwest와 Xillybus가 있다. IBM사에서는 CAPI라는 PCIe로 호스트 컴퓨터와 연결되는 FPGA 기반 하드웨어 가속기 기술을 제공한다. Academic 사이트에서는 ffLink, RIFFA, JetStream와 같은 기술이 Gen3를 지원하도록 구현되었고, DyRACT, EPEE, FlexWAFE와 Speedy와 같은 솔루션이 Gen2용으로 개발되었다 [2-4].

최근에 논문에 발표된 기술로는 JetStream[3]과 L. Rota[1, 5] 등의 결과물이 있다. 표 1에서는 위에서 소개된 PCIe 솔루션의 특징 및 성능을 비교하였다. 최고 전송속도는 쓰기과 읽기 속도가 서로 다르지만, 비슷한 수치이므로 쓰기 속도의 대략적인 값을 표시하였다. 표 1은 [3]에서 정리된 내용을 기반으로 수정, 추가 되었음을 밝힌다.

### 3.2. Xilinx의 PCI Express IP

Xilinx에서는 Virtex 5 시리즈에서부터 PCI express IP를 제공하기 시작하였고 6과 7시리즈 제품군이 나오면서 본격적으로 Gen1부터 Gen3까지 지원할 수 있는 새로운 버전의 IP들이 제공되고 있다 [13].

Table. 1 FPGA PCIe framework comparison.

Solution	Host	PCIe	Peak throughput (MB/s)
EPEE	Linux	Gen 2 (x8)	3,200
ffLink	Linux	Gen 3 (x8)	7,000

Solution	Host	PCIe	Peak throughput (MB/s)
DyRACT	Linux	Gen 2 (x4)	1,450
RIFFA 2.2	Linux Windows	Gen3 (x4)	3,040
Xillybus	Linux Windows	Gen3 (x4)	3,500
Northwest	Linux Windows	Gen3 (x8)	5,900
JetStream[3]	Linux	Gen3 (x8)	7,000
L. Rota[5]	Linux	Gen2 (x16)	6,920

추가 비용 없이 고성능 저전력 PCIe IP를 사용할 수 있으며, Linux와 Windows용 드라이버도 제공된다. 3.1절에서 알아본 다른 관련 기술들과 마찬가지로 Xilinx의 IP도 DMA 기반의 구조로 구현되어 있다. 표 2에서는 7시리즈 이상의 Xilinx FPGA에서 사용할 수 있는 PCIe 사양을 정리하였다.

Table. 2 PCIe solution portfolio in Xilinx FPGA.

Family	PCIe	Data Rate (GB/s)	Number of Blocks
Virtex Ultrascale+	Gen3 (x16) Gen4 (x8)	16	2 - 6
Zynq Ultrascale+	Gen3 (x16) Gen4 (x8)	16	0 - 5
Virtex Ultrascale	Gen3 (x8)	8	2 - 6
Zynq-7000	Gen2 (x8)	4	1

최신 디바이스인 Ultrascale+에서는 Gen4까지 지원 가능하다. Zynq는 ARM CPU가 내장되어 있는 디바이스이고, 본 논문에서는 Zynq-7000 FPGA를 사용해서 PCIe를 구현하고 성능 분석을 진행하였다.

Xilinx사에서는 7 시리즈 이상의 Xilinx FPGA에서 사용할 수 있는 DMA/Bridge Subsystem for PCI Express라는 IP를 XDMA라는 이름으로 제공하고 있다 [10]. XDMA는 쓰기과 읽기 기능을 H2C (Host-to-Card)와 C2H (Card-to-Host) 채널을 제공하여 지원한다. 또한, Scatter Gather DMA를 포함하며 AMBA의 AXI4 인터페이스를 사용하여 XDMA가 AXI의 마스터가 되어 메모리 및 기타 하드웨어와 연결할 수 있다.

XDMA는 Xilinx의 Vivado 개발 환경에서 쉽게 구현할 수 있으며, 사용자가 가변 할 수 있는 옵션을 다양하

계 제공하고 있다.

기본 설정 (Basic 옵션)에서는 DMA의 사용 여부, PCIe 레인의 수, 최대 링크 속도 및 AXI 인터페이스의 버스 폭을 조정할 수 있다. 그림 4는 XDMA의 기본 설정 창을 보여준다[10].

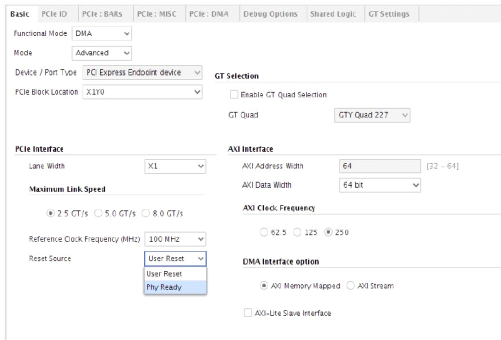


Fig. 4 Basic options for Xilinx XDMA IP.

XDMA는 PCIe endpoint(EP) 기능만을 제공하며, 레인의 수와 링크 전송속도 외에도 AXI의 버스폭과 클럭 속도도 성능에 영향을 미친다.

기본 설정 외에 PCIe의 주소와 인터럽트 체계에 대한 부가 옵션 설정도 제공한다. 본 논문에서는 Xilinx XDMA IP의 쓰기와 읽기 성능에 대한 분석만 다루기 때문에, 다른 옵션들에 대한 설명은 다루지 않는다.

#### IV. 구현 및 성능 분석

##### 4.1. PCIe Endpoint 하드웨어 구현

본 논문에서는 Zynq-7000 FPGA가 탑재된 ZC706 보드를 Windows10 64비트 OS 컴퓨터에 장착해서 성능 분석을 진행하였다. 그림 5에서 알 수 있듯이 ZC706 보드는 4개의 레인까지 지원할 수 있다.

XDMA의 성능 분석을 위해 그림 6과 같이 시스템을 구성하였다. XDMA가 AXI 마스터로 동작을 하고, AXI SmartConnect를 통해서 Xilinx BlockRAM과 테스트용 GPIO 모듈이 연결되어 있다. 메모리인 BlockRAM의 크기에 따라 호스트에서 한 번에 처리할 수 있는 데이터 버퍼의 크기가 결정되며, 이는 성능에 많은 영향을 미친다. 이에 대한 분석은 4.2절에 나와 있다.

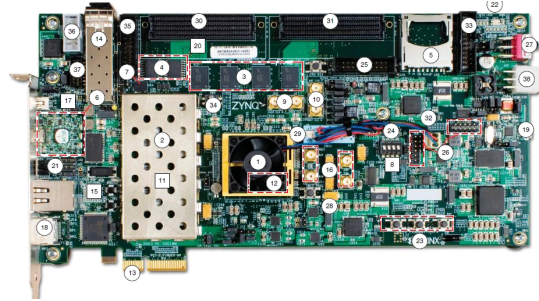


Fig. 5 Xilinx ZC706 EVM board [11].

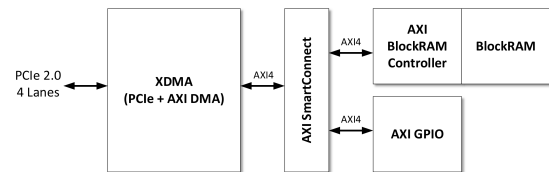


Fig. 6 Block diagram of XDMA evaluation system

PCIe를 통해서 RC와 EP사이에 100MHz 기준 클럭이 차동(differential) 신호로 연결된다. XDMA IP는 단종단 (single-ended) 클럭을 기준 클럭으로 입력 받기 때문에, 다음과 같이 상위 모듈에서 IBUFDS\_GTE2라는 클럭 버퍼를 삽입해서 연결해 주어야 한다.

```
// PCIe Reference Clock Input buffer
IBUFDS_GTE2 pcie_clk_ibuf (
    .I (pcie_ref_clk_p ),
    .IB (pcie_ref_clk_n ),
    .O (w_clk_100MHz ),
    .ODIV2 ( )
);
```

##### 4.2. Xilinx XDMA IP 성능 분석

4.1절과 같이 구현된 하드웨어와 Xilinx에서 제공하는 Windows 10 드라이버를 활용하여 성능 분석을 하였다.

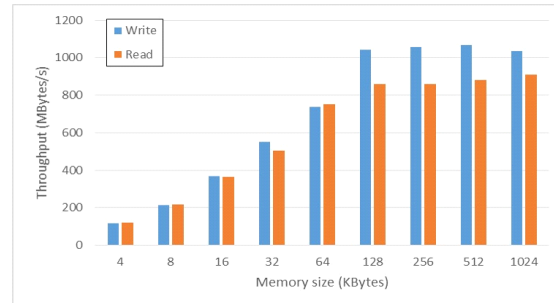


Fig. 7 PCIe throughput vs. internal memory size.



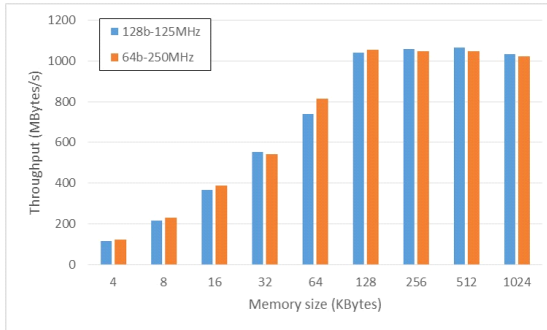


Fig. 8 Write throughputs with different AXI options

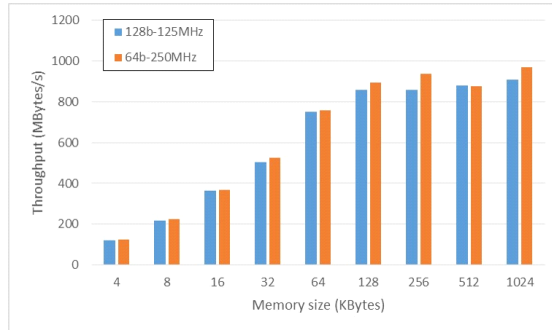


Fig. 9 Read throughputs with different AXI options

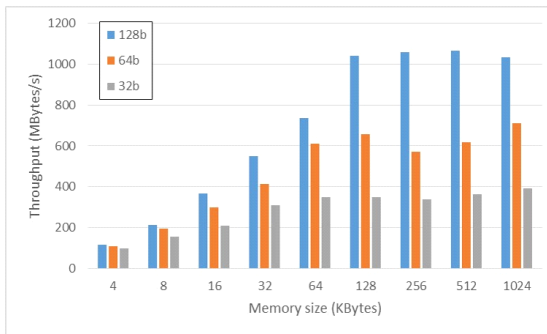


Fig. 10 Write throughputs with different memory widths

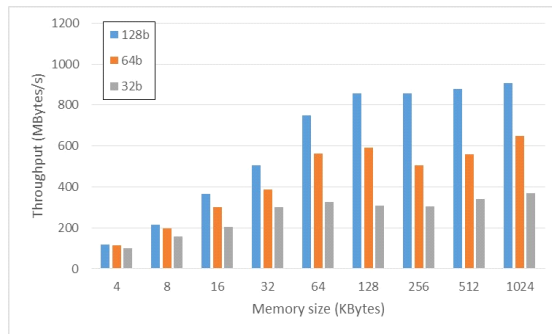


Fig. 11 Read throughputs with different memory widths

PCIe는 Gen2로 4개의 레인을 사용하도록 구성되었으며, 엔드포인트 내부의 메모리 크기와 메모리 인터페이스 사양에 따른 성능 변화를 분석한다.

XDMA의 AXI 인터페이스 버스폭은 128 비트 버스폭은 125MHz로 설정하였다.

그림 6에서의 BlockRAM의 크기 만큼 호스트에서 한번에 쓰기 또는 읽기를 처리한다. 드라이버에서 이를 MPS에 따라 분해하거나 재조립하게 된다. 그림 7은 BlockRAM 크기에 따른 최고 전송속도의 변화를 보여 준다. 전송 속도는 1GBytes의 데이터를 전송하는 시간을 10회 측정하여 평균하여 계산되었다.

그림 7에서 알 수 있듯이, 실험 결과 EP에서 128KByte 이상의 내부 메모리를 포함해야 호스트 드라이버와 연동되었을 때 최대의 성능을 낼 수가 있다. 즉, 128Kbyte보다 메모리가 큰 경우에는 성능 변화가 거의 없다. 쓰기의 경우 최대 1~1.05 GByte/s의 성능을 보이고, 읽기의 경우 최대 850~910MByte/s의 성능을 보인다. 하지만, MPS의 최대 값인 4KByte로 내부 메모리를 설정하면 쓰기와 읽기 전부 약 115MByte/s 전송 속도로 최대성능의 10~12% 밖에 사용할 수가 없다.

3.2절에서 설명했듯이, XDMA의 기본 설정에서 AXI 인터페이스의 버스폭과 동작 속도를 변경할 수가 있다. 두 가지 조건으로 변경 가능한데, 첫 번째는 128비트 버스폭과 125MHz를 사용하는 것이고, 다른 하나는 64비트 버스폭과 250MHz를 사용하는 것이다. 버스폭이 작을수록 하드웨어의 복잡도는 감소하는 대신에 같은 성능을 위해서 클럭속도가 높아져야 한다.

FPGA 내부의 시스템 구성에 따라 사용자가 선택을 해야 하며, 본 논문에서는 이 두 가지 경우에 성능 차이가 있는지 분석하여 그림 8과 9에 나타내었다. 내부 메모리의 크기가 충분할 때, 쓰기 동작의 성능은 큰 차이가 없다. 하지만, 읽기 동작의 성능은 64비트-250MHz 설정일 때 다소 향상됨을 볼 수가 있는데, FPGA 종류에 따라 250MHz의 동작속도를 갖도록 설계하는 것은 다소 무리가 있으므로 신중하게 선택해야 한다.

XDMA의 AXI 인터페이스 뿐만 아니라 내부 메모리인 BlockRAM의 파라미터도 성능에 큰 영향을 미친다. AXI에서는 서로 다른 버스폭을 갖는 모듈끼리도 연결이 가능하기 때문에, BlockRAM의 데이터버스 폭에 따라 성능이 어떻게 변하는지 분석할 필요가 있다.

기본적으로 AXI 버스와 메모리 데이터 버스는 정수 배의 관계가 있어야 하므로, 128비트-125MHz AXI 인터페이스일 때 128비트/64비트/32비트 메모리 데이터 버스를 사용하는 경우의 성능을 분석하였다. 동작 클럭이 동일한 경우 데이터 버스 폭에 비례하게 성능변화가 있다고 예상할 수 있지만, 그림 10과 11에서 알 수 있듯이 버스폭의 비율 차이보다는 성능 저하가 적게 일어난다. 내부 메모리 크기가 128KB 이상일 때의 성능 변화는 표 3과 같다. 버스 폭이 128비트일 때와 비교하여 64비트일 때 약 30~45%, 32비트일 때 약 60~70% 정도 성능 저하가 발생한다.

**Table. 3** Performance with different memory widths

Direction \ Width	128bit	64bit	32bit
Write	1040 MB/s	657 MB/s	350 MB/s
Read	857 MB/s	592 MB/s	309 MB/s

## V. 결 론

FPGA를 기반으로 하는 하드웨어 가속기를 활용해 다양한 실험을 하기 위해서는 호스트 컴퓨터와 PCIe로 연결되어야 한다. 기존의 다양한 FPGA용 PCIe 솔루션이 존재하지만, 개발자들이 실제로 적용하는 데는 많은 어려움이 있다. 본 논문에서는 Xilinx사에서 무상으로 Windows와 Linux 드라이버와 함께 제공하는 PCIe IP인 XDMA IP를 실제 구현하고 성능을 분석하였다. Gen2 x4 사양인 경우, 최대 전송속도는 쓰기일 때 1,050MByte/s, 읽기 일 때 970MByte/s로 측정되었다. 또한 본 논문에서는 내부 메모리의 크기, 시스템 AXI 버스 파라미터, PCIe와 내부 메모리 데이터 버스폭의 차이와 같이 시스템 구성할 때 변경 가능한 파라미터들이 PCIe 성능에 미치는 영향을 분석하였다.

## References

- [ 1 ] L. Rota, M. Vogelgesang, L. E. Ardila Perez, M. Caselle, S. Chilingaryan, T. Dritschler, N. Zilio, A. Kopmann, M. Balzer, and M. Weber, "A High-throughput Readout Architecture based on PCI-Express Gen3 and DirectGMA Technology," *Journal of Instrumentation*, vol. 11, pp. 1-9, Feb. 2016.
- [ 2 ] H. Kaviani-pour, S. Muschter, and C. Bohm, "High Performance FPGA-Based DMA Interface for PCIe," *IEEE Transactions on Nuclear Science*, vol. 61, no. 2, pp. 745-749, Apr. 2014.
- [ 3 ] M. Vesper, D. Koch, K. Vipin, and S. A. Fahmy, "JetStream: An Open-Source High-Performance PCI Express 3 Streaming Library for FPGA-to-Host and FPGA-to-FPGA Communication," in *Proceedings of the 26th International Conference on Field Programmable Logic and Applications*, Lausanne, Switzerland, pp. 1-9, Aug. 2016.
- [ 4 ] J. Gong, T. Wang, J. Chen, H. Wu, F. Ye, S. Lu, and J. Cong, "An Efficient and Flexible Host-FPGA PCIe Communication Library," in *Proceedings of the 24th International Conference on Field Programmable Logic and Applications*, Munich, Germany, pp. 1-6, Sep. 2014.
- [ 5 ] L. Rota, M. Caselle, S. Chilingaryan, A. Kopmann, and M. Weber, "A PCIe DMA Architecture for Multi-Gigabyte Per Second Data Transmission," *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 972-976, Jun. 2015.
- [ 6 ] N. Zilberman, Y. Audzevich, G. A. Covington, and A. W. Moore, "NetFPGA SUME: Toward 100Gbps as Research Commodity," *IEEE Micro*, vol. 34, issue. 5, pp. 32-41, Jul. 2014.
- [ 7 ] A. Byszuk, J. Kolodziejski, G. Kasproicz, K. Pozniak, W. M. Zabolotny, "Implementation of PCI Express Bus Communication for FPGA-based Data Acquisition System," in *Proceedings of Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2012*, vol. 8454, pp. 1-6, Oct. 2016.
- [ 8 ] S. M. Ryu, "Development of FPGA-based Meteorological Information Data Receiver Circuit for Low-Cost Meteorological Information Receiver System for COMS," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 19, no. 10, pp. 2373-2379, Oct. 2015.
- [ 9 ] J. S. Kang, and M. S. Kang, "FPGA Implementation of ARIA Crypto-processor Based on Advanced Key Scheduling," *Journal of Security Engineering*, vol. 13, no. 6, pp. 439-450, Dec. 2016.
- [ 10 ] Xilinx. DMA/Bridge Subsystem for PCI Express v4.0 Product Guide [Internet]. Available: [https://www.xilinx.com/support/documentation/ip\\_documentation/xdma/v4\\_0/pg195-pcie-dma.pdf](https://www.xilinx.com/support/documentation/ip_documentation/xdma/v4_0/pg195-pcie-dma.pdf).
- [ 11 ] Xilinx. ZC706 Evaluation Board for the Zynq-7000 XC7Z045 SoC User Guide [Internet]. Available:

[https://www.xilinx.com/support/documentation/boards\\_and\\_kits/zc706/ug954-zc706-eval-board-xc7z045-ap-soc.pdf](https://www.xilinx.com/support/documentation/boards_and_kits/zc706/ug954-zc706-eval-board-xc7z045-ap-soc.pdf)

- [12] PCI-SIG, *PCI Express Base Specification, Revision 3.0*, PCI-SIG Std., 2010.
- [13] J. Lawley, "Understanding Performance of PCI Express Systems," Xilinx: White Paper WP350, Oct. 2014.



**이 진 (Jin Lee)**

한국과학기술원 (KAIST) 전기 및 전자공학 박사

삼성전자 책임/수석 연구원

현재 평택대학교 조교수

※관심분야: 무선 통신 반도체, 임베디드 시스템, System on Chip