

## Implementation of High Speed Image Data Transfer using XDMA

Hyeok-Jin Gwon\*, Doo-Hyun Choi\*\*

\*Senior Engineer, ATE Team, Hanwha Systems, Gumi, Korea

\*\*Professor, School of Electronics Engineering, Kyungpook National University, Daegu, Korea

## [Abstract]

In this paper, we present an implementation of high speed image data transfer using XDMA for a video signal generation / acquisition device developed as a military test equipment. The technology proposed in this study obtains efficiency by replacing the method of copying data using the system buffer in the kernel area with the transmission and reception through the DMA engine in the FPGA. For this study, the device was developed as a PXIe platform in consideration of life cycle, and performance was maximized by using a low-cost FPGA considering mass productivity. The video I/O board implemented in this paper was tested by changing the AXI interface clock frequency and link speed through the existing memory copy method. In addition, the board was constructed using the DMA engine of the FPGA, and as a result, it was confirmed that the transfer speed was increased from 5~8Hz to 140Hz. The proposed method will contribute to strengthening defense capability by reducing the cost of device development using the PXIe platform and increasing the technology level.

▶ **Key words:** PXIe, XDMA, FPGA, Military Test Equipment, ViGA

## [요 약]

본 논문에서는 군용시험장비로 개발된 시험용 영상생성/수집 장치에 XDMA를 활용하여 고속 이미지 데이터 전송을 구현한다. 본 연구에서 제안하는 기술은 커널영역에서 시스템버퍼를 사용하여 데이터를 복사하는 방법을 FPGA내 DMA 엔진을 통한 송수신으로 대체하여 효율성을 얻는다. 본 연구를 위해 장치는 Life Cycle을 고려하여 PXIe 플랫폼으로 개발하였으며, 양산성을 고려하여 저가의 FPGA를 활용하여 퍼포먼스를 최대화하였다. 본 논문에서 구현한 영상입출력보드는 기존의 메모리복사방식을 통해 AXI 인터페이스 클럭 주파수, 링크속도를 변경하여 시험하였다. 그리고 FPGA의 DMA 엔진을 사용하여 보드를 구성하였으며, 그 결과 전송속도는 기존의 5~8Hz에서 140Hz로 증가함을 확인하였다. 제안된 방법은 PXIe 플랫폼을 이용한 장치개발의 비용절감, 기술수준을 높여 국방력 강화에 기여할 것이다.

▶ **주제어:** PXIe, XDMA, FPGA, 군용시험장비, ViGA

- 
- First Author: Hyeok-Jin Gwon, Corresponding Author: Doo-Hyun Choi
  - Hyeok-Jin Gwon (hyeokjin.gwon@hanwha.com), ATE Team, Hanwha Systems
  - Doo-Hyun Choi (dhc@ee.knu.ac.kr), School of Electronics Engineering, Kyungpook National University
  - Received: 2020. 04. 27, Revised: 2020. 06. 29, Accepted: 2020. 07. 03.

## I. Introduction

국방용 시험장비는 시험대상품(Unit Under Test, UUT)의 다양한 기능을 테스트하여 불량여부를 진단할 수 있도록 개발된다. 시험장비는 시험대상품의 발전에 따라 그에 상응하는 인터페이스를 제공해야 하며, 시험대상품의 Life Cycle이 통상 20년 이상이므로 장기적인 유지보수를 고려하여 개발되어야 한다[1]. 최근의 시험장비는 시험대상품과 동일한 시기에 납품이 되도록 계약이 이루어지는 경우가 대부분이다. 일반적으로 시험대상품의 개발과 동시에 시험장비를 개발하지만, 실제 시험대상품으로 시험할 수 있는 기간은 충분하지 않으며, 설계변경이 진행 중인 시험대상품으로 시험해야하거나 부품단종 발생 등의 상황에 유연하게 대처할 수 있는 시험장비 시스템의 구성은 필수이다.

최근의 시험장비는 표준화된 구조의 PXIe(PCI eXtension for Instrumentation Express) 플랫폼을 차용한 장비가 늘어나고 있다. 또한, PBA(Printed Board Assembly) 설계에서 FPGA의 소프트웨어 프로세서나 하드웨어 프로세서를 활용함으로써 집적도와 유연성을 증가시키는 추세이다. 이러한 경향을 바탕으로 기존의 K21 전투보병 장갑차에 장착되는 사수조준경과 차장조준경의 시험을 위한 ViGA(Video Signal Generation/Acquisition) 보드가 개발되어 사용하고 있다. ViGA 보드는 영상입출력 시험을 위해 아날로그 및 디지털 영상신호를 생성하고 수집하는 보드이다. 이 보드는 샘플영상을 생성하여 자체점검을 수행하고 시험대상품의 영상을 수집하여 일반 모니터에 전시할 수 있도록 영상처리를 수행한다[2]. 본 논문에서는 ViGA 보드의 PXIe용 커넥터를 활용하여 PCIe(PCI express) 버스통신과 FPGA를 활용한 프레임그래버 기능을 수행하는 PXIe 보드를 구현하고자 한다. 이 PXIe 보드는 테스트 영상을 생성하는 기존의 기능뿐만 아니라, 생성되는 영상을 PCIe 버스통신을 이용하여 호스트 컴퓨터와 영상데이터를 송수신하고 디스플레이장치에 전시한다.

본 논문에서는 개발과정 간 하드웨어, FPGA, 윈도우 디바이스 드라이버, 응용프로그램의 설계에 대해 기술한다. 특히, 윈도우 디바이스 드라이버와 Xilinx FPGA 칩을 활용하여 가장 간단한 PCIe 송수신장치에서부터 저비용 대비 퍼포먼스를 발휘할 수 있는 영상처리장치에 이르기까지의 시험과정을 순차적으로 기술하였다. 최종적으로는 디바이스와 메모리 간 DMA(Direct Memory Access)를 활용하여 최대속도를 낼 수 있음을 확인하였다. 그 결과에 도달하기까지 FPGA 설계, 드라이버 설계 등의 개선과정과 방법을 상세하게 기술하고자 한다.

본 논문의 구성은 다음과 같다. II장에서는 PCIe와 PXIe의 구조와 규격, 성능을 소개한다. III장에서는 ViGA 보드의 하드웨어, FPGA의 회로설계를 살펴본다. 그리고 운영체제와 ViGA 보드 사이의 데이터송수신을 위해 커널영역에서 시스템버퍼를 통한 통신방법을 확인하고 성능을 측정한다. IV장에서는 실시간영상처리를 위해 FPGA의 DMA 엔진을 사용하는 방안을 제시하고, ViGA 보드에 적용하는 방법을 기술한다. 시험결과를 통해 개선된 성능을 확인하고 기존방법과 비교하였다. 마지막으로 V장에서는 결론과 함께, 본 논문에서 개발한 장치를 활용한 국방산업 기여방안을 서술하였다.

## II. Preliminaries

### 1. PCI Express Specification

PCIe는 프로세서와 시스템의 I/O 장치들을 연결하기 위하여 널리 사용된다. PCIe는 PCI의 병렬 버스 기술의 성능한계를 극복하기 위하여 개발된 기술로써 고속 신호 전송 기법에 기반한 직렬 버스 기술이다. PCIe는 이전 세대 기술인 PCI와 호환을 유지하도록 개발된 기술 표준으로, 다만 패킷 기반의 직렬 버스 기술을 적용한 구조상 차이점이 있다.

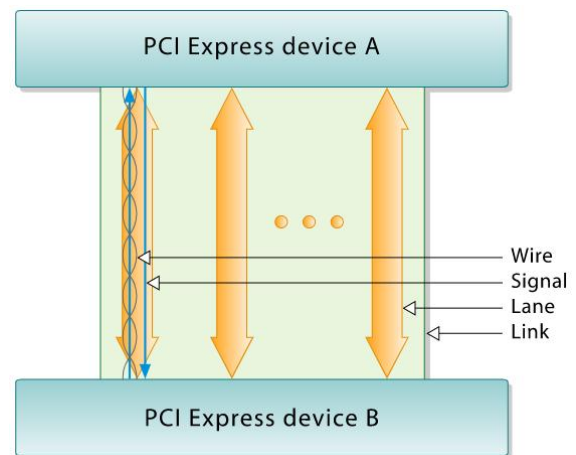


Fig. 1. PCIe Link Structure

PCIe 장치와 시스템 간의 연결을 Fig. 1과 같이 Link라고 하며 이 Link는 Lane이라는 연결구조로 이루어진다. Lane은 차동 신호를 전달하는 2쌍의 Signal로 이루어져 있으며 한 쌍은 데이터의 수신, 다른 한 쌍은 데이터의 송신을 담당하여 양방향 통신(Full Duplex)을 가능하게 한다[3]. 따라서 각 Lane은 4개의 Wire(2쌍의 Signal)로 구성된다. 더 높은 대역폭을 지원하기 위해 PCIe 장치 및 시

시스템은 동시에 여러 Lane을 사용하는 Link를 지원 할 수 있다. 예를 들어 Lane당 250MB/s의 데이터 전송인 경우, x16은 4GB/s(250MB/s\*16)의 대역폭을 갖는다[4]. 프로세서와 I/O 장치간의 연결을 위한 시스템 버스인 PCIe 규격은 Table 1과 같이 단일 Lane 기준 2.5Gbps(Gen1)를 시작으로, 8Gbps급의 성능(Gen3)을 가지는 규격이 널리 사용되고 있다[5].

Table 1. PCIe Specification

|                          |           |           |        |        |
|--------------------------|-----------|-----------|--------|--------|
| PCIe Version             | 4.0       | 3.0       | 2.0    | 1.0    |
| Introduced               | 2017      | 2010      | 2007   | 2003   |
| Throughput [MB/s]        | 1969      | 984       | 500    | 250    |
| Line code                | 128b/130b | 128b/130b | 8b/10b | 8b/10b |
| Data Transfer Rate[GT/s] | 16.0      | 8.0       | 5.0    | 2.5    |

### 2. PXI Express Specification

PXI(PCI eXtensions for Instrumentation)는 측정 및 자동화 시스템을 위한 견고한 PC 기반 플랫폼이다. PXI는 PCI 전기 버스 기능을 CompactPCI의 견고한 모듈형 Eurocard 패키지와 통합한 후, 특화된 동기화 버스 및 주요 소프트웨어 기능을 추가하였다. PXI는 측정 및 자동화 시스템을 위한 고성능 및 저가형의 배포 플랫폼이다. PXI 시스템은 제조 테스트, 군사 및 우주항공, 머신 모니터링, 자동차 및 산업 테스트와 같은 어플리케이션에 사용된다[6].

Table 2. PXI related Modules




| Chassis   | PXI board   | PXIe board  |
|---|---|---|
|  |  |  |

Table 2와 같이 PXIe board와 PXI board는 기구적인 규격이 다르며 전기적인 규격도 차이가 있다. PXIe는 PCIe 직렬버스에 기반하여 동작하며, PXI는 PCI 병렬버스에 기반하여 동작한다. 이러한 이유로 고속 데이터의 전송에는 PXIe가 PXI보다 빠른 속도로 처리가 가능하다. 또한 PXIe 장치의 슬롯연결은 Link의 Lane폭을 기반으로 변동되지 않는다[4]. 많은 수의 슬롯을 지원하는 Chassis는 상대적으로 고비용이며, 경제적인 시스템을 구축하기 위해서는 세심한 시스템 설계가 요구된다. 이러한 이유로 대부분

의 Chassis는 PXI와 PXIe 보드를 공용으로 사용가능한 하이브리드 슬롯을 일부 제공한다. 하지만 일반적으로 시중에서는 PXI/CompactPCI 장치가 PXIe 장치보다 훨씬 선택의 폭이 더욱 넓다.

## III. Video Signal Generation-Acquisition (ViGA) Board

### 1. Architecture of the ViGA Hardware

하드웨어는 기존 K21 장갑차 조준경용 시험장비의 구성품인 영상신호 생성/수집 회로카드인 ViGA 보드를 활용하였다. ViGA 보드의 하드웨어 구성은 Fig. 2와 같으며 FPGA부, 아날로그 영상부, 디지털 영상부로 구성된다. 아날로그 영상부는 모의영상을 생성하는 부분이며, RS-170(mono)과 NTSC(color)신호를 생성해 Video DAC, Video Encoder를 각각 거쳐 RS-170, RS170A 영상을 생성한다. RS-170 신호는 ADV7125와 AD8042를 활용하여 구현되었다. NTSC 신호는 ADV7342와 ADA4430을 사용하여 구현되었다. 디지털 영상부는 Camera Link 영상을 입력받아 FPGA로 전달하여 일반 모니터로 확인가능한 DVI 규격의 영상으로 변환한다. 영상의 입력을 위해 SN65LVDS 94A(LVDS Receiver) IC를 사용하였다.

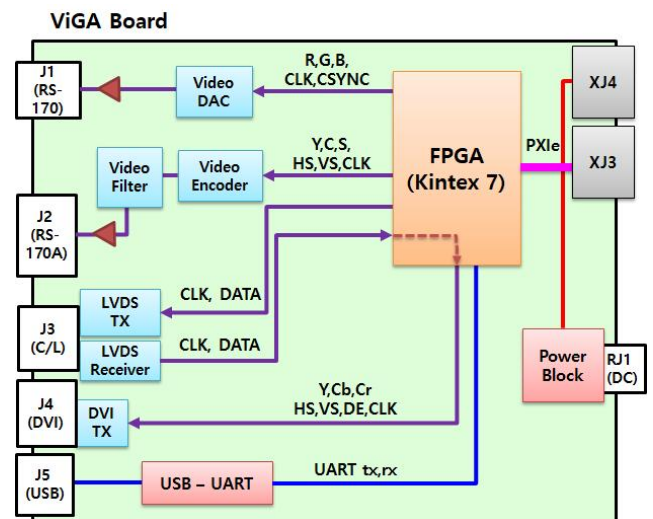


Fig. 2. Design of ViGA Hardware

FPGA는 Xilinx의 Kintex7 XC7K160T 칩을 사용하였다. Fig. 3과 같이 아날로그 영상을 생성하기 위해 MicroBlaze와 Video Pattern Generator IP를 사용하였

으며, 디지털영상을 생성하기 위해 MicroBlaze, Video Timing Generator, DVI Video Generator IP를 사용하였다. 또한 입력되는 LVDS 영상신호를 DVI 규격으로 변환하기 위해 DPRAM(Dual Port BRAM)을 활용하였다. 이러한 설계를 통해 Video Pattern Generator에 의한 아날로그 샘플영상 출력기능을 수행한다.

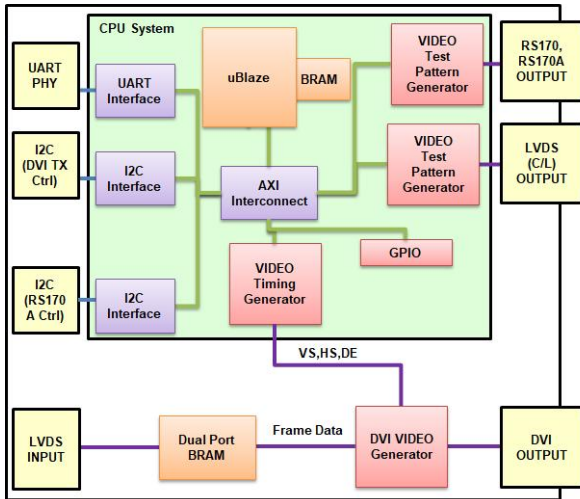


Fig. 3. Internal Structure of FPGA

ViGA 보드의 하드웨어는 FPGA와 PXIe를 연결하는 구조이다. FPGA는 검증된 IP 라이브러리, FPGA 프로세서를 사용할 수 있다. 야전환경에서 활용 가능한 PXIe 시스템에 FPGA를 활용함으로써, 개발 간 잠재적인 휴먼에러와 검증시간을 줄일 수 있으며, 향후 설계변경이 용이하다는 장점을 가진다.

2. Data Transfer between ViGA and Host

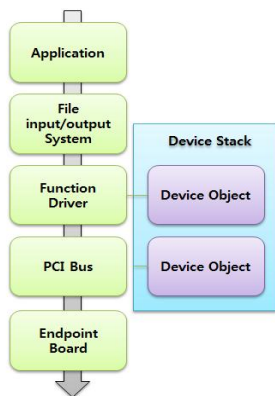


Fig. 4. Device Access using File Input/Output System

윈도우즈 OS에서는 ViGA 보드의 내부메모리에 데이터를 송수신하기 위해 파일입출력시스템을 사용한다. 이 방법은 Fig. 4와 같이 운영체제의 파일입출력시스템과 장치의 디바이스 스택을 통해 ViGA 보드에 접근하는 것이다. ViGA 보드의 드라이버는 디바이스 오브젝트를 생성하며, 디바이스 오브젝트의 집합인 디바이스 스택을 생성한다. 디바이스 스택에 접근하기 위해 응용프로그램은 파일에 대해 열기, 쓰기, 읽기, 닫기 기능을 수행할 수 있다. 이러한 기능은 Win32 API 함수인 CreateFile(), ReadFile(), WriteFile(), CloseHandle()로 실행된다. 응용프로그램이 해당 함수를 수행하면 파일입출력시스템을 통해 ViGA 보드와 운영체제 간 읽기, 쓰기가 수행된다. 응용프로그램이 파일에 대해 읽기, 쓰기 작업을 수행할 때, I/O 관리자는 IRP(I/O Request Packet)를 만들어 커널드라이버, 즉 디바이스 스택으로 IRP를 전달한다. IRP를 받은 디바이스 오브젝트는 명령어와 파라미터를 확인 후, 디바이스 드라이버의 코드를 수행한다.

|                            |             |               |                     |              |  |
|----------------------------|-------------|---------------|---------------------|--------------|--|
| 31                         |             | 16 15         |                     | 0            |  |
| Device ID                  |             | Vendor ID     |                     | 00h          |  |
| Status                     |             | Command       |                     | 04h          |  |
| Class Code                 |             |               | Revision ID         |              |  |
| BIST                       | Header Type | Lat. Timer    | Cache Line S.       |              |  |
| 10h                        |             |               |                     |              |  |
| 14h                        |             |               |                     |              |  |
| 18h                        |             |               |                     |              |  |
| 1Ch                        |             |               |                     |              |  |
| 20h                        |             |               |                     |              |  |
| 24h                        |             |               |                     |              |  |
| 28h                        |             |               |                     |              |  |
| Cardbus CIS Pointer        |             |               |                     |              |  |
| Subsystem ID               |             |               | Subsystem Vendor ID |              |  |
| 2Ch                        |             |               |                     |              |  |
| Expansion ROM Base Address |             |               |                     |              |  |
| Reserved                   |             |               |                     | Cap. Pointer |  |
| 34h                        |             |               |                     |              |  |
| Reserved                   |             |               |                     |              |  |
| 38h                        |             |               |                     |              |  |
| Max Lat.                   | Min Gnt.    | Interrupt Pin | Interrupt Line      |              |  |
| 3Ch                        |             |               |                     |              |  |

Fig. 5. PCI Configuration Space

ViGA 보드에는 Fig. 5와 같이 버스에서 장치를 인식시키기 위해 디바이스 정보를 저장하는 Configuration Space라는 데이터구조체가 존재한다. 이 데이터구조체에 존재하는 PCIe 장치정보를 버스와 공유함으로써 ViGA 보드는 운영체제와 연결이 가능하다. 또한 Configuration Space에는 BAR(Base Address Registers)가 존재하며, BAR는 디바이스 드라이버와 운영체제가 PCI 장치의 일반적인 입출력 데이터를 저장하기 위해 주로 사용되어지는

공간이다. 사용자레벨에서 커널레벨에 해당하는 BAR공간의 주소를 읽거나, 쓰기 위해 드라이버는 MmMapIoSpace 함수를 제공한다. MmMapIoSpace는 커널에서만 접근가능한 물리메모리 주소를 가상주소로 매핑해주는 함수이다[7]. 이러한 함수수행의 결과로서 응용 프로그램은 커널레벨의 주소에 접근할 수 있으며, 데이터를 송수신 할 수 있게 된다.

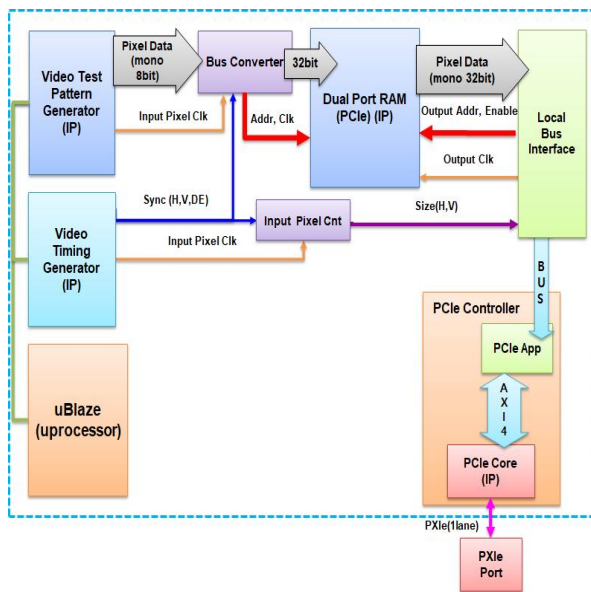


Fig. 6. Signal Flow inside FPGA

ViGA 보드에서는 Fig. 6과 같이 Video Pattern Generator와 Video Timing Generator IP를 이용하여 640\*480픽셀의 이미지를 만들어낸다. 해당 이미지는 1픽셀에 32bit의 데이터를 가지고 있으며 FPGA의 구동과 함께 DPRAM으로 영상데이터를 전달하게 된다. DPRAM에 쓰여진 영상데이터는 BAR 공간에 저장되며, 계속 업데이트된다. BAR에 저장된 데이터를 운영체제에서 읽어가기 위해서는 PCIe 컨트롤러를 구성해야 하며, 7 Series Integrated Block for PCI Express IP를 컨트롤러로 활용하였다. 드라이버에서 메모리 복사 방식은 Buffered 방식을 사용하였으며 32bit 데이터가 시스템버퍼에 복사될 수 있도록 WRITE\_SYSTEM\_BUFFER\_ULONG 함수를 사용하였다. 파일입출력시스템을 사용하여 데이터 전송률을 높이기 위해 드라이버는 BAR의 물리주소를 가상주소로 변환하여 OS에 제공한다. 그리고 응용프로그램은 DeviceIoControl API를 사용하여 가상주소의 포인터로 접근 후 BAR의 모든 주소에 접근하는 방법을 사용하였다. 또한 응용프로그램 연산과정은 데이터수신 쓰레드 외에는 최소화하여 속도측정 간 다른 요소는 영향성이 거의 없도록

수행하였다. 시스템구성 후 영상신호의 1프레임(640\*480\*32bit)을 OS로 수신하는데 소요되는 시간은 0.2초로 측정되었다. 이를 주파수로 환산하면 5Hz로 계산된다. 사수조준경과 차장조준경 시험장비로 설계된 ViGA 장치는 일반 VGA 모니터로 영상을 전시하기 때문에 운영체제와의 데이터송수신이 필요하지 않다. 하지만 PXIe 플랫폼을 활용하고 영상캡처장치의 역할을 수행하기 위해서는 운영체제와의 데이터송수신이 필수적이다. 기존의 파일입출력시스템과 FPGA의 7 Series Integrated Block for PCI Express IP를 활용한 방법은 영상을 전시하기에는 적합하지 않다. 시험장비가 영상시험을 수행하기 위해서는 최소 25Hz 이상의 주파수가 요구된다. 요구사항을 만족하기 위해 COTS(Commercial Off The Shelf)모듈을 사용할 수 있다. 그러나 비표준 영상을 처리해야 하는 국방사업에서는 FPGA를 사용하여 성능을 만족하는 장치를 개발하는 방법이 범용성과 장기적인 유지보수 관점에서 활용도가 높다고 볼 수 있다.

## IV. Experimental Results

### 1. Data Transfer without DMA

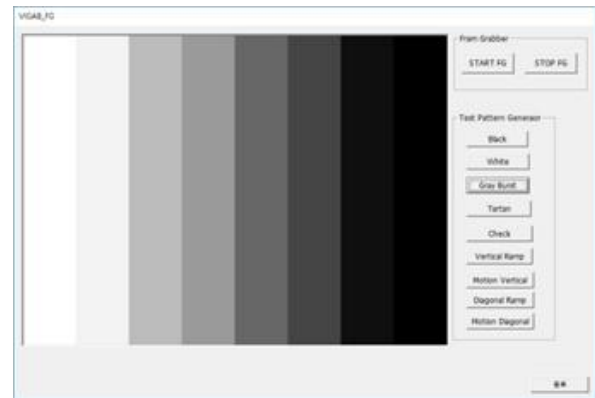


Fig. 7. Application program

ViGA 보드의 성능을 측정하기 위한 응용프로그램을 Fig. 7과 같이 개발하였다. 프로그램의 개발환경은 Intel Celeron 1020E CPU, 4 GB RAM, Windows 10 64bit 이다. IP에서 제공하는 샘플영상에 대한 뷰어로서 기능하며, 1프레임에 해당하는 영상을 지속적으로 수신할 수 있도록 쓰레드가 동작한다. 또한 쓰레드의 1회 수행은 1프레임의 영상데이터 수신을 의미하므로 수행시간에 의해 주파수를 계산할 수 있다.

Xilinx의 IP인 7 Series Integrated Block for PCI Express IP는 기본 설정을 통해 PCIe Lane 수, PCIe 링크 속도, AXI 클럭 등을 변경 할 수 있다. AXI 클럭은 62.5, 125, 250MHz로 설정이 가능하며, 기존에는 62.5MHz에서 1프레임을 얻는데 0.2초가 소요되었다. 그리고 250MHz에서 1프레임을 얻는데 0.17초가 소요되었다. AXI 클럭이 4배 증가하더라도 1프레임 당 소요시간은 15% 감소하였다. 그리고 해당 AXI 인터페이스 클럭 주파수의 경우에는 FPGA 종류에 따라 250MHz의 동작속도를 갖도록 설계하는 것은 다소 무리가 있으므로 신중하게 선택해야 한다[8].

ViGA 보드의 경우는 1Lane으로 구성되어 있다. FPGA는 PCIe 2.0을 지원하는 Kintex7 칩을 사용하며 CPU도 PCIe 2.0을 지원한다. 그러므로 ViGA 보드 내부의 7 Series Integrated Block for PCI Express IP의 링크속도를 2.5GT/s(Giga Transfer per Sec), 5.0GT/s로 선택 가능하다. 2.5GT/s에서는 1프레임 수신에 걸리는 시간이 0.17초이며, 5.0GT/s에서는 0.135초가 소요되었다. 최종적으로 1프레임에 해당하는 640\*480\*8bit, 307KByte의 데이터를 전송하는데 걸리는 시간은 0.135초이다. 전송속도는 2.3MByte/s이며, 주파수는 Fig.8과 같이 7.4Hz로 나타났다.

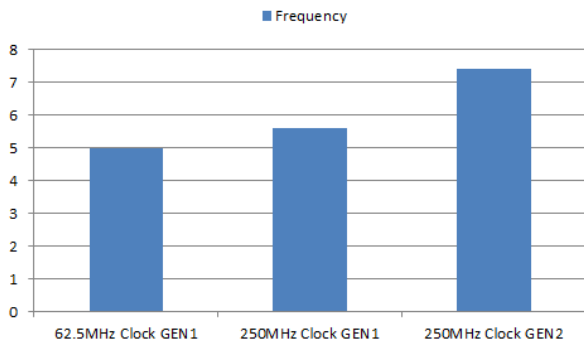


Fig. 8. Test Result

Fig. 8의 결과와 같이 파일입출력시스템을 이용한 데이터 수신주파수는 PCIe 링크속도, AXI 클럭에 따라 높여질 수 있다. 하지만 평균적인 실시간처리를 위해서는 초당 30프레임이 가능해야 한다[9]. 이는 Lane 수의 증가, PCIe 상위버전의 사용 등과 같이 하드웨어 성능을 높임으로서 목표치를 달성할 수도 있지만, 대용량 데이터의 실시간 처리를 위해서는 DMA 시스템을 활용해야 한다는 결론으로 귀결된다.

## 2. Data Transfer using XDMA

PCIe 시스템에서 DMA를 활용하는 다양한 방법이 있지만, FPGA의 IP를 활용하는 방안이 시스템의 유연성 및 확장성 측면에서 장점을 가진다. 640\*480, 1024\*768이 아닌 표준을 벗어난 비표준영상을 처리하는 측면에서는 더욱 활용도가 높을 것이다. 이를 위해 기존의 시스템에서 사용한 PCIe 컨트롤러 7 Series Integrated Block for PCI Express IP 대신 XDMA(DMA for PCI Express Subsystem) IP를 활용하였다. XDMA IP에 내장된 DMA 엔진을 사용해 호스트 메모리 대상으로 DMA를 수행할 수 있다[10]. 응용프로그램이 FPGA의 DMA 엔진을 사용해 DMA를 시작하기 위해 디바이스 드라이버를 통해 DMA 데이터 전송의 크기와 출발지, 도착지 주소들과 같은 정보들을 FPGA의 DMA 엔진에 전달하여야 한다[11]. 해당 IP를 활용하기 위해 Xilinx에서 제공하는 윈도우드라이버와 응용프로그램을 활용할 수 있다.

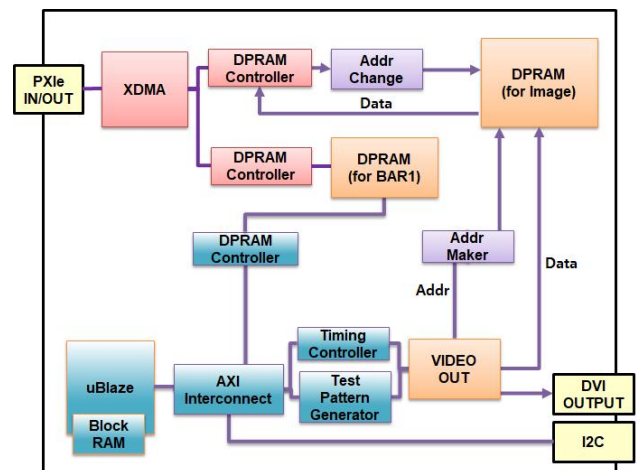


Fig. 9. Internal Structure of FPGA using DMA

Fig. 9의 FPGA 내부 구조는 DMA의 활용을 위해 구성되어졌다. 640\*480\*32bit 영상을 재현하기 위해서는 DPRAM 메모리가 Width 32, Depth 307,200(640\*480)으로 설정되어야 하나, FPGA를 설계하기 위한 Vivado 툴의 라이선스 제한이 발생하며, 해결을 위해 추가적인 비용이 발생한다. 라이선스의 제한없이 해당 사이즈 영상의 처리를 수행하기 위해 RGB의 32bit 중 하위 8bit인 Alpha 비트를 제외하고 24bit 단위로 DPRAM을 구성하였다. 해당 DPRAM은 Width 24, Depth 307,200으로 변경하였다. 영상데이터용 DPRAM은 Addr Maker 블록에 의해 데이터를 Address 0에서 쓰기를 시작한다. 이 때 주소는 +1단위로 증가하여 쓰기작업을 수행한다. 하지만 XDMA에서

DPRAM의 데이터를 읽어올 때, Addr Change 블록에 의해 주소는 +4단위로 증가한다. 추가적으로 Bar1 DPRAM 및 XDMA의 AXI-Lite Master Interface를 이용하면 BAR에 접근하기 위해 수행했던 Fig. 6의 PCIe App 블록의 생성 및 PCIe App 블록과 PCIe Core, Local Bus Interface 블록 간의 연결이 필요하지 않다. XDMA는 실제 사용결과 Vivado 2019.1버전을 사용하였을 때 정상적으로 기능이 동작하였다. 그리고 1Lane, PCIe 링크속도 2.5GT/s, AXI 클럭을 62.5MHz로 XDMA IP를 설정하여 시험을 진행하였다. 시험결과 속도는 0.014초 (데이터수신:0.007초, 응용 프로그램 화면전시:0.007초)로 측정되었다. 이는 640\*480\*32bit 기준으로 70Hz이다. 그리고 프로그램 화면전시시간을 제외한 시간으로 환산 시, 140Hz이다.

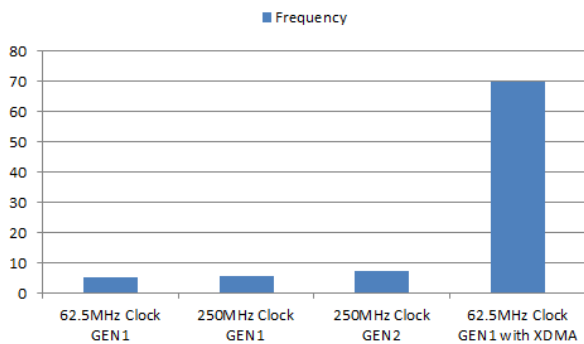


Fig. 10. Test Result using XDMA

개별 모델의 주파수를 Fig. 10과 같이 확인하였다. 기존의 초당 2.3MByte에서 86MByte로 영상수신 성능을 개선하였다. 250MHz AXI 클럭, 링크속도 5.0GT/s 속도를 적용한 모델과 비교하면 37배의 속도향상이 발생한다. 그리고 응용프로그램의 전시시간을 제외하면 초당 172MByte이며 74배 향상하였다. 기존모델이 PCIe 2.0과 250MHz의 AXI 클럭을 적용하였기에 PCIe 1.0과 62.5MHz의 AXI 클럭을 적용한 현재모델은 더욱 속도향상요소를 가질 것이다. 파라미터별 성능변화에 대한 분석결과도 확인할 수 있다. AXI 클럭을 62.5MHz에서 250MHz로 변경 후 시험했을 때, 초당 5프레임에서 5.6프레임으로 12% 성능이 향상되었다. PCIe 링크속도를 2.5 GT/s에서 5.0 GT/s로 변경 후에는 5.6프레임에서 7.4프레임으로 32% 성능이 향상되었다. 성능향상을 위한 요소는 XDMA IP의 사용, PCIe 링크속도 조정, AXI 클럭 조정 순으로 영향성이 높았다.

## V. Conclusions

본 논문에서는 시험용 영상신호 생성/수집 회로카드의 PXIe 플랫폼 적용 및 성능개선을 설계하고 시험하였다. 기존의 하드웨어를 활용하여 FPGA의 7 Series Integrated Block for PCI Express IP와 디바이스 드라이버의 파일 입출력시스템을 이용한 시스템버퍼 복사방식에 대한 시험 결과를 확인하였다. 또한 영상처리장치의 성능을 충족하기 위해 FPGA의 XDMA IP와 디바이스 드라이버를 활용하여 시험하였다. 그 결과, 5~8Hz의 주파수는 70Hz로 향상하며, 전시시간을 제외하면 140Hz로 계산되는 것을 시험으로 확인하였다.

최근에는 PXI, PXIe 장치들의 선택권이 넓어지고 더욱 다양한 제품들을 상용으로 구매할 수 있다. 반면에 FPGA의 성능향상으로 슬롯의 낭비를 줄임과 동시에, 기능의 중첩을 가능하게 하는 장치를 개발하려는 시도도 가능하다. 전반적으로 PXI/PXIe 제품군이 다양하다고 해도, PCI/PCIe 제품군과 비교하면 소비자층이 적기 때문에 해당개발방안은 유용할 수 있다. 특히 국방산업에서는 환경요소 기준을 만족시키기 위해 PXI, PXIe 기반의 장치들이 차용되고 있다. 본 논문의 방법을 통해 기 운용중인 유사한 장치의 활용도를 높일 수 있을 것이다. 본 논문에서 제시한 방식의 장점은 크게 3가지로 요약할 수 있다. 첫째, 환경시험을 통해 국방산업에서의 환경요구조건을 만족할 수 있다. 둘째, 640\*480, 1024\*768이 아닌 비표준 사이즈의 영상처리에 대해 유연하게 대처할 수 있다. 셋째, FPGA를 이용해서 필요성능을 구현하였으므로, 향후 설계변경에 유연하게 대처할 수 있다.

본 연구는 국방산업 외에도 FPGA기반의 PCIe 연결 시스템을 사용하는 다양한 산업현장에서 활용할 수 있다. FPGA 기반의 PCB 패턴 검출 프로그램[12], 적외선 영상 기반 안정화 성능 개선[13], 대용량 영상 데이터의 저장 및 통신을 위한 PCIe 설계[14], 실시간 영상처리를 위한 SVM 분류기 FPGA 구현[15] 등에서 활용되고 있다. 하지만 타 논문에서 제시한 FPGA는 대부분 ARM 코어를 내장한 가격대가 높은 Zynq 디바이스이거내[13][14][15], Virtex 5 FPGA 디바이스인 경우 초당 10회 이상의 영상처리 성능을 보여주었다[12]. 본 연구를 적용한다면 소형 이미지 처리 부분에서 기존대비 저비용과 고효율의 장비를 제작할 수 있을 것이다. 본 연구의 시험 및 검증결과를 제시함으로써 PXIe 플랫폼을 이용한 장치개발의 비용절감, 기술 활용도를 높여 국방력 강화와 산업발전에 기여할 것으로 기대한다.

## REFERENCES

- [1] M. Y. KIM, "A Study on Determining Method of Integrated Logistics Support Elements in Defense system Test Development Phase," KIIE 2nd Annual Technology Conference, pp. 996-1001, Nov. 2011.
- [2] K. H. KIM, "Implementation of Video Signal Generation Acquisition Board for KIGPS/KICPS Tests," Summer Annual Conference of IEIE, pp. 852-854, Jun. 2018.
- [3] R. Budruk, "PCI Express Basics" PCI-SIG, pp. 3, 2007
- [4] National Instruments, <http://www.ni.com/ko-kr/innovations/white-papers/06/pxi-express-faq.html>
- [5] Y. W. KIM, Y. REN, W. H. CHOI, "Design and Implementation of an Alternate System Interconnect based on PCI Express," Journal of the Institute of Electronics and Information Engineers, Vol. 52, No. 8, pp. 74-85, Aug. 2015. DOI: 10.5573/IEIE.2015.52.8.074
- [6] National Instruments, <http://www.ni.com/tutorial/4811/ko/>
- [7] B. S. LEE, "Window Device Driver 2/e" Acorn, pp. 688, 2019.
- [8] J. LEE, "Implementation and Performance Evaluation of PCI express on Xilinx FPGA," Journal of the Korea Institute of Information and Communication Engineering, Vol. 22, No. 12, pp. 1667-1674, Dec. 2018. DOI: 10.6109/JKIICE.2018.22.12.1667
- [9] D. K. SHON, C. H. KIM, J. M. KIM, "An Implementation of a Video-Equipped Real-Time Fire Detection Algorithm Using GPGPU," Journal of The Korea Society of Computer and Information, Vol. 19, No. 8, pp. 1-10, Aug. 2014. DOI: 10.9708/JKSCI.2014.19.8.001
- [10] Xilinx, "DMA/Bridge Subsystem for PCI Express v4.1 Product Guide" pp. 19, 2019.
- [11] J. S. LEE, J. J. LEE, "Measuring Performance of FPGA/GPU direct communication via PCI express," Korea Computer Congress 2019, pp. 31-33, Jun. 2019.
- [12] C. H. MOON, "Implementation of an FPGA-based Frame Grabber System for PCB Pattern Detection," Journal of the KIECS, Vol. 13, No. 2, pp. 435-442, Apr. 2018. DOI:10.13067/JKIECS.2018.13.2.435
- [13] H. CHOI, J. H. CHO, Y. M. KIM, S. H. KANG, "Improve Stability of Military Infrared Image and Implement Zynq SoC," IEMEK Journal of Embedded Systems and Applications, Vol. 13, No. 1, pp. 17-24, Feb. 2018. DOI:10.14327/IEMEK.2018.13.1.17
- [14] C. H. HONG, "Design and Implementation of SW based on PCIe for Large Image Data Storage and Communication," Korea Software Congress 2019, pp. 56-58, Dec. 2019.
- [15] W. S. NA, S. W. HAN, Y. J. JEONG, "FPGA Design of SVM Classifier for Real Time Image Processing," Journal of IKEEE, Vol. 20, No. 3, pp. 209-219, Sep. 2016. DOI:10.7471/IKEEE.2016.20.3.209

## Authors



Hyeok-Jin Gwon received the B.S. degrees in Electronics Engineering from Yeungnam University, Korea, in 2010. Mr. Gwon joined the ATE Team of Hanwha Systems, Korea, in 2010. He is currently a Senior Engineer in the ATE Team, Hanwha Systems. He is interested in embedded test system for national defense.



Dr. Doo-Hyun Choi has received his BS degree in electronics engineering from Kyungpook National University (KNU), S. Korea, in 1991 and MS and Ph. D. degrees in electrical and electronics engineering from

Pohang University of Science and Technology (Postech) in 1993 and 1996, respectively. The title of his dissertation is "Neural network applications for autonomous land vehicles. After graduating, he joined to KNU as an assistant professor until Feb. 2000 and he moved to Seoul National University (SNU) as an assistant professor until Mar. 2003. He rejoined to KNU at Mar. 2003 and now he's managing "Intelligent information systems laboratory" and a full professor at the university. Also he is enrolled at Marquis Who's who in the World. From 2012 to 2013, he spent his sabbatical at the University of Florida, Gainesville, Florida, USA. He wrote more than 150 journal and conference papers and performed many projects relating on intelligent signal processing. His research interest includes intelligent systems, soft computing algorithms, signal processing and its application.