JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 2: 259~266, Feb. 2021

FX3 USB 3 브릿지 칩과 slave FIFO 인터페이스를 사용하는 FPGA 검증 시스템 구현

최병윤*

Implementation of FPGA Verification System with Slave FIFO Interface and FX3 USB 3 Bridge Chip

Byeong-Yoon Choi*

*Professor, Department of Computer Engineering, Dong-Eui University, Busan, 47340 Korea

요 약

USB 버스는 편리하게 사용할 수 있고 빠르게 데이터를 전송하는 장점이 있어서, FPGA 개발보드와 PC 사이의 표준적인 인터페이스이다. 본 논문에서는 Cypress FX3 USB 3 브릿지 칩에 대한 slave FIFO 인터페이스를 사용하여 FPGA 검증 시스템을 구현하였다. slave FIFO 인터페이스 모듈은 FIFO 구조의 호스트 인터페이스 모듈과 마스터 버스 제어기와 명령 해독기로 구성되며, FX3 브릿지 칩에 대한 스트리밍 데이터 통신과 사용자 설계 회로에 대한 메모리 맵 형태의 입출력 인터페이스를 지원한다. 설계 검증 시스템에는 Cypress FX3 칩과 Xilinx Artix FPGA (XC7A35T-1C5G3241) 칩으로 구성된 ZestSC3 보드가 사용되었다. C++ DLL 라이브러리와 비주얼 C# 언어를 사용하여 개발한 GUI 소프트웨어를 사용하여, 사용자 설계 회로에 대한 FPGA 검증 시스템이 다양한 클록 주파수 환경에서 올바로 동작함을 확인하였다. 설계한 FPGA 검증 시스템의 slave FIFO 인터페이스 회로는 모듈화 구조를 갖고 있어서 메모리맵 인터페이스를 갖는 다른 사용자 설계 회로에도 응용이 가능하다.

ABSTRACT

USB bus not only works with convenience but also transmits data fast and becomes a standard peripheral interface between FPGA development board and personal computer. In this paper FPGA verification system with slave FIFO interface for Cypress FX3 USB 3 bridge chip was implemented. The designed slave FIFO interface consists of host interface module based on FIFO structure, master bus controller and command decoder and supports streaming communication interface for FX3 bridge chip and memory-mapped input and output interface for user design circuit. The ZestSC3 board with Cypress FX3 USB 3 bridge chip and Xilinx Artix FPGA(XC7A35T-1C5G3241) was used to implement FPGA verification system. It was verified that the FPGA verification system for user design circuit operated correctly under various clock frequencies using GUI software developed by visual C# and C++ DLL. The designed slave FIFO interface for FPGA verification system has modular structure and can be applicable to the different user designs with memory-mapped I/O interface.

키워드 : FPGA, P/Invoke, USB 3 인터페이스, 검증 시스템

Keywords : FPGA, Platform invoke, USB 3 Interface, Verification System

Received 2 December 2020, **Revised** 5 December 2020, **Accepted** 23 December 2020

*** Corresponding Author** Byeong-Yoon Choi(E-mail:bychoi@deu.ac.kr, Tel:+82-51-890-1706)
Professor, Department of Computer Engineering, Dong-Eui University, Busan, 47340 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.2.259 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/li-censes/ 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. 서 론

시스템 설계 아이디어를 구현하는 방안은 소프트웨어 설계, 하드웨어 설계와 하드웨어와 소프트웨어 통합 설계(codesign) 형태로 나누어진다[1]. 디지털시계 등 소규모 설계는 LED, Text LCD, 키패드, 버튼 등을 갖는 교육용 Field Programmable Gate Array(FPGA) 보드로 검증이 가능하며 검증용 소프트웨어가 필요하지 않는다. 대규모 하드웨어 작품은 FPGA로 구현 시 대부분 통합설계 형태가 필요하며, 컴퓨터와 인터페이스 되어서 동작하므로, 이를 반영하는 검증 환경이 필요하다. 컴퓨터의 외부 인터페이스가 프린터 포트, 비동기 직렬 버스에서 현재 Universal Serial Bus(USB) 3 버스가 대중화되고 있다. 그러나 국내에서 발표되는 많은 FPGA를 이용한 시스템 설계 논문[2]의 검증 환경은 Universal Asynchronous Receiver/Transmitter(UART)를 사용하는 비동기 직렬 버스를 이용한 경우가 대부분으로, 실제 하드웨어의 빠른 동작 특성을 검증하지 못하고, 낮은 속도의 논리적인 동작 확인만 가능한 문제점이 있다. USB 인터페이스를 사용하는 경우에도 USB 2 버스를 사용하거나[3], 고속 스트리밍(streaming) 방식의 데이터 전달이 아닌 Inter-Integrated Circuit(I2C) 버스 등을 이용한 직렬 전송을 사용하는 경우가 많다. 따라서 빠른 동작 속도를 갖는 FPGA의 올바른 동작 특성 검증을 위해 USB 3 버스 인터페이스를 사용하는 하드웨어 검증 시스템을 구축할 필요가 있다. 그런데 시스템 설계자에게 사용자 설계 회로이외에 FPGA 내부에 USB 3 인터페이스 모듈을 직접 설계하는 것은 큰 부담이 된다. 현실적인 대안은 USB 3 신호를 FPGA 칩에 연결을 용이하도록 변환하는 USB 3 브릿지 칩을 사용하는 방안이다. USB 3용 인터페이스 브릿지 칩으로 대표적인 2가지는 FTDI사의 FT600/FT601 칩[4]과 Cypress사의 EZ-USB FX3 칩[5]이다. FT600은 16-비트 데이터 입출력 인터페이스를 제공하며, FT601과 FX3 칩은 32-비트 데이터 입출력 인터페이스를 지원하며, 사용자 설계 회로의 입출력 특성을 고려하여 다양한 동작 모드를 지원하고 있다. 이러한 USB 브릿지 칩에 대한 펌웨어 설정 및 입출력 신호가 매우 복잡하여 FPGA 설계자가 사용하기 어렵다[6]. 따라서 FPGA 보드 개발 업체가 USB3 브릿지 칩과 FPGA를 하나의 Printed Circuit Board(PCB) 보드로 제작하고, USB 3 브릿지 칩 사용을 위한 C++ Dynamic Link Library(DLL) 라이브러리와 단순한 예제 코드를 제공하고 있다. 본 연구에서는 이전 USB 1 버스 기반의 ZestSC1 보드를 사용한 설계 경험을 바탕으로 Cypress 사의 EZ-USB FX3 칩과 Xilinx Artix FPGA 칩을 내장한 Orange Tree사의 ZestSC3[7] 보드를 구매하여 USB 3 인터페이스용 FPGA 검증 환경을 구축 하였다.

본 논문에서는 USB 3버스용 Cypress FX3 브릿지 칩에 대한 모듈화 구조의 Slave FIFO 인터페이스 회로를 설계하고, FPGA로 구현 한 후 C# 검증용 소프트웨어를 사용하여 설계한 하드웨어의 올바른 동작을 검증하였다. Ⅱ장에서는 USB 3 인터페이스를 갖는 FPGA 설계 시 고려해야 할 사항을 설명하고, Ⅲ장에서는 FX3 slave FIFO 인터페이스 설계를 기술하였다. FPGA 구현과 기능 검증을 Ⅳ장에서 서술하였으며, Ⅴ장에서 결론을 맺는다.

Ⅱ. USB 3 인터페이스를 갖는 FPGA 설계 시 고려 사항

Cypress FX3 브릿지 칩과 FPGA 칩이 연결되는 방식은 일반적으로 slave First In First Out(FIFO) 와 master FIFO 방식으로 나뉜다[5,8]. slave FIFO 방식은 FX3 브릿지 칩 모듈이 슬레이브(slave)이고, FPGA가 데이터 전송의 주체가 되는 마스터(master)로 동작하는 방식이며, Master FIFO 방식은 FX3 칩이 데이터 전송의 주체가 되는 master로 동작하는 방식이다. 그림 1은 컴퓨터, EZ-USB FX3 모듈과 FPGA 칩이 slave FIFO 방식으로 연결되는 구조를 나타낸다.

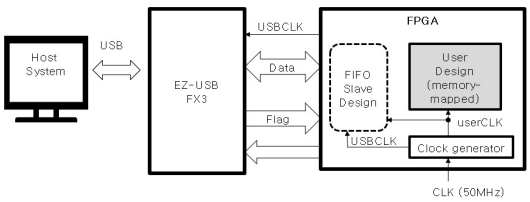

Fig. 1 FPGA verification system with slave FIFO interface for EZ-USB FX3

본 연구에서는 그림 1의 USB 버스용 FPGA 검증 시스템을 위한 설계 기준을 다음과 같이 결정하였다.

첫째, FX3 브릿지 칩과 FPGA 칩을 연결하는 방식으

로 FPGA 칩이 데이터 전송 주도권을 갖는 slave FIFO 방식을 채택하였다.

둘째, FX3 브릿지 칩에서 동작하는 클록 주파수와 사용자가 설계한 회로의 동작 주파수가 다를 수 있으므로, 2가지 종류의 클록 주파수로 동작하는 FIFO 구조의 송수신 버퍼를 갖는 호스트 인터페이스를 포함한다.

셋째, 호스트 인터페이스 모듈에 2가지 다른 클록 사용으로 클록 주파수 경계를 넘는 다수 비트로 구성된 데이터와 제어 신호의 동기화를 위해, 기존 체인 형태의 플립플롭 동기화 기술과 함께 참고 문헌[9]에서 제안한 그레이 코드(gray code) 방식을 함께 사용하였다.

넷째, 사용자 설계 회로는 주소와 제어 신호(Read, Write 등), 종료 상태 신호, 32-비트 데이터 신호 등으로 제어 가능한 메모리 맵(memory-mapped) 구조의 모듈로 설계하도록 하여 범용적인 인터페이스가 가능하도록 한다.

다섯째, 검증용 GUI 소프트웨어에서 제공하는 데이터와 이벤트 신호가 메모리 맵된 사용자 설계 모듈에 대한 주소, 데이터, 제어 신호로 변경되어야 하므로, 이를 위한 특화된 소수의 명령어를 정의한다.

여섯째, 정의한 명령어를 해독하고 사용자 모듈에 필요한 제어 신호와 주소, 데이터를 발생시키며, 사용자 설계 모듈의 상태 정보와 회로 동작 결과 값을 호스트 인터페이스 모듈로 전달하는 기능을 갖춘 별도의 마스트 버스 제어기와 명령어 해독기를 호스트 인터페이스 출력단에 두도록 한다.

이러한 설계 사양을 바탕으로 본 논문에서 제안하는 USB 3 버스용 slave FIFO 인터페이스를 갖는 FPGA 검증 시스템 구조는 그림 2와 같다. 그림 2에서 PC와 FX3 칩간 데이터 통신은 FPGA 보드 업체가 제공하는 C++ Application Programming Interface(API) 함수를 이용해서 이루어지며, C++ API 함수에 포함되어 FPGA 칩으로 전달되는 데이터의 양식과 내용은 설계자의 몫이 된다. 본 연구에서는 검증용 C# Graphic User Interface(GUI) 소프트웨어의 특정 이벤트 처리기에서, 검증용 소프트웨어의 텍스트 박스 혹은 이벤트 신호를 자체 정의한 특화된 명령어 형식으로 바꾸어 FX3 칩으로 전달하며, 마스트 버스 제어기와 명령어 해독기는 정의한 명령어를 해독하는 역할을 수행한다.

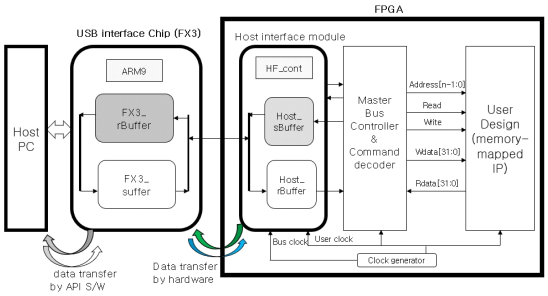

Fig. 2 Proposed FPGA verification system with slave FIFO interface based on USB 3

Ⅲ. FX3 Slave FIFO 인터페이스 설계

FX3 Slave FIFO 인터페이스는 호스트 인터페이스 모듈과 마스트 버스 제어기와 명령어 해독기로 구성된다.

3.1. 호스트 인터페이스 모듈

호스트 인터페이스 모듈은 사용자 설계 모듈의 클록 주파수와 FX3 칩의 클록 주파수(100 MHz)가 다르므로, 클록 경계를 넘는 신호에 대한 동기화와 병렬 데이터에 대한 스트리밍 동작을 위한 입출력 FIFO 모듈과 FIFO 제어 모듈을 포함한다. 송수신 FIFO의 읽기 포인터(read pointer)와 쓰기 포인터(write pointer)가 다른 클록 주파수를 사용할 수 있으므로, 데이터 신호가 클록 신호의 경계를 넘는 동기화 문제가 발생한다. 이러한 동기화 문제를 해결하기 위해 기존 동기화 기법과 함께 FIFO 제어기내에 이진 카운터(binary counter) 기반의 포인터와 그레이 카운터(gray counter) 포인터를 함께 사용하며 [9], 전송의 안전성을 위해 FIFO의 상태를 나타내는 기존 Full, Empty 신호이외에 Almost Full과 Almost Empty 신호를 함께 사용하였다. 그림 3은 호스트 인터페이스 모듈의 외부 인터페이스 신호를 나타낸다. 외부의 마스터 제어기는 SFIFO_Empty와 RFIFO_Full 신호를 활용하여 데이터 도착 여부와 busy 여부를 판단한다. 내부 송수신 FIFO는 현재 16개의 32-비트 구조를 갖고 있는데, 사용자 설계의 동작 특성에 따라 FIFO의 요소 수의 조정이 가능하다.

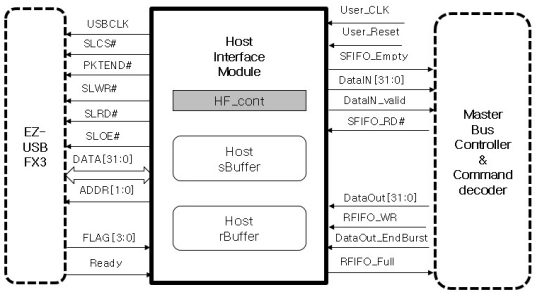

Fig. 3 I/O interface signals of host interface module

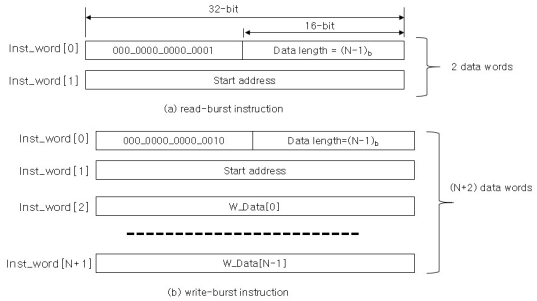

Fig. 4 Instructions to control user design
(a) read-burst instruction, (b) write-burst instruction

3.2. 사용자 설계 모듈을 제어하는 명령어 정의

FPGA 보드 업체가 제공하는 C++ DLL 라이브러리는 호스트 PC에서 FX3 칩 사이에 데이터를 송・수신하는 함수만 정의하고 있다. 따라서 GUI 형태의 검증용 소프트웨어에서 버튼 클릭 이벤트 발생 시 텍스트 박스(Textbox) 컨트롤 등에 담긴 데이터를 사용자 설계 회로에 적절한 형식의 데이터, 주소, 제어 신호로 전달하기 위해, 자체적인 명령어를 정의하여 해당 형식으로 명령어를 전달하는 방식이 바람직하다. 본 연구에서는 C++ DLL 함수를 통해 전달될 데이터에 담을 명령어 종류와 형식에 대해 여러 가지 방안을 검토하였다. 최종적으로 메모리 맵 방식이 주소에 다른 제어 기능을 정의할 수 있는 특성을 고려하여, 그림 4의 2가지 read-burst와 write-burst 명령을 채택하였다. read-burst 명령의 경우 2개의 32-비트 데이터로 구성되며, 첫 번째 32-비트의 상위 비트의 bit[16]=1로 read-burst를 판단할 수 있으며, 하위 16-비트는 읽을 데이터 수를 나타낸다. 읽을 데이터 수가 N일 경우 Data length 필드에서 (N-1) 값을 담도록 하였다. 두 번째 32-비트 워드에는 사용자 설계 회로에서 읽을 데이터의 레지스터의 시작 주소를 저장한다. burst 동작의 읽기를 수행하므로 마스터 제어기에서 시작 주소를 저장한 후 자동 증가시켜 사용하며, 읽은 데이터는 호스트 인터페이스 모듈의 수신 버퍼(rBuffer)에 저장된다. 이러한 burst-read 명령은 연산 결과를 읽는 역할과 함께 연산 종료의 상태 신호도 읽어서 호스트로 전달할 수 있다. read-buffer 명령을 수행한 결과는 호스트 rBuffer를 통해서 FX3 칩내 FIFO에 저장되며, 검증용 응용 프로그램에서 FX3 모듈에서 데이터를 읽는 C++ DLL API 함수를 통해서 FX3 모듈내 FIFO의 내용을 읽어서 C# 응용 프로그램의 출력 텍스트 박스에 표시하게 된다.

반면, write-burst 명령은 read-burst 명령에 비해, 사용자 설계 회로로 전달할 N개의 데이터가 추가되므로, 전달할 데이터가 N개인 경우 전체 명령어는 (N+2) 개의 32-비트 워드로 구성된다. 이러한 명령은 검증용 C# 소프트웨어에서 특정 이벤트 발생 시 이벤트 처리 함수에서 적절한 내용의 명령어를 만들어 FX3 칩으로 전달되며, FX3 모듈내 FIFO를 거쳐 호스트 인터페이스의 송신 버퍼(sBuffer)에 저장되며, SFIFO_empty 신호의 0(Low)의 값을 통해 마스터 제어기와 명령어 해독기가 수신된 명령어가 있음을 알 수 있게 된다.

3.3. 마스터 버스 제어기와 명령어 해독기

마스터 버스 제어기와 명령 해독기는 호스트 인터페이스의 송신 버퍼(sBuffer)에 담긴 명령을 해독하여 사용자 설계 회로에 적합한 주소, 제어신호와 데이터로 변환하는 역할을 수행한다. read-burst 명령의 경우는 주소를 통해 사용자 설계의 내부 레지스터 결과를 읽어서 호스트 인터페이스 모듈을 통해 FX3 칩으로 전달하는 역할을 한다. 마스터 제어 회로는 상태와 제어 동작을 함께 표현하는 Algorithmic State Machine(ASM) 도표 방식을 이용하여 설계하였다. 단, 일부 파이프라인 처리가 필요한 동작을 구현하기 위해, 항상 1개 상태가 활성화되는 기존 ASM 도표를 변형해서 여러 개의 상태가 활성화되는 변형된 방식을 사용하였다. 그리고 사용자 설계 회로에 1 주기 동안 타당한 제어 신호를 제공하기 위해, 조합회로를 거치지 않은 레지스터 출력을 제공하는 방식을 사용하였다[10]. 그림 5는 마스터 제어기와 명령어 해독기의 단순화 구조를 나타낸다.

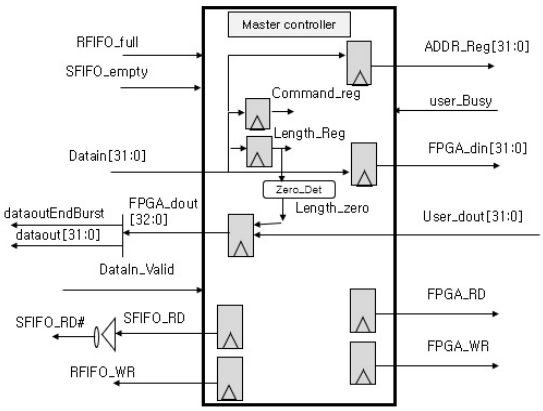

Fig. 5 Simplified hardware for master controller and command decoder module

그림 6은 write-burst 명령을 해독해서 필요한 제어 신호를 생성하는 ASM 도표를 나타낸다. 그림 6의 ASM 도표에서 SFIFO_empty 신호는 호스트 인터페이스의 송신 버퍼에 데이터가 존재 유무를 알려주는 상태 신호이며, 2개의 32-비트 워드를 받아서 명령어 종류와 데이터의 길이에 따라 적절한 버스트(burst) 전달 동작을 제어한다. user_Busy 신호는 사용자 회로 설계 부분이 데이터를 받을 수 있는 상태인지를 알려주는 상태 신호이다. 점선으로 표시한 파이프라인 처리 부분은 기존 ASM 도표에서 결정된 상태를 지연 시켜 사용자 설계 부분에 안정적인 FPFA_WR 신호와 주소, 데이터 등을 제공하기 위해 부분이다. 단, 이러한 파이프라인 처리되는 부분의 동작이 마칠 때까지 ASM 도표 동작이 종료하지 않도록 2개의 WR4와 WR5 상태를 거쳐서 초기 상태(WAIT) 상태로 이동한다.

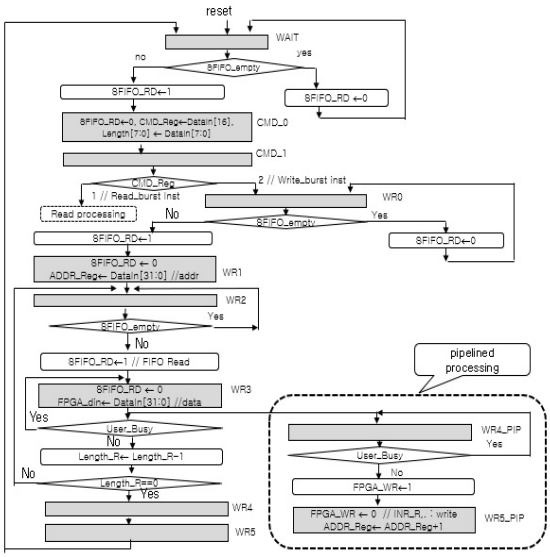

Fig. 6 ASM chart to handle write-burst instruction

Ⅳ. FPGA 구현과 동작 검증

3장에서 설계한 FX3 Slave FIFO 인터페이스의 올바른 동작을 검증하려면 사용자 설계 회로가 필요한 데, 사용자 설계 모듈을 포함한 전체 회로를 FPGA로 구현한 후 비주얼 C# 소프트웨어로 올바른 동작을 확인하였다.

4.1. 사용자 설계 회로

본 논문에서는 그림 4의 burst-read와 burst-write 동작의 burst 동작 확인에 초점을 맞추어, 입출력 데이터의 비트 수를 크게 하도록 3개의 64-비트 입력, 즉 192-비트의 입력을 받아서, mode 레지스터 값에 따라 3쌍의 64-비트 상수 덧셈과 상수 뺄셈으로 간단한 암호 및 복호 동작을 하는 회로를 구현하였다.

즉, mode 값이 0일 경우 상수 덧셈(암호), mode 값이 1일 경우 상수 뺄셈(복호)을 하는 회로이다. 그림 7은 사용자 설계 회로의 메모리 맵 인터페이스 구조의 입출력 인터페이스를 나타낸다.

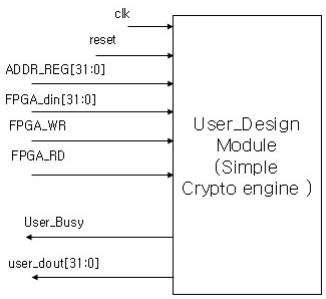

Fig. 7 I/O interface signals of user design circuit

표 1은 그림 7의 메모리 맵을 나타낸다. INP_Rx 와 OUT_Rx(x=0,1,2)는 64-비트 레지스터로 2개의 주소로 대응된다. 단, user_busy 상태 값은 read-burst로 읽을 수 있지만, 현재는 단순하게 상태 신호로 마스터 제어회로에 바로 제공될 수도 있도록 설계하였다.

Table. 1 Memory map for user design circuit

instruction	address (3-bit)	registers
write-burst	000, 001	INP_R0
	010, 011	INP_R1
	100, 101	INP_R2
	110	mode
	111	Start
read-burst	000, 001	OUT_R0
	010, 011	OUT_R1
	100, 101	OUT_R2
	111	User_Busy

4.2. 검증용 C# 소프트웨어 설계

본 연구에서는 검증을 위해 Visual Studio 2019 환경에서 C# 프로그램을 사용하였다. 그런데 FPGA 보드 개발 업체에서 제공한 FX3 칩을 제어하는 라이브러리가 .NET 환경에서 바로 사용할 수 없는 C++ DLL로 구성되어 있었다. NET 환경과 C++ 언어는 객체의 가비지 콜렉션 처리와 데이터 형 등에서 차이가 존재한다. 데이터의 형 차이를 고려하여 C++ DLL 라이브러리를 C#언어에서 호출해서 사용할 수 있는 대표적인 방안으로 Platform Invoke(P/Invoke) 기술[11,12] 이다. 이를 위해 C# 코드에 System.RunTime.InteropServices라는 네임스페이스를 포함하였으며, C++ DLL 라이브러리 함수를 C# 함수로 대응시키는 그림 8의 P/Invoke 선언을 포함하였다.

```
.....................
[DllImport("kernel32.dll")]
    extern static short QueryPerformanceCounter(ref long
x);
[DllImport("ZestSc3.dll")]
    public static extern ZESTSC3_STATUS
ZestSC3LoadFile(String FileName, ref IntPtr Image);
...............................
```

Fig. 8 P/Invoke declaration for C++ DLL library

4.3. FPGA 구현과 검증

본 논문에서 설계한 하드웨어는 Verilog HDL을 사용하여 기술하였으며, Xilinx Vivado 소프트웨어[13] 내 Xsim 소프트웨어로 검증하고 FPGA로 구현하였다. 그림 9는 마스터 제어기에서 호스트 인터페이스 모듈에서 write-burst 명령을 받아서, 사용자 설계 모듈로 데이터를 버스트 모드로 전송하는 동작에 대한 XSim 시뮬레이션 파형을 나타낸다. Xsim 검증 후 설계된 회로는 Xlinx Artix FPGA(XC7A35T-1C5G3241)로 구현한 후, 비주얼 C#과 C++로 구현한 검증 소프트웨어 환경에서 전체 시스템 동작을 검증하였다. FX3 칩에 제공되는 클록 주파수는 클록 발생기 내부 PLL에서 만든 100 MHz를 사용하고, 사용자 설계 부분은 각각 50 MHz와 25 MHz로 설정하여 올바른 동작이 이루어짐을 확인하였다. 그림 10은 Orange Tree 사의 FPGA 보드를 활용하여 C# 프로그램으로 사용자 설계 모듈의 복호 동작을 검증한 모습을 나타낸다. GUI 프로그램의 내용은 초기에 입력한 3개의 64-비트 임의의 값으로 설정한 후 암호 및 복호 버튼을 차례로 클릭할 경우 암호와 복호 동작이 순차적으로 수행되어, 복호 동작 결과가 원래 입력과 동일함을 나타낸다. 표 2는 USB 3 인터페이스를 위해 설계한 slave FIFO 모듈과 검증 소프트웨어 환경의 특성을 나타낸다. 설계된 slave FIFO 모듈은 2개의 다른 클록 주파수를 사용하는 시스템의 동기화 문제를 해결하는 호스트 인터페이스 모듈과 2가지 형식의 명령을 처리하는 마스터 버스 제어기와 명령어 해독기가 존재하는 모듈화 구조를 갖고 있어서, 메모리 맵 구조의 다양한 사용자 설계 회로에 적용 가능하다. 설계된 호스트 인터페이스 모듈내 송수신 FIFO가 각각 16개의 32-비트 워드 구조의 FIFO를 사용하는데, 사용자 설계 회로가 연속적인 데이터 수신이 필요하지 않는 경우, 용도에 맞게 FIFO 크기를 조정할 수 있다. 또한 slave FIFO 인터페이스는 연속적인 데이터가 송수신 가능한 스트리밍 방식으로 동작하므로, 마스터 버스 제어기와 명령어 해독기 설계를 변경할 경우 디지털 오실로스코프 등의 계측기 시스템에도 적용이 가능하다.

Table. 2 Characteristics of FPGA verification system

I/O interface	USB 3
USB bridge chip	EZ-USB FX3
slave FIFO slave module	Host interface module + Master Controller and command decoder
User design	hardware module with memory-mapped I/O interface
FPGA Chip	Xilinx Artix FPGA (XC7A35T-1C5G3241)
communication protocol	streaming
number of defined Instructions	2 (read-burst, write-burst)

clock frequency of FX3 chip	100 MHz
clock frequency of user design	variable (different from clock frequency of FX3 chip)
data marshalling method between C/C++ and C#	Platform Invoke
verification GUI software	C#

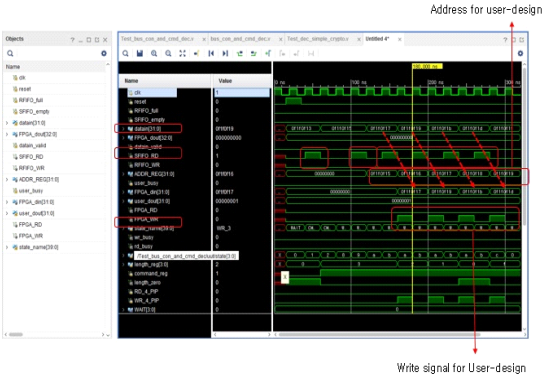

Fig. 9 Xsim simulation waveform for write-burst instruction

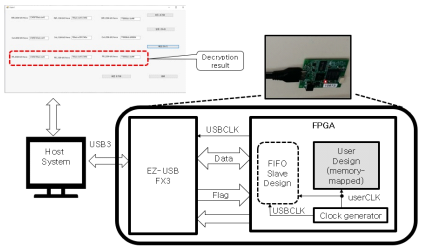

Fig. 10 FPGA verification result for user design circuit

V. 결 론

본 논문에서는 USB 3 인터페이스가 가능한 FPGA를 설계하기 위해서, USB 3 버스에 브릿지 역할을 하는 Cypress FX3 칩에 연결되는 FX3 FIFO Slave 인터페이스 모듈을 Verilog HDL로 설계하고, 메모리 맵 구조의 입출력 인터페이스를 사용하는 사용자 설계 회로를 연결하여 전체 검증 시스템의 올바른 동작을 확인하였다. 전체 시스템 검증을 위해 FX3 칩과 Xilinx Artix FPGA (XC7A35T-1C5G3241)를 포함하는 ZestSc3 보드를 사용하였으며, C++ DLL 라이브러리를 P/Invoke 기술로 호출하여 사용하는 C# 소프트웨어를 개발하여 GUI 환경에서 하드웨어와 소프트웨어 통합검증에 사용 하였다. 설계된 FX3 slave FIFO 인터페이스 모듈은 FX3 모듈과 다른 클록 주파수를 사용하는 메모리 맵 인터페이스 구조의 사용자 설계 회로를 지원하고, 모듈화된 구조를 가지고 있어서 FX3 칩을 활용하는 USB 3 버스용 FPGA 설계와 검증에 활용할 수 있다. 또한, FTDI사의 FT601 등의 USB 3 브릿지 칩을 사용하는 경우 FT601 칩과 연결되는 호스트 인터페이스의 제어 회로만 일부 변경할 경우 적용이 가능하다. 본 논문에서는 메모리 맵 인터페이스 구조를 갖는 사용자 설계 회로에 맞는 slave FIFO 회로에 초점을 맞추었는데, 내부 구조가 스트리밍 방식의 입출력을 지원하므로, 아날로그 데이터를 A/D 변환기를 통해 연속적으로 수신하는 디지털 오실로스코프 분야와 같은 계측기 시스템에도 약간의 회로 변형을 통해 응용 가능하다고 판단된다.

References

[1] K. Guo, S. Han, S. Yao, Y. Wang, Y. Xie, and H. Yang, “Software-Hardware Codesign for Efficient Neural Network Acceleration,” *IEEE Micro*, vol. 37, no. 2, pp. 18-25, Apr. 2017.

[2] W. L. Cho and K. Y. Shin, “2,048 bits RSA Public-key Cryptography Processor Based on 32-bit Montgomery Modular Multiplier,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 21, no. 8, pp.1471-1479, Aug. 2017.

[3] G. M, Sung. H. K. Wang, and J. H. Lin, “Serial Interface Engine ASIC with USB Physical Transceiver based on FPGA Development Board,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics(SMC)*, Banff: CA, pp. 410-413, Oct. 2017.

[4] FTDI Ltd., FT600-FT601Q IC Datasheet(USB 3.0 to FIFO Bridge), [Internet], Avaiable:https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT600Q-FT601Q%20IC%20Datasheet.pdf.

[5] J. Hyde, *SuperSpeed Device Design By Example*, 2nd ed, Lexington, KY: CreateSpace, 2016.

[6] Numato Ltd., USB 3.0-A Cost Effective High Bandwidth Solution for FPGA Host Interface, [Internet]. Available: https://numato.com/help/wp-content/uploads/2018/06/USB3.0-FPGAHostCommunication.pdf.

[7] Orange Tree Technologies Ltd. Datasheet, *ZestSC3 User*

Guide, 2017. [Internet]. Available: https://www.orangetreetech.com/
[8] R. S. Krishna, AN65974-Design with the EZ-USB FX3™ Slave FIFO Interface, May, 2020, [Internet]. Available: https://korea.cypress.com/documentation/application-notes/an65974-designing-ez-usb-fx3-slave-fifo-interface.
[9] C. E. Cummings, Simulation and Synthesis Techniques for Asynchnonous FIFO Design [Internet]. Available: http://discourse-production.oss-cn-shanghai.aliyuncs.com/original/3X/4/7/47fa3a7c83edfeea24029df12d0ae32fd9ed7610.pdf.
[10] R. A. Johnson, Registered-Output FSMs synchronize outputs to state transitions [Internet]. Available:http://www.mrc.uidaho.edu/mrc/people/jff/digital/fsm_timing.html.
[11] M. Borg, C++/C# interoperability, Sep. 2017. [Internet]. Available : https://mark-borg.github.io/blog/2017/interop/.
[12] J. AllWork, *Visual Studio C# 2010 Programming and PC Interfacing*, 1st ed, Elektor Pub. ch. 21, pp. 271-281, 2010.
[13] Y. J. Woo, *Verilog HDL Design using Vivado*, Inipro pub, 2017.

최병윤(Byeong-Yoon Choi)
1985년 2월 : 연세대학교 공학사
1992년 8월 : 연세대학교 공학 박사
1993년 3월~현재 : 동의대학교 컴퓨터공학과 교수
※ 관심분야 : SoC 설계, 임베디드 시스템