

Mobile Display Digital Interface 표준용 영상 데이터 전송기 설계

Design of Image Data Transmitter for Mobile Display Digital Interface

Ho-Kyung Lee*, Tae-Ho Kim**, Jin-Ku Kang***

이 호 경*, 김 태 호**, 강 진 구***

Abstract

This paper describes the design of image data transmitter for portable media device for Mobile Display Digital Interface standard. The transmitter uses connection lines to 6 to transmit QVGA image data. In this paper, the transmitter is using only 9 packets for image processing and a state-machine based design is adapted for packet processing. The design was verified using FPGA Xilinx virtex4-LX60. Data rate of the transmitter is 363Mbps with six connection lines. The transferring capability is 30 frame of 24bit RGB 500,000 pixel image data per second.

요 약

본 논문은 MDDI(Mobile Display Digital Interface) 표준을 이용하여 휴대용 디지털 미디어 기기에서 사용 가능한 영상 데이터 전송 시스템을 구현하였다. 설계된 영상 데이터 전송 시스템은 QVGA급 영상을 전송하기 위해 필요한 연결선 수 6개를 사용한다. 본 논문에서는, MDDI의 영상관련 9개의 패킷을 사용하였고, 패킷프로세서는 유한상태머신 기반의 설계로 이루어졌다. Xilinx 사의 FPGA virtex4-LX60을 이용하여 제작 및 검증을 수행하였다. 설계된 영상 데이터 전송 시스템은 6개의 연결선 수로 363Mbps 데이터 전송 대역폭을 갖는다. 이는 24비트 RGB 50만 화소의 영상 데이터를 초당 30 프레임까지 전송할 수 있는 성능이다.

Key words : MDDI(Mobile Display Digital Interface), transmitter, portable media device

1. 서론

최근 휴대용 전자 장치는 시간이 지날수록 고용량 메모리, 무선 네트워크 통신, 고성능 프로세서 및 여러 종류의 입력 장치가 하나의 휴대용 장치 안으로 탑재되어 다 기능화 되어가고 있는 반면, 크기와 전력 소비는 점차 줄어들고 있다. 특히, 디스플레이 장치의 경우, 고화질의 이미지 및 동영상 등 영상 데이터의 전송량이 점차 증가하는 추세이다. 이러한 영상 데이터를 병렬 데이터 전송 방식으로 처리할 경우, 데이터

전송선 수 혹은 송·수신단의 동작 속도를 증가 시켜야 한다. 데이터 전송선 수를 증가 시키는 것은 장치 간의 연결선 수가 증가를 의미하며, 이는 휴대용 장치의 크기를 크게 만드는 요인이 된다. 송수신단의 동작 속도를 높이기 위해서는 연결선 간에 발생하는 상호 간섭(EMI)과 영상 데이터 입력단의 스큐 발생으로 인한 데이터 전달의 신뢰성 저하가 문제점으로 지적되고 있다. 이러한 문제점들로 인해 병렬 데이터 전송 방식은 직렬 데이터 전송 방식으로 교체되고 있는 실정이다.

MDDI(Mobile display Digital Interface)는 디지털 패킷 데이터 연결(Digital packet data link)을 이용하여 가까운 거리에 있는 영상장치간의 이미지 데이터 전송을 저비용, 저전력으로 구현할 수 있고, 각 신호선의 다양한 데이터 전송률, 영상 기술의 독립적인 특성, 다양한 미디어 데이터의 지원은 시스템 구현에 있어 유동적인 설계를 가능하게 한다. 또한 기존 병렬 인터페이스에 소요되는 20~40개의 연결선 수를 6개(차동 데이터 신호, 차동 스트로브 신호, 전원라인)로 감소시킨다. 감소된 데이터 전송선 수는 다수의 PCB 기판을 이용하는 휴대용 전자 장치의 크기를 작게 하는데 많은 이점을 갖게 한다[1].

* (주) 실리콘웍스 (Silicon Works Co., Ltd.)

** 인하대학교 전자공학부

(School of Electronics Engineering, Inha University)

★ 교신저자 (Corresponding author)

※ 감사의 글 (Acknowledgment)

본 연구는 지식경제부의 시스템집적반도체기반기술개발사업(System IC 2010), 전략기술인력양성사업, ETRI의 IT-SoC 핵심설계인력양성사업에 의해 지원되었음. 반도체 설계교육센터(IDEC)에서 제공된 설계 툴을 지원받아 수행되었음.

接受日:2009年 12月 5日, 修正完了日: 2009年 12月 27日

본 논문에서는 직렬 방식으로 데이터를 전달하는 영상 데이터 전송기를 MDDI 표준의 42개 패킷 중에 영상관련 9개의 패킷만 사용하였다. 패킷프로세서는 전용프로세서를 사용하지 않는 순수하게 유한상태머신 기반의 설계방법을 제안하고 검증한다.

II. 구조

설계된 영상 데이터 전송기는 영상 데이터 전송을 주로 담당하는 호스트와 전송된 영상 데이터를 수신하는 클라이언트로 구성된다. 호스트와 클라이언트는 1쌍의 데이터 라인과 1쌍의 동기 라인이 차동방식으로 연결된다. 그림 1은 이를 도시한 것이며, 클라이언트가 호스트로부터 전원을 공급받지 않는 경우, 전원선(Host_Pwr, Host_Gnd)은 생략할 수 있다. 특히, 데이터 전송선은 호스트와 클라이언트 상호간 반이중 방식으로 통신을 하게 된다.

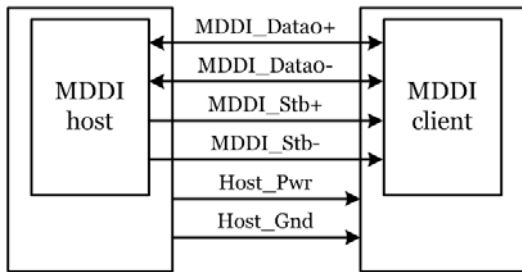


Fig. 1. Physical connection of host and client
그림 1. 호스트와 클라이언트의 연결

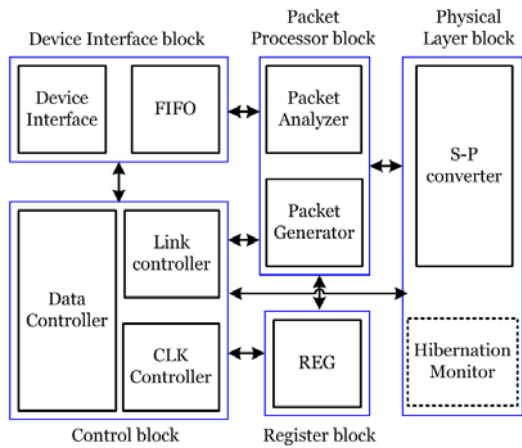


Fig. 2. Block diagram of host
그림 2. 호스트 블록 다이어그램

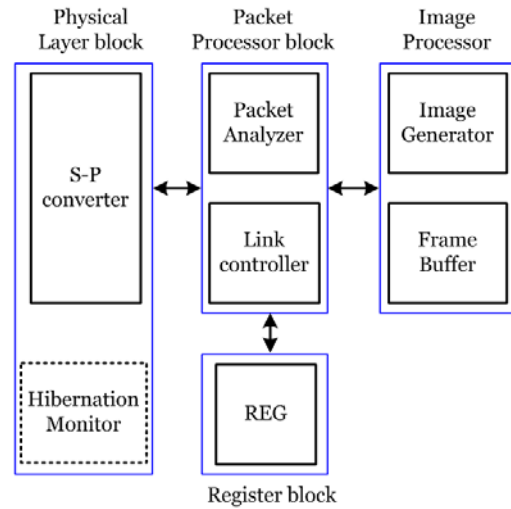


Fig. 3. Block diagram of client
그림 3. 클라이언트 블록 다이어그램

그림 2와 그림 3은 각각 호스트와 클라이언트의 블록 다이어그램을 보여주고 있다. 호스트와 클라이언트의 구조는 대칭적인 구조를 갖도록 설계하였다. 다만, 호스트는 주로 데이터의 송신 및 클라이언트와의 통신 상태를 제어하기 때문에 클라이언트에 비해 좀 더 복잡한 구조를 갖는다.

그림 2와 그림 3에서 보는 것과 같이 전송 데이터의 패킷화 및 복원을 담당하는 패킷 처리기(Packet Processor)를 중심으로 송·수신 데이터를 직렬화 혹은 병렬화 시키는 직·병렬 변환기(S-P Converter), 호스트와 클라이언트의 내부 스펙의 정보화 상태를 저장하는 레지스터 블록(REG) 그리고 영상 데이터 정보를 받아오는 디바이스 인터페이스 블록 (Device interface block, 호스트), 수신된 패킷 데이터로부터 영상 데이터를 추출하여 프레임 단위로 저장하는 영상 처리기(Image Processor, 클라이언트)로 구성되어 있다.

호스트와 클라이언트의 구조적인 차이점은 크게 제어부, 패킷 처리기 그리고, 이미지 데이터의 입출력 부분에서 차이가 난다. 이는 호스트는 주로 패킷 데이터를 생성하여 전송하는 역할을 하고, 클라이언트는 전송된 패킷 데이터를 분석하여 원 데이터를 복원하는 역할의 차이 때문이다. 일반적으로 패킷 데이터로부터 원본 데이터를 복원하는 것보다는 원본 데이터로부터 패킷 데이터를 생성하는 방법이 더 복잡하다. 그렇기 때문에, 호스트에서는 패킷 생성기와 분석기를 하나의 패킷 처리기 안에 설계했으며, 이를 통제하는 제어부를 분리시켰다. 클라이언트에서는 패킷 생성의 역할이 매우 작은 부분으로 패킷 분석기에 이를 포함시켰으며, 제어부 또한 패킷 처리기 안에 포함시켰다.

III 회로 설계

MDDI 표준에서는 미디어 데이터의 전송을 위해 42개의 패킷을 제공한다. 본 논문에서는, 영상 데이터 전송을 위해 9개의 패킷만 사용한다. 표 1은 영상 데

이더 전송에 사용한 패킷들을 나타낸 것이다. 9개의 패킷만으로도 영상데이터의 송수신이 가능하며 실장 테스트에서 검증되었다.

영상 데이터 전송기의 송·수신부의 연결 상태를 안정적으로 유지하기 위해서 MDDI 표준은 CRC-16 (Cyclic Redundancy Check)을 이용한다[2]. 그림 4는 이 CRC를 체크하여 링크의 연결 상태를 관리하는 상태 다이어그램을 보여주고 있다. 상태1은 시스템이 구동된 초기의 상태이며, 송신단과 수신단의 동기가 이루어지지 않은 상태이다. 이 상태에서는 클라이언트는 송신부에서 송신되어지는 데이터를 모니터링하게 된다. 클라이언트에서 동기를 맞추기 위한 서브-프레임 헤더 패킷(Sub-frame header packet)을 수신하게 되면, CRC 에러를 체크하지 않고 상태2로 천이한다. 상태2에서부터는 수신되어지는 각 패킷의 CRC 에러를 체크한다. CRC 에러의 발생 유무에 따라 다음 상태로의 천이를 결정하게 된다. CRC 에러가 발생하게 되는 경우, 상태1로 천이하여 호스트와의 동기를 다시 맞추게 되고, CRC 에러가 발생하지 않은 경우, 상태3으로 천이하여 데이터 수신을 계속 수행하게 된다. 상태3 이상의 상태는 동기가 확립된 상태(IN-SYNC STATES)이며, 이후 CRC 에러가 연속 3회 이상 발생하게 되면 다시 호스트와의 동기를 맞추는 상태(AQUIRING-SYNC STATES)로 천이하게 된다.

Table 1. Considering packets for image data transmitter
표 1. 영상 데이터 전송기 설계 시 고려된 패킷

Packet name	Packet type	Valid in forward	Valid in reverse
Sub-frame header packet	71	x	
Filler packet	72	x	
Reverse link encapsulation packet	73	x	
Link shutdown packet	74	x	
Round-Trip delay measurement packet	143		x
Video stream packet	136	x	
Client capability packet	137		x
Client request and status packet	18	x	
Register access packet	146	x	x

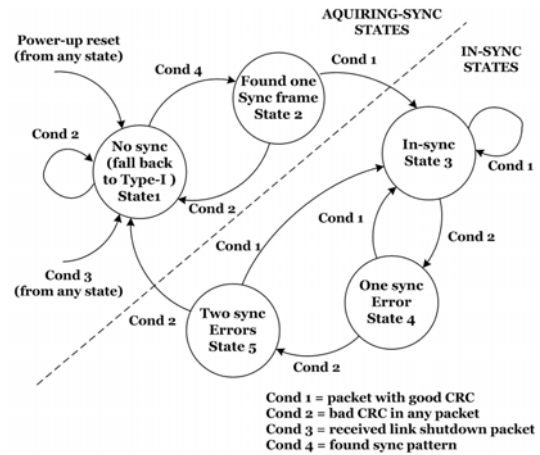


Fig. 4. State diagram for link synchronization and its stability

그림 4. 링크 동기 및 안정성을 위한 상태도

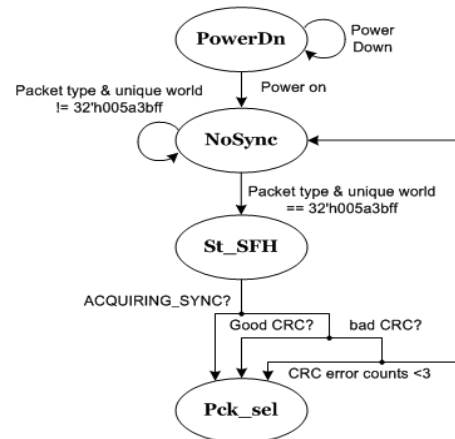


Fig. 5. S State diagram of client packet processor

그림 5. 링 클라이언트 패킷 처리기의 상태도

그림 5는 설계된 클라이언트부의 패킷 처리기의 무어머신 상태도를 나타낸 것이다. 시스템 초기 기동시, 클라이언트는 수신되는 데이터로부터 서브-프레임 헤더 패킷의 데이터인 005A3BFF(hex) 값을 모니터링 한다. 동기를 나타내는 4바이트 데이터를 수신하게 되면, 클라이언트의 패킷 처리기는 서브-프레임 헤더 패킷을 처리한다. 이후 상태의 천이는 현재 상태가 ACQUIRING-SYNC 상태인지 IN_SYNC 상태인지에 따라 나뉘는데 두 상태에 대한 상태 값은 상태로서가 아닌 독립된 레지스터에 저장하여 이후의 상태 천이에 참조하는 구조로 설계 하였다. 또한 CRC 에러의 수도 레지스터에 따로 저장하여, 패킷을 처리하는 Pck_sel 상태로 천이할 것인지에 대해 판단할 때 사용된다.

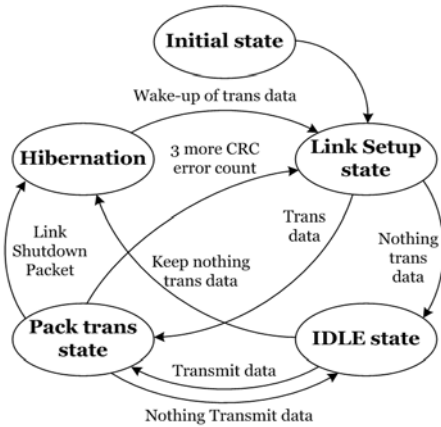


Fig. 6. State diagram of host link controller
 그림 6. 호스트 링크 제어기의 상태도

호스트와 클라이언트의 링크 상태를 제어하는 호스트의 링크 제어부의 상태를 그림 6에 나타내었다. 호스트의 초기화(Initial state) 이후 호스트는 클라이언트와의 연결을 확립하게 된다. 이때 호스트는 서브-프레임 패킷과 리버스 링크 인캡슐레이션 패킷을 클라이언트로 전송하여 클라이언트의 기능들에 대한 데이터를 수신한 후, 클라이언트에 적합한 링크 환경을 설정한다. 링크 설정이 완료하게 되면, 호스트는 외부 장치로부터 전송될 데이터의 유무를 확인하고, 전송할 영상 데이터가 존재하면 Pack trans 상태로 넘어가 클라이언트에게 영상 데이터를 전달한다. 만약, 외부 장치로부터 전달되는 영상 데이터가 없는 경우, IDLE 상태에 진입하여 외부 영상 장치로부터 데이터의 입력을 모니터링 한다. IDLE 상태가 일정기간 지속되게 되면, 호스트는 클라이언트에 링크 셋다운 패킷을 전달하고 링크 상태 및 호스트의 상태를 하이베이션 상태로 전환하고, 외부 영상 장치로부터 전송될 데이터를 수신하게 되면 호스트는 pack trans 상태로 넘어가 수신된 영상 데이터를 전달하게 된다. 영상 데이터 전달 상태에서 호스트는 서브-프레임 헤더 패킷, 비디오 스트림 패킷, 리버스 링크 인캡슐레이션 패킷을 반복하여 외부 영상 장치로 받은 영상 데이터를 전송한다. 이 때 리버스 링크 인캡슐레이션 패킷에서 받은 데이터 중 클라이언트가 수신한 데이터의 CRC error가 3 이상의 값을 갖게 되면 호스트는 링크 상태를 재설정하기 위해 링크 설정 상태로 천이하게 된다. 영상 장치로부터 수신한 데이터를 모두 전송한 경우, IDLE 상태로 천이하게 되고 클라이언트로부터 링크 셋다운 패킷을 수신할 경우, 하이베이션 상태로 천이한다.

패킷을 생성하는 방법은 크게 호스트의 컨트롤러 블록으로부터 기동 신호를 이용하여 생성하려는 패킷 생성기를 동작시킨다. 이때 각 패킷 생성기 블록의 출력은 맥스와 기동신호를 이용하여 선택적으로 출력할 수 있다. 그림 7은 패킷생성기를 구현한 회로를 보았다. 그림 8의 비디오 스트림 패킷 예로, 고정 크기를 갖고 있는 필드들은 그림 7의 패킷 생성 카운터 2(Packet generation counter2-PGC2)의 값을 증가시

킨다. 즉, VSPG_en_H가 1이 되면, PGC2 카운터의 값이 매 클럭마다 1씩 증가한다. PGC2 카운터의 값이 변화할 때마다 그림 7의 패킷 필드 상단에 숫자에 해당하는 필드의 값이 맥스로 전달된다. 패킷 파라미터의 값은 레지스터 블록으로부터 PAR_in을 통해 해당 파라미터 값들을 전달 받는다. 레지스터 블록은 듀얼포트 램(Dual-port RAM)을 이용하였다. 레지스터 블록으로부터 받을 데이터의 주소는 PGC2와 각

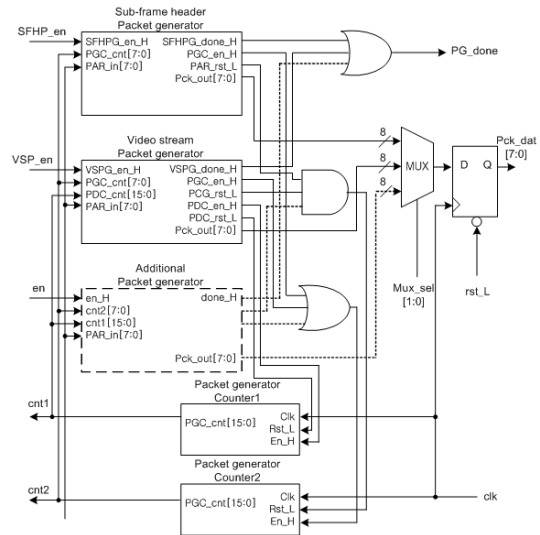


Fig. 7. Block diagram of packet generator in host
 그림 7. 호스트 패킷 생성기의 블록 다이어그램

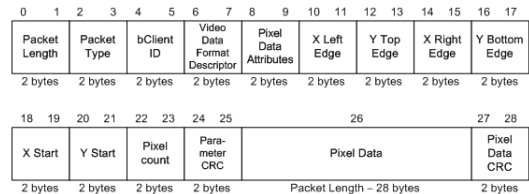


Fig. 8. Video stream packet structure
 그림 8. 비디오 스트림 패킷의 구조

Table 2. Summary of image data transmitter performance
 표 2. 영상 데이터 전송기 성능 요약

	Client	Host
Total Number slice Register	115	778
Total Number of 4 input LUTs	913	3014
Total equivalent gate count for design	74084	94108
Maximum Data rate(Mbps)	380Mbps	363Mbps
Logic delay(ns)	2.119	1.617
Route delay(ns)	3.163	1.138
Total delay(ns)	5.282	2.775

패킷 생성블록의 기동신호를 이용하여 생성한다. 패킷

필드 중에서는 호스트 혹은 외부 상태에 따라 크기가 변화하는 필드가 존재한다. 그림 8의 픽셀 데이터 필드가 이에 해당한다. 이때는 픽셀 데이터 필드 값이 모두 전달될 때까지 PGC2의 카운트 값은 26으로 유지되고, PGC1의 카운트 값을 증가시키면서, 전송하고자 하는 픽셀 데이터의 양을 체크하게 된다. 픽셀 데이터를 모두 전송하면, PGC1 카운트의 값을 초기화시키고, PGC2 카운트의 값을 증가시켜 픽셀 데이터 필드 이후의 필드 값을 전송하게 된다. 마지막 패킷 필드를 전달할 때, 해당 패킷 생성기는 패킷 생성완료 를 알리는 PG_done_H 신호를 한 사이클 동안 '1' 값으로 출력하여 컨트롤 블록에 패킷 생성의 완료 를 알린다.

MDDI 표준을 이용한 영상 데이터 전송기는 verilog-HDL을 이용하여 Xilinx 사의 FPGA Virtex4-LX60에 구현하였다[3]-[7]. 표 2는 영상 데이터 전송기 구현 시 소요된 FPGA의 소자의 수와 설계된 영상 데이터 전송기의 성능을 요약한 것이다. 클라이언트는 호스트로부터 받은 클럭의 상승, 하강 에지를 모두 사용하기 때문에 호스트의 최대 동작 주파수의 반으로 동작해야 한다. 설계된 데이터 전송기의 데이터 전송 대역폭은 약 363Mbps까지 가능하다는 것을 확인할 수 있다. 참고논문 [9]의 경우 외부 CPU와 연결된 MDDI 호스트 시스템 설계시 최고 데이터 속도는 50Mbps였다. 본 논문은 패킷처리의 유한상태머신기반의 설계로 게이트 효율성을 높여 속도를 높일 수 있었다.

IV 시뮬레이션 및 측정결과

설계된 영상 데이터 전송기의 모의실험 구성은 그림 9와 같다[8]. 이미지 파일을 이용하여 호스트의 제어 신호와 영상 데이터를 테스트 벡터로 입력하고 클라이언트에서 수신한 영상 데이터를 텍스트 파일로 출력하고 이를 다시 영상 이미지로 변환하여 입력한 영상과 출력한 영상을 비교하였다

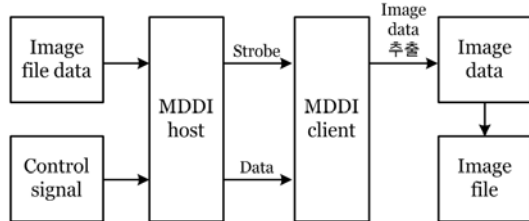


Fig. 9. Verification method of image data transceiver
그림 9. 이미지 데이터 전송기의 검증 방법

보드의 검증은 먼저 호스트에서 클라이언트로 한 프레임 영상을 전송하고 검증하였다. 한 프레임의 영상 전송시 영상 해상도에 맞는 서브-프레임의 숫자가 전송됨을 확인할 수 있었다. QVGA급 영상 한 프레임 을 보내기 위해서 14개의 서브-프레임이 전송되어 졌다.

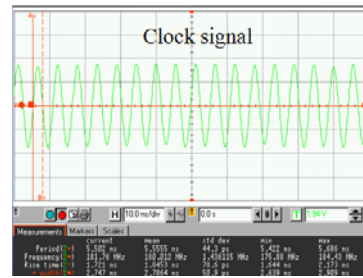


Fig. 10. Measured clock signal
그림 10. 출력 클럭 측정 파형

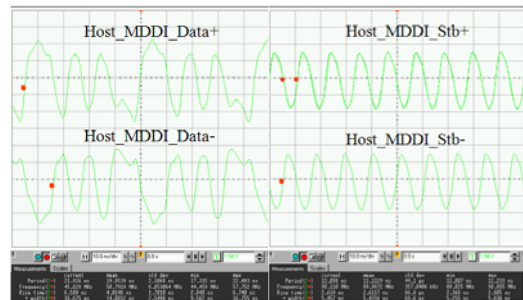


Fig. 11. Measured Data and Strobe waveform
그림 11. 데이터 와 스트로브 신호 출력 측정파형

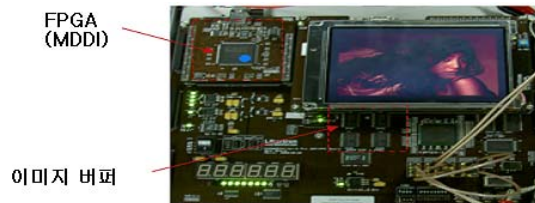


Fig. 12. Test Board Implementation for Image Transfer
그림 12. 실장 이미지 전송 테스트 보드

그림 10과 그림 11은 보드상에서 출력되는 주요 신호들을 보였다. 그림 10의 클럭 파형은 동작속도가 약 180MHz 근처에서 동작하여 그림 11의 363Mbps 데이터와 대응하는 스트로브신호를 생성할 수 있다.

그림 12는 고정된 영상 이미지를 FPGA로 구현한 MDDI를 통해 송신하여 디스플레이 장치에 보여주는 테스트 보드를 보였다. 원래 저장된 영상이 보여짐을 확인 하였다.

V 결론

본 논문에서는 Mobile Display Digital Interface 표준을 이용하여 영상 데이터를 전송하는 시스템을 설계하였다. 설계된 영상 데이터 전송 시스템은 Xilinx 사의 Virtex4 LX60 칩을 타겟으로 구현되었으며, 최대 동작속도는 호스트의 경우 약 363Mbps, 클라이언트는 약 380Mbps를 얻을 수 있었다. 이는 4개(6개)의 연결선을 이용하여 SVGA급 영상이 전송 가능하며, 24비트 RGB 30프레임 영상을 기준으로 약 50만 화소의 영상 데이터 전송이 가능한 성능이다. 이때, 소비 는 되는 칩 면적은 게이트 카운트 환산 시 호스트는 약 9만 5천, 클라이언트는 약 7만 5천개의 게이트가

사용되었다. ASIC을 이용하여 영상 데이터 전송 시스템을 구현할 경우, MDDI 표준에서 지원하는 400Mbps의 데이터 전송 대역폭을 충분히 구현할 수 있을 것으로 본다.

참고문헌

- [1] VESA, *Mobile Display Digital Interface Standard Version 1*, 2004
- [2] G. Albertengo and R. Sisto, "Parallel CRC Generation," *IEEE Micro*, vol.10, no.5, pp.64-71, Oct. 1990
- [3] Zainalabedin Navabi, *Verilog Digital System design : Register transfer level, synthesis, testbench, and verification*, second edition, McGraw-Hill, 2005
- [4] Michael D. Ciletti, *Advanced digital design with the verilog HDL*, Prentice Hall, 2002
- [5] Douglas J. Smith, *HDL chip design: A practical Guide for designing, synthesizing & simulating ASIC & FPGA using VHDL or verilog*, Doone Publications, 1998
- [6] Samir Palnitkar, *Verilog HDL*, Second Edition, Prentice Hall, 2003
- [7] Ze-Nian Li, Mark S. Drew, *Fundamentals of multimedia*, Prentice Hall, 2007
- [8] R. Kakerow, "Low Power Design Methodologies for Mobile Communication," *IEEE International Conference on Computer Design*, pp.8-13, Sept. 2002.
- [9] J. Park, et al., "An MDDI-host Architecture with Low complexity for SoC Platforms," *IEEE Trans. on Consumer Electronics*, pp. 1668-1673, Nov. 2007

 저 자 소 개

이 호 경 (정회원)



2006년 : 인하대학교 전자공학과
졸업 (공학사)
2008년 : 인하대학교 전자공학과
(공학석사)
2008년 1월~현재: (주)실리콘웍스
재직중

<주관심분야> Digital System for high speed
interface

김 태 호 (학생회원)



2007년 : 인하대학교 전자공학과
졸업 (공학사)
2009년 : 인하대학교 일반대학원
전자공학과 (공학석사)
2009년 3월~현재 : 인하대학교
전자공학과 박사과정
<주관심분야> High Speed
CMOS Interface IC

강 진 구 (정회원)



1983년 : 서울대학교 (공학사)
1990년 : New Jersey Institute of
Technology 전기 및 컴퓨터공학
(공학석사)
1996년 : North Carolina State
University 전기 및 컴퓨터공학 (공
학박사)
1983년 ~ 1988년 : 삼성전자(반도체)
1996년 ~ 1997년 : 미국 INTEL Senior Design Engineer
1997년 3월~현재 : 인하대학교 전자공학부 교수
<주관심분야> 고속 CMOS 회로 설계, Mixed IC 설계,
PLL/DLL/CDR, High Speed Interface IC, Display IC