

# 디스플레이포트 인터페이스의 AUX 채널 설계

## A Design of DisplayPort AUX Channel

차성복\*, 윤광희\*, 김태호\*\*, 강진구\*\*\*  
 Seong-Bok Cha\*, Kwang-Hee Yoon\*, Tae-Ho Kim\*\*, Jin-Ku Kang\*\*\*

### Abstract

This paper presents an implementation of the DisplayPort AUX(Auxiliary) Channel. DisplayPort uses Main link, AUX Channel and Hot Plug Detect line to transfer the video & audio data. For isochronous transport service, source device converts to image and audio data which are to be transported through the Main Link and transports the restructured image and audio data to sink device. The AUX Channel provides link service and device service for discovering, initializing and maintaining the Main link. Hot Plug Detect line is used to confirm the connection between source device and sink device. The AUX Channel is implemented with 3315 LUTs(Look Up Table), 1466 Flip Flops and 168.782MHz max speed synthesized using Xilinx ISE 9.2i at SoC Master3.

### 요약

본 논문은 디스플레이포트 v1.1a 표준에 적합한 AUX(Auxiliary) 채널 구현에 대한 논문이다. 디스플레이포트는 영상 및 음성을 전달하기 하기 위해 메인 링크, AUX 채널, 핫 플러그 검출 라인을 사용한다. 등시적 전송 서비스를 제공하기 위해서 소스 디바이스는 메인 링크를 통해 전달될 영상 및 음성 신호를 특정 형태로 변환하여 재구성하고 싱크 디바이스로 전달한다. AUX 채널은 메인 링크를 구성하고 유지하기 위해 링크 서비스를 제공한다. 그리고 디스플레이 장치가 소스 디바이스에서 전송된 데이터를 정상적으로 나타낼 수 있는지 파악하기 위해 디바이스 서비스를 제공한다. 핫 플러그 검출 라인은 두 디바이스간의 연결을 확인하기 위해서 사용한다. 본 논문은 AUX 채널 구현을 목표로 설계하였으며 설계된 시스템은 SoC Master3를 이용하여 검증을 수행하였다. 합성 틀은 Xilinx ISE 9.2i를 사용하여 3315개의 LUTs와 1466개의 Flip Flops을 사용하였고 최대 168.782MHz 동작 속도의 결과를 얻었다.

*Key words : DisplayPort, AUX Channel, DPCD, EDID, Policy Maker*

## 1. 서론

\* (주) 실리콘웍스 (Silicon Works Co., Ltd.)

\*\* 인하대학교 전자공학부  
 (School of Electronics Engineering, Inha University)

★ 교신저자 (Corresponding author)

※ 감사의 글 (Acknowledgment)

This work is supported by the Ministry of Knowledge Economy (MKE) through System IC 2010, Human Resource Training Project for Strategic Technology from KIAT, and IT-SoC program from ETRI. Authors also thank IDEC program.

接受日:2010年 3月 8日, 修正完了日: 2010年 3月 26日

최근 수년간 고해상도 영상 기술의 발달에 맞추어 새로운 고속 디스플레이 인터페이스 표준이 계속 발표되어 왔다. 현재 시장은 Full HD 디스플레이 장치와 Blu-Ray, HD-DVD등의 차세대 미디어들이 등장하고 있다. 기존의 DVD의 경우 영상 전송량이 5~7Mb/sec인 반면에 Blu-Ray나 HD-DVD의 경우 40Mb/sec정도로 영상 전송량이 6배 이상 증가하였다. 즉, 이러한 고해상도 영상을 재생하기 위해서는 대용량 데이터 전송이 가능한 새로운 인터페이스 방식이 필요하게 되었다.

현재의 디스플레이에는 주로 세 가지 방식이 사용되고 있다. 노트북 내부에는 LCD패널로 LVDS(Low-Voltage Differential Signaling) 기술이 사용되고 있고, PC와 외부 모니터의 연결은 VGA아날로그 인터페이스와 DVI 기술이 사용되고 있다. TV와 PC를 연결할 경우 또는 다른 AV 기기와 연결할 경우 S-video, Component 등의 영상 케이블과 스테레오, 광신호 등의 음성 케이블을 별도로 연결해야 한다. 이러한 현재의 기술들은 설치상의 불편함이 존재할 뿐만 아니라 소스가 디지털 소스라 하더라도 component, S-video 등을 사용할 때 영상 정보의 훼손이 이루어지게 된다. DVI(Digital Visual Interface)를 사용하는데 있어서도 영상 정보만을 전달할 수 있어 별도의 케이블이 필요로 하게 된다. 또한 추후 사용을 위한 확장성을 갖추지 못한 한계를 가지고 있다 [1].

차세대 디지털 인터페이스로 HDMI(High Definition Multimedia Interface)와 디스플레이포트가 등장하고 있다. HDMI는 디지털 영상 소스 및 음성 소스를 전달하는 인터페이스 규격 중 하나로 기존 DVI 규격을 AV(Audio-Video) 전자제품용으로 변경한 것이다. 최대 10.2Gbps의 대역폭을 가지고 TMDS(Transition Minimized Differential Signaling) 프로토콜을 사용하며 영상 저작권을 보호하기 위해 HDCP(High-bandwidth Digital content Protection)를 지원한다. TMDS 프로토콜의 사용으로 라이선스 사용료를 지불해야 하며 프로토콜 변환 블록이 설계되어야 하는 단점이 있다.

디스플레이포트는 송신기로서 주로 PC의 그래픽카드 부분에 탑재되고 수신기로서 모니터에 도입되어 기존의 디스플레이 인터페이스를 대체할 기술이다. 디스플레이포트의 최대 대역폭은 10.8Gbps를 가지며 영상 저작권을 보호하기 위해 DPCP(DisplayPort Content protection)를 지원하고 마이크로패킷 아키텍처로 다중 스트림을 지원한다. 또한 1Mbps 대역폭의 양방향 AUX 채널(Auxiliary Channel)을 탑재하여 별도의 인터페이스 없이 영상과 음성을 동시에 전달할 수 있다. 그리고 라이선스를 제거하고 차후 새로운 디스플레이 기능이나 어플리케이션 수용을 위한 확장성에 주안을 두어 내부 및 외부 디스플레이 시그널 통합이 가능하다[2].

본 논문에서는 디스플레이포트의 비디오 데이터 전송 부분과 AUX채널 송수신기를 통합하여 AUX 채널의 신호에 따라 영상 데이터 전송이 조절 되는지 확인하였다.

## II. 디스플레이포트 인터페이스 개요

디스플레이포트는 VESA(Video Electronics Standards Association)에서 제안한 차세대 디지털 디스플레이

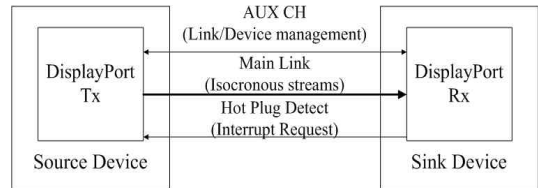


Fig. 1. DisplayPort Data Transport Channels

그림 1. 디스플레이포트 데이터 전송 채널

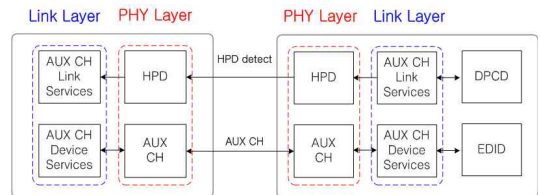


Fig. 2. Block Diagram of DisplayPort AUX CH

그림 2. 디스플레이포트 AUX CH 블록도

레이 인터페이스 표준이다. 컴퓨터와 모니터, 홈시어터 시스템 등을 연결하는데 사용되며 라이선스 사용료가 없는 최신 디지털 비디오/오디오 연결 표준이다.

그림 1에 디스플레이포트의 데이터 전송 채널 구성도를 나타낸다. 소스 디바이스와 싱크 디바이스 사이는 메인 링크, AUX 채널, HPD(Hot Plug Detect) 신호 라인으로 이루어져 있다. 메인 링크는 영상 및 음성 데이터를 소스 디바이스에서 싱크디바이스로 전송시키며 대역폭이 높고 호출 시간이 짧은 채널이다. AUX CH 블록도를 그림 2에 나타낸다. AUX 채널은 반이중 양방향(Half-duplex, bidirectional) 채널로 링크 관리 및 장치 제어에 사용된다. HPD 신호는 소스 디바이스와 싱크 디바이스의 연결 상태를 알리기 위해 사용된다. 메인 링크와 AUX 채널은 AC 차동쌍(differential pair)로 이루어져 있다. 메인 링크는 최대 4쌍(1,2,4 pair)로 되어 있고 각각은 270Mbytes/sec의 어플리케이션 대역폭을 지원하고 AUX 채널은 1Mbps의 대역폭을 갖는다.

디스플레이포트는 클럭 전송을 위한 별도의 쌍(pair)이 필요 없다. 2.7Gb/s 데이터에서 3m 거리를 연결하고 1.62Gb/s에서 최장 15m 케이블 커넥터를 연결하여 사용할 수 있다[3].

디스플레이포트는 소스 디바이스와 싱크 디바이스 간의 연결 상태 및 제어를 위해 AUX 채널을 사용한

다. 양방향 반이중 통신을 지원하고 차동 쌍으로 구성되어 있으며 데이터 전송의 신뢰성을 높이기 위하여 Manchester II 코딩 방식을 사용한다. 또한 지원되는 케이블 길이에 1Mbps의 데이터 속도를 제공한다. 마스터로서 소스 디바이스는 요청 신호를 초기화해야 하며 싱크 장치는 요청 신호에 따른 응답신호를 AUX 채널 문법에 맞게 전송하여야 한다. AUX 채널 문법은 다양한 보조 채널 서비스를 지원하기 위해 사용된다.

### III AUX 채널 설계

제안하는 AUX 채널 송수신기의 구조를 그림 3 및 그림 4에 각각 나타낸다. AUX 채널은 소스 디바이스에 Stream source, Link policy maker, Stream policy maker로 구성되어 있으며 싱크 디바이스에는 Link policy maker, Stream policy, EDID (Extended display identification data), DPCD로 구성되어 있다. Link policy maker는 링크를 초기화하고 구성하며 유지하기 위해 사용되며 Stream policy maker는 스트림 영상 및 음성 데이터 전송을 초기화하고 관리하기 위해 사용된다. DPCD는 현재 구성된 메인 링크의 성능 및 상태 등을 특정 메모리 번지에 저장하고 있으며 EDID는 싱크 디바이스의 성능을 저장하고 있다.

AUX 채널의 동작 제어는 소스 디바이스와 싱크 디바이스의 AUX 채널 상태 제어기 기반으로 설계한다. 메인 링크의 구성 및 유지를 위해 Link Policy Maker는 Link Training을 수행한다. Link Training은 4가지 상태로 구성된다. 메인 링크가 확립이 되지 않은 상태인 Main Link Disabled 상태, 소스 디바이스에서 전송된 데이터 심벌을 가지고 싱크 디바이스에서 클럭을 복원하기 위한 Clock Recovery Pattern 상태, 특정 심벌을 송신단에서 받아 Channel Equalization을 수행하기 위한 Channel Equalization Pattern 상태, 마지막으로 메인 링크가 확립 되었을 때 메인 링크에서 데이터를 전송하는 Normal Operation 상태로 나누어진다.

각각의 상태 변이는 DPCD의 해당 어드레스의 데이터 값에 의해서 결정되며 상태가 바뀌면 소스 디바이스와 싱크 디바이스가 AUX 채널을 통해 데이터를 주고받아 메인 링크를 재확립한다. Link Training의 클럭복원 패턴(Clock Recovery Pattern)상태에서는 소스 디바이스에서 Scrambling 되지 않은 8B/10B Encoder로 인코딩된 D10.2의 데이터 심벌을 반복적으로 전송한다. 소스디바이스에서는 반드시 Pre-emphasis가 해제되어 있어야 한다. 데이터를 보내고 소스 디바이스는 일정시간 동안 기다리고 싱크

디바이스는 물리층의 CDR(Clock Data Recovery -클럭 데이터 복원)이 위상고정(Lock) 될 경우 DPCD의 CR\_DONE비트를 설정하고, 만약 CDR이 일정시간 안에 Lock이 되지 않으면 소스 디바이스의 차동 전압

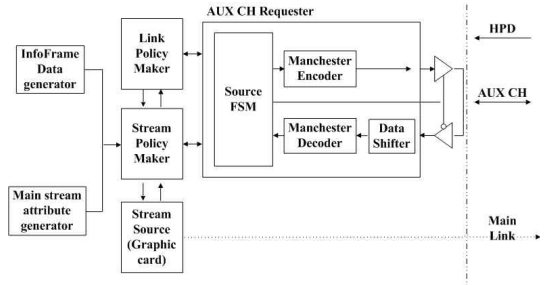


Fig. 3. The structure of the proposed transmitter for AUX Channel

그림 3. 제안하는 AUX 채널 송신기의 구조

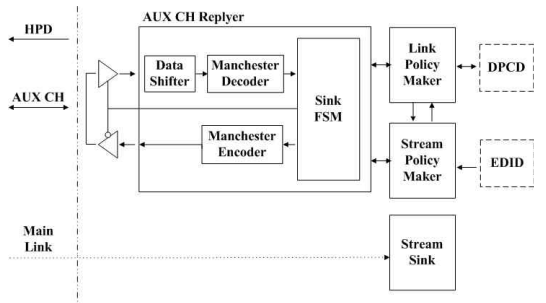


Fig. 4. The detail structure of the proposed receiver for AUX Channel

그림 4. 제안하는 AUX 채널 수신기의 구조

스윙을 증가시킨다. 이후에도 일정 시간 이상 CDR이 위상 고정 안 될 경우 데이터 전송 비트 속도를 낮추게 되고 이 과정 역시 실패하게 되면 Link Training은 중지되고 데이터 전송을 할 수 없게 된다.

성공적으로 클럭복원(Clock Recovery) 과정으로 이동하면 채널등화(Channel Equalization) 과정을 수행하는데 이 과정에서 차동 신호는 반드시 Pre-emphasis가 해제되어 있어야 한다. 소스 디바이스는 수신측에 있는 DPCD에 해당 어드레스의 해당 값을 쓰고 열 개의 반복적인 패턴을 송신한다. 이때 이 심벌은 Scrambling이 해제되어 있어야 한다. 싱크 디바이스는 송신측에서 보낸 데이터를 바탕으로 채널등화(Channel Equalization)과정을 성공적으로 수행했는지 판단한다. 성공적으로 Equalization 과정을 수행하면 DPCD에 판별한 데이터 값을 쓰게 되고 메인 링크가 구성되게 된다.

영상 및 음성 데이터를 디스플레이 장치에서 정상적으로 재생할 수 있는지 파악하기 위해 Stream Policy Maker는 전송할 스트림 데이터의 속성(attribute)와 디스플레이 장치의 성능 정보를 비교한다. 이를 위해서 Stream Policy Maker는 세 가지 정보를 필요로 한다. 첫째로 소스 디바이스에서 보내는 데이터의 타이밍 정보가 필요하고 두 번째로 음성 데이터의 정보가 필요하다. 마지막으로 디스플레이 장치의 성능 정보가 필요하다[4],[5].

소스 디바이스에는 그래픽 카드(stream source)의 영상 데이터 타이밍을 분석하는 Main Stream Attribute Generator를 설계하여 전송하고자 하는 데이터의 타이밍을 분석하고 그 정보를 소스 Stream Policy Maker에 전송한다. 음성 데이터는 소스 디바이스의 InfoFrame Data Generator에서 오디오 채널, 코딩 타입, 샘플링 수 등의 정보를 분석하여 소스 Stream Policy Maker에 전송한다[5]. 그리고 싱크 디바이스의 EDID(Extended Display Identification Data)에서 타이밍 정보를 담고 있는 어드레스를 읽어 디스플레이 장치의 정보를 파악한다. EDID는 제조사 이름, 제품 유형, EDID버전, 비디오 타이밍, 화면 크기, 휘도, 화소 등에 대한 정보를 담고 있다[6].

Stream Policy Maker는 Link Policy Maker로부터 HPD(Hot Plug Detect) 신호를 입력 받아 동작이 실행된다. 소스 디바이스의 Stream Policy Maker는 EDID 정보를 AUX 채널을 통해서 읽기 위해 I<sup>2</sup>C를 AUX 채널에 매핑하는 규정에 따라 정해진 문법을 사용한다. 그림 3에서 Stream Policy Maker 블록과 Stream Source 블록 그리고 그림 4에서 Stream Policy Maker 블록과 EDID는 타이밍을 늘여뜨린 Stretched I<sup>2</sup>C 프로토콜을 사용한다. Stretched I<sup>2</sup>C란 일반적 I<sup>2</sup>C 프로토콜과 같이 데이터 전송을 위해 SCL(Serial Clock Line)과 SDA(Serial Data Line)를 사용하며 싱크 장치의 I<sup>2</sup>C 마스터가 START, 명령과 주소를 전송하면 I<sup>2</sup>C 슬레이브는 응답신호로써 ACK를 전송한다. 하지만 AUX 채널을 통해 데이터가 전송되는 동안 싱크 장치의 Stretched I<sup>2</sup>C 마스터 블록은 SCL을 "Low"로 유지하여 대기한다. 다음 명령과 주소가 전송이 되면 다시 Stretched I<sup>2</sup>C블록은 활성화되어 EDID 정보를 액세스한다. 그림 5는 Stretched I<sup>2</sup>C의 데이터 전송 과정을 나타낸다. 그림과 같이 데

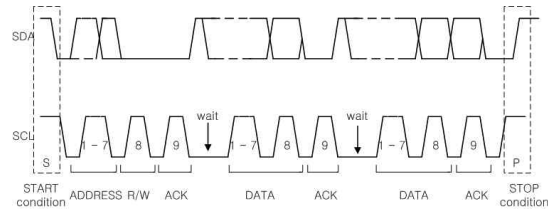


Fig. 5. The data transmission sequence of I<sup>2</sup>C stretched  
그림 5. Stretched I<sup>2</sup>C 데이터 전송 과정

이터가 전송되지 않는 구간에는 SCL을 "Low"로 유지하여 전력 소모량을 줄일 수 있게 된다.

세 가지 정보를 바탕으로 싱크 디바이스가 소스 디바이스에서 전송하는 데이터를 재생시킬 수 있다면 Stream Policy Maker는 Stream Ready 신호를 Link Policy Maker에 전송하여 Link Policy Maker가 Link Training 작업을 수행할 수 있게 한다. 그렇지 않다면 Link Training이 이루어지지 않아 데이터 전송을 할 수 없게 되고 메인 링크는 구성되지 않는다.

DPCD에는 다양한 정보를 담고 있지만 본 구현에 제안된 보조 채널 수신기의 DPCD는 AUX 채널 구성을 위한 수신기의 성능 정보와 링크 트레이닝 정보 등의 제한된 번지를 사용하였다.

현재 공개된 EDID 버전은 디스플레이포트를 지원하지 않기 때문에 모델링을 통하여 구현하였다. 제안된 EDID는 재생할 수 있는 영상의 타이밍 정보와 음성의 속성 정보를 저장하였다.

#### IV 시뮬레이션 및 측정결과

회로의 모의실험은 Synopsys사의 VCS 시뮬레이터를 이용하였고 FPGA 검증에 위한 합성 툴로 Xilinx ISE 9.2i를 사용하였으며, 검증에 사용된 보드는 SoC Master3를 이용하였다.

그림 6은 Link Policy Maker에 의해서 Link Training 상태의 변화를 나타내는 모의실험 결과이다. Link Training의 주요 상태인 클록복원 시퀀스(Clock Recovery Sequence)와 채널복원 시퀀스(Channel Equalization Sequence)는 클록을 복원할 데이터가 필요하고 채널상에서 데이터의 신호 감쇠가 일어나는 실제적인 상황이 필요하므로 물리층의 IP들을 모델링하여 Link Training을 검증하였다. LinkTrainingState는 Link Training의 상태를 나타내고 CRlock,

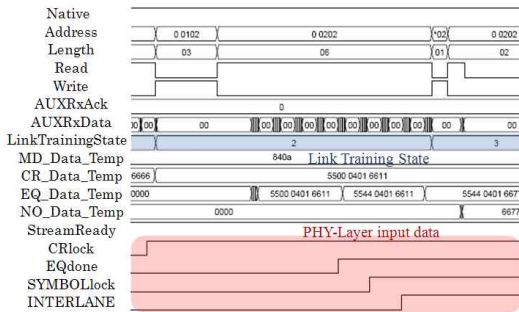


Fig. 6. Link Training simulation waveform of Link Policy Maker

그림 6. Link Policy Maker의 Link Training 실험

EQdone, SYMBOLlock, INTERLANE는 각각 물리층의 CDR(Clock Data Recovery), Word Aligner와 Equalizer로부터 입력되는 신호를 나타낸다.

그림 7은 Stream Policy Maker에서 디스플레이 장치가 수신된 데이터를 재생할 수 있는지 검사하는 모의실험 결과이다. HOR\_ACTIVE와 VER\_ACTIVE는 그래픽 카드에서 송신하는 영상 데이터의 해상도를 나타낸다. audio\_spec은 InfoFrame data generator로부터 수신된 데이터이며 max\_hor\_freq, max\_ver\_freq, audio\_info 등은 EDID로부터 수신된 데이터이다. 이 세 가지 정보를 바탕으로 디스플레이 장치에서 영상 재생이 가능하면 stream\_ready는 "High"가 되어 Link Training이 시작될 준비가 완료된다.

AUX 채널을 통해서 이루어지는 데이터의 전송 흐름은 짧은 시간에 종료되므로 Xilinx의 ChipScope를 사용하여 데이터의 흐름을 확인하였다. 그리고 메인 링크 구성 완료를 확인하기 위해서 그림 8과 같이 FPGA보드의 LED를 사용하여 그 동작을 확인하였다. 그림 8에서 1,2,3번은 각각 CRdone, DQdone, SYMBOLlock & INTERLANE을 나타내고 4번은 Link 구성 실패 5번은 링크 구성 완료를 나타낸다. 정상동작을 확인하는 그림 8의 1,2,3,5번의 불이 켜져 있음을 볼 수 있다.

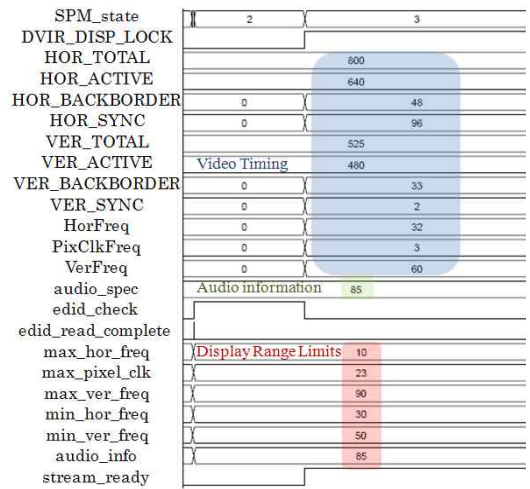


Fig. 7. Stream ready transport simulation waveform of Stream Policy Maker

그림 7. Stream Policy Maker의 스트림 데이터 전송 준비 실험

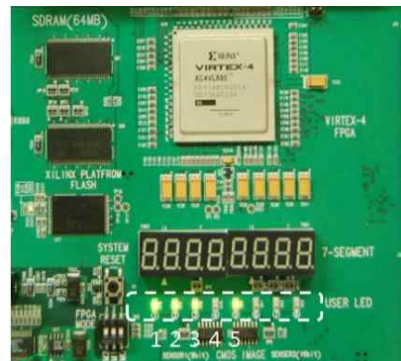


Fig. 8. The result of link training of auxiliary channel for DisplayPort

그림 8. 디스플레이포트 보조 채널 링크 트레이닝 결과

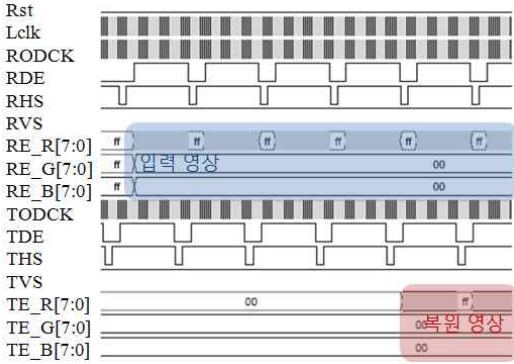


Fig. 9. The simulation result of reconstruction video of receiver

그림 9. Stream Policy Maker의 스트림 데이터 전송 준비 실험



Fig. 10. The result of video data waveform by ChipScope

그림 10. ChipScope를 이용한 영상 복원 결과 파형

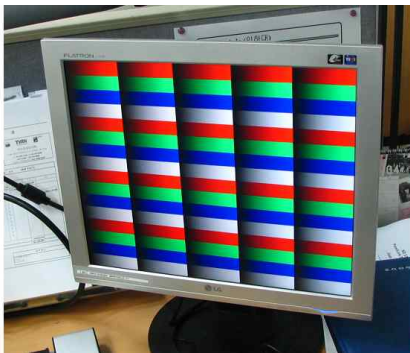


Fig. 11 The result of main link test with test pattern after the link training

그림 11 링크 트레이닝 후 메인 링크 테스트 패턴 결과

그림 9는 링크 트레이닝이 완료된 후 메인 링크를 통해 테스트 패턴이 모니터에 출력 되는 것을 시뮬레

이션으로 확인하였고, 그림 10은 Chipscope를 이용하여 입력영상과 출력영상을 얻은 것으로 두 데이터가 일치함을 확인할수 있다. 그림 11은 AUX 채널의 확립 후 테스트 영상을 수신단으로 보내 정상적인 영상 데이터 송수신이 이루어짐을 보였다. 메인링크와 디스플레이포트 AUX 채널의 성능을 표 1에 정리하였다.

Table 1. Capability summary of DisplayPort AUX Channel

표 1. 디스플레이포트 AUX 채널 성능 요약

DisplayPort AUX Channel	
LUTs	3,315(6%)
Flip Flops	1,466(2%)
Maximum Frequency	168,782MHz
Gate count	623,229

## V 결론

본 논문에서는 DisplayPort의 AUX채널의 설계 및 동작 검증을 보였다. 본 논문은 SoC Master3 FPGA 보드를 사용하여 3315개의 LUTs, 1466개의 Flip Flops, 최대 동작 속도 168.782MHz의 결과를 얻었다. 이를 위해 Link Training 과정 중 실제 물리층으로부터 피드백 되어 입력되는 신호는 스위치로 그 역할을 대신하였는데 추후 실제 물리층 블록이 완성되면 FPGA 보드와 통합 연결하여 테스트를 진행하면 DisplayPort 전체 시스템의 검증이 될 것이다. 디스플레이포트는 현재 개발 중인 인터페이스로 비교할 수 있는 칩이 없고, FPGA의 내부 구조가 고정되어 있어 낮은 속도를 얻었지만 디스플레이포트 1.1a의 AUX 채널을 동작시키기에는 충분한 속도이다. 그리고 구현된 AUX 채널은 약 62만 게이트를 사용하였는데 동작 검증을 용이하게 하기 위해서 메모리 블록(EDID, DPCD)을 FPGA에 함께 구현하였기 때문에 사이즈 증가를 초래하였다. 메모리 블록과 메모리 컨트롤 블록이 약 58만 게이트를 사용하므로 AUX 채널이 차지하는 블록은 약 4만 게이트로 작은 사이즈를 갖는다.

## 참고문헌

[1] Yong-woo Kim, Seong-bok Cha, Jin-ku Kang, "A Design of DisplayPort Link Layer," International SoC Design Conference, pp.45-48, Busan, Korea, November 2008.

- [2] VESA, *VESA DisplayPort Standard*, Version 1, Revision 1a, January 2008.
- [3] VESA, *VESA DisplayPort Link Layer Compliance Test Standard*, Version 1.0, September 2007.
- [4] VESA, *VESA and Industry Standards and Guidelines for Computer Display Monitor Timing*, Version 1.0, Revision 0.8, September 1998.
- [5] CEA, *A DTV Profile for Uncompressed High Speed digital Interface*, August 2005.
- [6] VESA, *ENHANCED EXTENDED DISPLAY IDENTIFICATION DATA STANDARD*, Release A, Revision 2, September 2006.

저 자 소 개

**차 성 복** (정회원)



2007년 : 목원대학교 전자공학과 졸업 (공학사)  
 2009년 8월: 인하대학교 전자공학과(공학석사)  
 2009년 8월~현재: (주)실리콘웍스 재직중  
 <주관심분야> High Speed

Digital Interface

**윤 광 희** (정회원)



2008년 : 인하대학교 전자공학과 졸업 (공학사)  
 2010년 : 인하대학교 전자공학과 (공학석사)  
 2010년 1월~현재: (주)실리콘웍스 재직중  
 <주관심분야> High Speed

Serial 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 (공학박사)

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