

SOPC기반 광-센서 인터페이스에 관한 연구

손홍범, 박성모

전남대학교 컴퓨터정보통신공학과

e-mail : momil@chonnam.ac.kr, smpark@chonnam.ac.kr

A Study on Photonic sensor Interface in SOPC platform

Hong-Bum Son, Seong-Mo Park

School of Computer Information and Communication Engineering
Chonnam National University

Abstract

In this paper, we describe photonic sensor interface in SOPC(System on a programmable chip) platform. This platform uses device that has ARM922T processor and APEX FPGA area on a chip. We use two development kits. The one is embedded kit that using Intel's Xscale device, the another is SOPC kit that using Altera's Excalibur device. We implement some device logic that DMAC, ADCC, etc. and application.

I. 서론

바이오 센싱 분야는 의료 진단, 환경 모니터링, 식품 품질 검사 등의 높은 응용가치로 세계적으로 중요성이 점차 커지고 있으며 그중 광학적 바이오 센싱은 다른 측정 방법에 비해 감도가 높고, 신호 간섭이 없으며, 관련 기술이 많이 개발되어 있다는 장점을 가진다. 그리고, 최근 각광 받고 있는 시스템온칩(SoC) 기술은 하나의 칩에 여러 기능을 집적하여 복합기능, 저전력,

고성능을 가능하게 한 차세대 반도체 이다. 본 논문에서는 바이오 광-센서 응용시스템을 SoC화 하기 위해 필요한 인터페이스 회로들을 개발하고 검증할 수 있는 SOPC환경을 구축하고, 각종 센서들과의 인터페이스를 정의 하고 이를 구동할 수 있는 프로그램의 동작을 기술 한다. 구동 프로그램은 기존 MCU를 사용한 시스템과 달리 고 성능 저 전력의 ARM을 기반으로 하여 센서로부터 입력된 소스 신호를 GUI환경에 자유롭게 표시 한다. 센서 인터페이스를 위해서 광학 기반 거리, 온도, 조도 센서들과 더불어 각종 센서들을 내장하여 원하는 센서의 값을 가지고 테스트 할 수 있는 장비를 사용 하였다. 개발 환경은 임베디드 개발 장비와 SOPC 개발 장비를 사용 하였다. 임베디드 개발 장비는 인텔(Intel)사의 Xscale(PXA255)를 사용하였으며, SOPC 장비는 알테라(Altera) 사의 엑스칼리버(Excalibur) 장치를 사용한 개발 보드로 ARM922T 와 APEX FPGA를 내장한 장치이다. 이들 장비들은 리눅스를 기반으로 하여 구동 하였고, 임베디드 장비에서 구동프로그램을 개발하였으며, SOPC 개발 장비는 시스템 동작을 위해 필요한 각종 로직들을 설계에 사용 하였다. 2장에서는 개발 시스템 및 환경에 대하여, 3장에서는 SOPC를 위해 설계된 각종 하드웨어에 대하여, 4장에서는 구동 소프트웨어에 대하여 설명하고 결론으로 본 시스템의 활용 분야에 대하여 기술 한다.

본 연구는 산업자원부 지방기술혁신사업(RTI04-03-03) 지원과 IDEC 톨 지원으로 수행되었음

II. 개발 환경

개발은 각 개발 장비와 더불어 리눅스가 탑재된 호스트PC에서 이루어졌다. 그림 1은 개발을 위한 구성도이다.

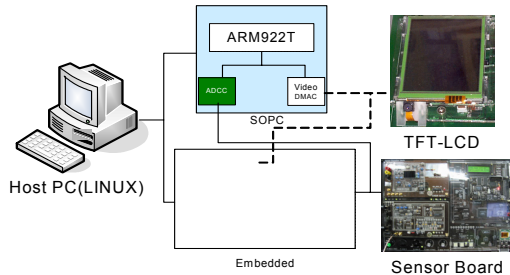


그림 1. 개발환경

2.1 Intel Xscale 기반 시스템

Xscale은 ARM명령어 셋으로 구동되는 인텔사의 마이크로프로세서이다. 이는 ALU (Arithmetical and Logical Unit), 주소 버스, 데이터 버스, 레지스터 등의 기본 기능에 내부 메모리, 산술 연산 장치, 메모리 제어 장치, TFT-LCD 구동 장치, USB 클라이언트 등의 각종 주변 장치들을 내장한 온칩 형태의 프로세서이다.[1] 사용한 보드는 이 마이크로프로세서를 중심으로 64MB의 SDRAM과 32MB의 플래쉬 메모리를 탑재하고, 메모리 맵 방식과 GPIO를 사용한 주변장치 확장을 위해 알테라사의 Cyclone FPGA를 탑재하고 있다. 이 FPGA영역을 설계하여 센서 드라이버의 출력단과 연결하였고 응용 프로그램에 의하여 이 값을 처리하도록 하였다. 응용 프로그램은 Linux 기반으로 O/S 수준에서 지원하는 프레임버퍼 영역에 그래픽 환경을 사용하여 센서의 값을 표시하게 된다. 응용 프로그램에 대해서는 뒷장에서 자세히 다룬다.

2.2 SOPC 시스템

알테라사의 엑스칼리버는 ARM922T 코어와 FPGA, SDRAM 컨트롤러 등을 내장한 SOPC 개발 및 검증을 위한 디바이스이다. 엑스칼리버는 FPGA의 크기에 따라 3가지 종류로 구분 되는데 그중 40만 게이트의 FPGA를 내장하고 있는 EPXA4 디바이스를 사용하였다. 사용한 개발 보드는 이 EPXA4 디바이스를 중심으로 하여 TFT-LCD, VGA 컨트롤러, SDRAM, CIS 등 각종 주변장치를 장착 하고 있으며 제어할 수 있는 제어 로직을 설계하여 FPGA 영역에 다운로드 함으로써 각 장치를 구동할 수 있다.[2] 이 개발 보드에서 센서 인터페이스 시스템을 구현하기 위해서는 센서로부터 입력을 받기위한 ADC를 구동할 수 있는 제어 로직과

또 이를 처리한 후 출력하기 위한 비디오 DMA 제어 로직을 설계 하여야 한다.

2.3 개발 환경 구축

인텔의 PXA255를 사용한 임베디드 개발장비와 Altera사의 엑스칼리버를 사용한 SOPC 개발장비 모두 임베디드 리눅스를 사용하여 구동 하였다. ftp://krenel.org에서 다운받을 수 있는 2.4.x버전의 커널소스를 각 타겟 디바이스의 특성에 맞도록 패치를 하고 장비별 특성(메모리 맵등)을 추가로 수정하여 컴파일 하기 위하여 호스트의 O/S는 레드햇 리눅스(Redhat Linux)를 사용 하였다. 디바이스 패치 파일은 ftp://ftp.arm.linux.org.uk/ 에서 얻을 수 있다. PXA255나 엑스칼리버는 모두 ARM명령어 셋에 의하여 동작하도록 되어 있으므로 이들 디바이스를 구동할 수 있는 프로그램을 생성하기 위해서 호스트의 레드햇 리눅스상에 교차 컴파일러(Cross Compiler)를 설치하였다. 이 교차 컴파일러는 http://gnu.org 에서 다운받을 수 있는 gcc를 호스트에서 특정 옵션을 주어 생성 할 수 있다. 또 다른 방법으로는 커널 패치 자료를 얻은 ftp://ftp.arm.linux.org.uk/ 에서 바이너리 형태의 컴파일러를 구할 수 있다.

III. 하드웨어 설계

인텔사의 PXA255는 각 주변장치들에 대한 제어 로직을 탑재한 온칩 형태의 프로세서 이지만, Altera사의 엑스칼리버를 사용한 SOPC 개발 장비에서는 비록 각종 주변장치들이 구성은 되어 있으나 실질적으로 이를 구동하기 위해서는 디바이스 내 FPGA영역에 이들 장치들을 구동할 수 있는 제어 로직을 설계하여 다운로드 하여야 한다. 이를 위해 Altera사에서 제공하는 Quartus 4.0 툴을 사용하여 센서 인터페이스를 위한 ADC 제어 로직, 출력을 위한 비디오 DMA 제어 로직 그리고 이들 제어 로직을 프로세서와 연결하기 위한 버스 설계를 하였다. 그림 2는 Quartus의 RTL뷰어를 이용하여 나타낸 탑-레벨 블록도이다.

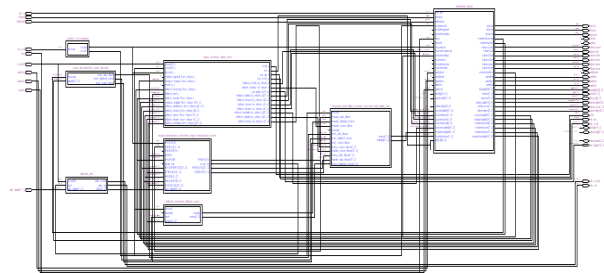


그림 2. 탑-레벨 블록도

3.1 비디오 DMAC

비디오 DMA는 영상 데이터를 출력하는데 있어서 중앙처리장치의 부담을 덜어주고 지속적인 영상 데이터의 출력을 위해 고안된 방법이다. 비디오 DMA 제어 로직은 그 특성상 대상 주소는 고정된 형태이며 구동을 위해서는 영상의 크기, 영상이 저장된 메모리 주소, 시작 신호등을 프로그래밍 해주도록 설계되어야 한다. 비디오 DMA 제어 로직은 Altera사에서 제공하는 Application Note 287[3]의 매뉴얼과 소스를 토대로 설계하였다. 그림 3은 비디오 드라이버의 흐름도이다.

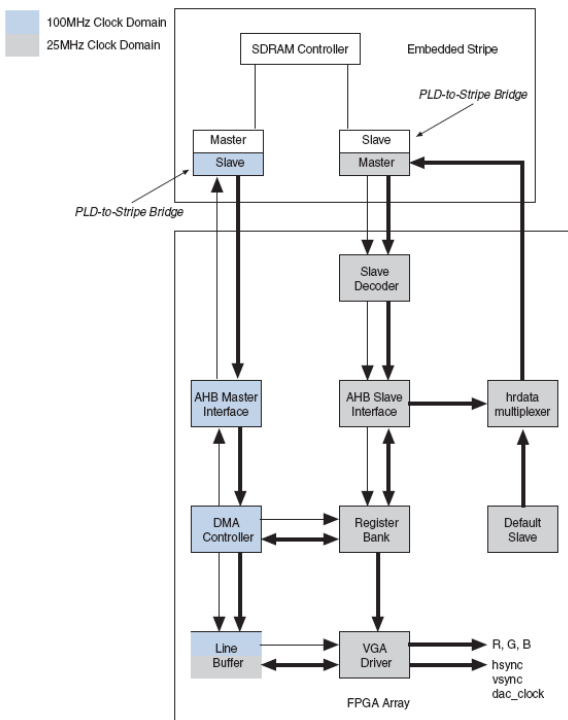


그림 3. 비디오 드라이버 흐름도[3]

비디오 드라이버는 수평 동기 신호를 그림 4와 표1과 같은 타이밍에 맞추어 출력 하게 된다. 수직 동기 신호 또한 640x480 VGA를 표현하기위한 타이밍에 맞추어서 출력 하여야 한다.

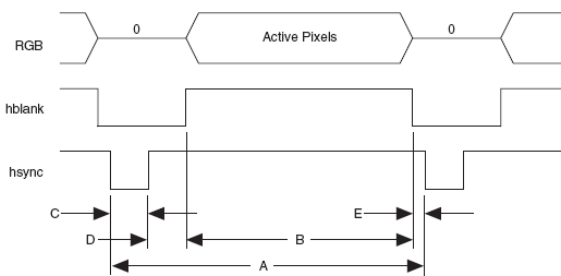


그림 4. 640x480 VGA 수평 타이밍[3]

표 3. 640x480 VGA 수평 타이밍[3]

Parameter	Description	Time(μs)
A	Line scan Period	31.77
B	Active video period	25.17
C	Sync period	3.77
D	Back porch	1.89
E	Front porch	0.94

3.2 ADC 인터페이스

ADC는 TI-ADS805를 사용하였다.[4] ADC 인터페이스 부는 ADC가 동작하기 위해 필요한 클럭을 생성 공급하여 주어야 하며, ADC로부터 데이터를 읽어 오기 위해 필요한 Data Out 신호등을 생성 하여야 한다. 또 데이터를 버퍼에 저장하고 이 내용을 버스에 실어 보낼 수 있는 로직도 고려되어야 한다. 그림 5와 표 2는 사용한 ADC의 타이밍을 보여준다. 여기서는 시스템 클럭을 1/2로 다운하여 ADC 클럭으로 사용하였다.

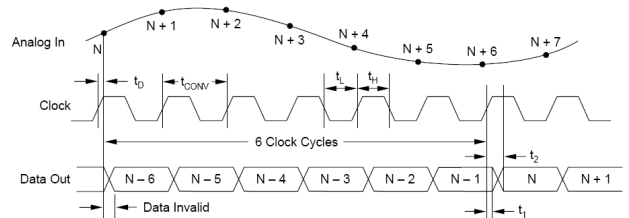


그림 5. ADC (TI-ADS805) 타이밍[4]

표 4. ADC(TI-ADS805) 타이밍(단위 ns)[4]

SYM	DES	MIN	TYP	MAX
t _{CONV}	Convert Clock period	50		100μs
t _L	Clock pulse LOW	24	25	
t _H	Clock pulse HIGH	24	25	
t _D	Aperture Delay		3	
t ₁	Data Hold Time	3.9		
t ₂	New Data Delay Time			12

3.3 버스

ARM에서는 AMBA(Advanced Microcontroller Bus Architecture)라는 버스 규격을 사용한다.[5] 이 버스는 마스터와 슬레이브가 존재하고 마스터에서 슬레이브를 제어하기 위한 신호를 생성하도록 되어 있으며 슬레이브는 이 신호에 응답하여 동작하거나 신호를 보내주도록 구성되어 있다. 위에서 설계한 비디오 DMA 제어 로직과 같은 경우는 마스터 또는 슬레이브로 동작하게 되며 ADC 제어 로직과 같은 경우에는 슬레이브로 동작하게 된다. Altera사의 엑스칼리버 디바이스는 AHB로만 구성되어 있는데 AHB는 다시 AHB1(고속)과 AHB2(저속)으로 구분된다.[3] 그림6은 엑스칼리버의

버스 구조와 DMA 제어로직에 의한 메모리 I/O를 보여주고 있다.

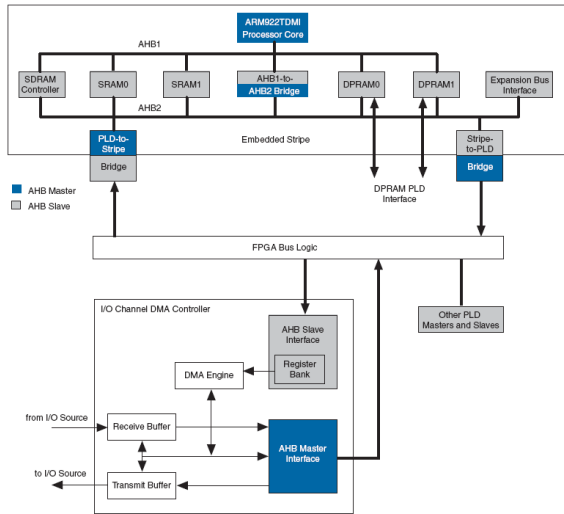


그림 6. DMA 제어로직에 의한 메모리 I/O[3]

비디오 DMA 제어로직은 0x80000000의 주소에, ADC 제어로직은 0x90000000의 주소에 각각 위치 한다.

IV. 구동소프트웨어

인텔의 PXA255를 사용한 임베디드 장비와 Altera의 엑스칼리버를 사용한 SOPC 개발 장비에서 소프트웨어의 동작은 동일하나 그 구성에는 약간의 차이가 있다. PXA255 기반의 장비는 리눅스 상에서 프레임버퍼를 제공 하여 손쉽게 센서 데이터의 그래프 표시를 쉽게 구현할 수 있다. 반면 엑스칼리버의 경우에는 프레임버퍼를 제공하지 않기 때문에 사용자가 임의로 유사한 기능을 구현하여야 한다.

4.1 인텔 PXA255 기반

PXA255를 사용한 개발 장비에 장착된 Altera사의 Cyclone디바이스를 사용한 FPGA보드를 프로그래밍하여 ADC디바이스를 부착 하였다. FPGA에서는 중앙 처리장치로부터 오는 주소를 디코딩하여 ADC디바이스의 것과 일치 할 경우 ADC에 의하여 변환된 센서의 값을 데이터 버스에 실어주게 된다. 따라서 ADC제어 로직을 설계할 필요가 있으며 이를 위해 마찬가지로 Altera사의 Quartus 4.0 툴을 사용하였다. 이렇게 구성된 환경에서 ADC제어를 위한 디바이스 드라이버를 작성하고 이 디바이스 드라이버로부터 데이터를 읽어와 그래프로 변환하여 프레임 버퍼에 기록하는 것으로 센서 데이터의 그래프 출력이 구현 되었다.

4.2 Altera 엑스칼리버 기반

엑스칼리버 기반의 환경에서는 설계된 비디오 DMA 제어로직의 구동을 위한 드라이버와 마찬가지로 ADC 제어로직을 구동을 위한 드라이버를 작성하였다. 이때 비디오 메모리 영역으로 전체 128MB중 8MB를 할당 하였다. 응용프로그램은 프레임 버퍼 대신 이 8MB영역에 센서의 데이터를 그래프로 변환하여 기록하게 된다. 따라서 비디오 DMA제어 로직의 Source 주소에도 이영역의 주소가 기록되게 된다.

그림 7은 구현한 프로그램의 동작 사진이다. 현재 까지 입력된 센서의 값을 그래프화하여 TFT-LCD화면에 표시하였다.

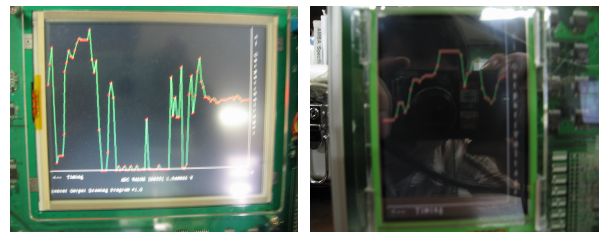


그림 7. 동작화면(좌: PXA255, 우: 엑스칼리버)

V. 결론

지금까지 SOPC환경에서의 센서 인터페이스를 기술 하였다. 여기에 좀더 실용적인 광-센서를 접합하고 구동 회로를 SoC기술로 소형화, 다기능화, 저 전력화 한다면 여러 가지 바이오광 응용 시스템에 사용될 수 있을 것이다.

참고문헌

- [1] Intel Corporations, Intel PXA255 Microprocessor Developer's Manual, January, 2004.
- [2] 한백전자, Hanback SoC-Entry II User's manual v1.0 Edition, 2005.
- [3] Altera, Using Excalibur DMA Controllers for Video Imaging , Application Note 287 , February 2003
- [4] Texas Instruments, 12-Bit, 20MHz Sampling ANALOG-TO-DIGITAL CONVERTER, ADS805, January 1997
- [5] ARM, AMBA™ Specification Rev 2.0, May 1999.
- [6] <http://www.gnu.org>
- [7] <http://www.arm.linux.org.uk>