

A Reconfigurable Image Processing SoC Based on LEON 2 Core

이 봉 규*
(Bongkyu Lee)

Abstract - This paper describes the design and implementation of a System-on-a-Chip (SoC) for image processing applications to use in wearable/mobile products. The target Soc consists of LEON 2 core, AMBA/APB bus-systems and custom-designed controllers. A new FPGA-based prototyping platform is implemented and used for design and verification of the target SoC. To ensure that the implemented SoC satisfies the required performances, an image processing application is performed.

Key Words : SoC (System-on-a-chip), LEON2, FPGA

1. 서 론

휴대폰이나 개인정보단말기(PDA)와 같은 개인용 이동기에 카메라(영상입력 장치)가 보편화되면서, 기존에 데스크탑 PC나 서버 등에서 이루어지던 영상처리/인식 기능을 개인용 이동기에 구현하려는 연구가 활발히 진행되고 있다. PDA를 이용한 얼굴인식 시스템 [1], 웨어러블 (wearable) 컴퓨터에 개인 얼굴 인식시스템 구축 [2], self-tracker 시스템 [3] 그리고 모바일용 카메라 자동초점 (Auto Focusing) 시스템 [4] 등이 실제적인 예이다. 이런 영상처리/인식 기능들을 개인용 이동기기 같은 임베디드 (Embedded) 시스템에 구현하는 방법은 1) 순수 소프트웨어적인 구현 (Fully Software Implementation), 2) 순수 하드웨어적 구현 (Fully Hardware Implementation) 그리고 3) 소프트웨어/하드웨어 공동 구현 방법으로 나누어 볼 수 있다[5]. 순수 소프트웨어로의 구현방법은 가장 유연하고 편리한 방법이지만, 부동소수점 연산을 포함한 많은 연산시간을 요구하는 영상처리/인식 기능의 구현에는 적합하지 못하다. 그 이유는 개인 이동형 기기는 전력 소모량 및 가격문제로 인하여 별도의 부동소수점 유닛 (Floating Point Unit, FPU)을 지원하지 않는 낮은 성능의 프로세서와 제한된 하드웨어 자원으로 구성되기 때문이다. 순수 하드웨어적인 구현은 가장 효율적인 방법이나, 구현과정이 복잡하고 설계에 유연성이 없으며 개발비용도 많이 드는 단점이 있다. 이런 단점들을 보완하기 위해 단일 칩에 소프트웨어/하드웨어를 동시에 집적할 수 있는 시스템은 칩 (System-on-a-chip, SoC)을 통해 영상처리/인식 기능을 개인 이동기에 구현하는 것이 주목 받고 있다[6].

SoC 설계방법은 다양한 Intellectual Property (IP)와 개발 플랫폼을 이용하여 하드웨어와 소프트웨어를 동시에 개발/통합/검증할 수 있기 때문에 다양한 기능의 단일 칩을 쉽게 구현할 수 있는 방법이다[6]. IP를 활용하면 기존의 검증된 설계 도면을 하나의 기능 블록으로 사용할 수 있기 때문에 설계 시간을 단축할 수 있다. 플랫폼 기반의 설계 방법은 먼저 기본이 되는 플랫폼을 설계하고, 응용 분야에 따라 확장/변화시켜 사용하는 방법이다. 현재 국내에서도 다양한 SoC의 개발에 이런 설계 방법을 적극적으로 활용하고 있다. 그러나 이런 설계방법의 사용에도 불구하고 국내에서 다양한 형태의 SoC를 구현하여 상용화를 하는데 많은 제약이 따른다.

가장 큰 문제점은 사용 IP에 대한 로열티 (Royalty)이다. 자체적으로 임베디드용 프로세서와 같은 핵심 IP를 보유하지 못한 우리의 경우, 대부분의 핵심 IP를 외국의 기술에 의존하고 있는 것이 현실이다. 예로써 현재 임베디드 시스템에서 가장 많이 이용되고 있는 ARM (Advanced RISC Machine) 프로세서를 이용하여 응용에 적합한 SoC를 개발/상용화할 경우 로열티가 요구되어진다. 따라서 다양한 응용 SoC의 개발에 적합한 핵심 IP의 개발이 시급하다. 두 번째는 핵심 IP를 효과적으로 개발할 수 있는 개발 플랫폼의 확보이다. 현재 사용되는 대부분의 플랫폼에서는 임베디드 프로세서가 고정된 형태로써 추가적으로 구현할 여러 하드웨어 블록들과 분리된 형태이다. 따라서 구현될 기능 블록들과 임베디드 프로세서를 단일 칩에 구현할 경우에 따른 통합테스트가 어렵다는 단점이 있다. 구현된 기능블록들과 임베디드 프로세서가 단일 칩과 같은 형태로써 플랫폼에서 완벽하게 검증이 이루어진다면 효과적인 개발이 가능할 것이다.

본 논문은 개인 이동형 기기에서 사용될 수 있는 응용 SoC개발에 사용이 가능한 핵심 IP를 구축하기 위하여, LEON 2 프로세서를 기반으로 한 재구성 가능 영상처리용

* 교신저자, 중신회원 : 제주대학교 전산통계학과

E-mail : bklee@cheju.ac.kr

접수일자 : 2008년 9월 20일

최종완료 : 2009년 3월 6일

SoC 코어를 설계하는 것을 목적으로 한다. 구현 과정은 다음과 같다. 먼저 목표로 하는 SoC의 설계/구현/검증을 위하여 Field Programmable Gate Array (FPGA)에 기반한 새로운 형태의 개발 플랫폼을 구현한다. 구현되는 플랫폼은 FPGA를 중심으로 외부 메모리 시스템, 디바이스 연결단, 버튼 및 LED 등을 가지는 보드로 제작된다. 그런 다음 LEON 2에 대한 VHDL 코드를 바탕으로 LEON 2 코어와 필요한 기능 블록, 내부 버스 시스템과 외부 디바이스 인터페이스를 포함하는 하드웨어 구조를 설계하고 구현한다. 구현된 SoC 코어 블록은 합성 툴(Synthesize tool)에 의해서 개발 플랫폼에 있는 FPGA에 매핑되어 Register Transfer Level (RTL)에서 검증된다. 구현된 SoC 코어의 초기화, 구동 및 소프트웨어 개발 환경을 제공하기 위하여 모니터 프로그램을 기반으로 한 독자적인 소프트웨어 환경을 구현한다. 실험에서는 특정 영상처리 관련 알고리즘을 구현된 SoC 상에서 실행시킨 결과를 일반 PC에서 실행시킨 결과와 비교하여 구현된 SoC가 유효함을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 LEON 2 프로세서에 대한 소개와 목표 SoC의 설계 및 검증을 위한 FPGA 기반의 개발 플랫폼에 대해서 기술한다. 3장에서는 목표로 하는 SoC에 대한 구현과정을 기술한다. 4장에서는 구현된 SoC를 이용한 영상처리 시스템의 예를 보이고 5장에서 결론을 맺는다.

2. LEON 2 기본 구조 및 개발 플랫폼 구축

2.1 LEON2 프로세서의 기본구조

LEON 2 프로세서는 [7] 로열티 없이 사용이 가능한 32 비트 범용 RISC 프로세서로써 VHDL 소스 형태로 제공되어 설계자들이 자신의 응용에 맞추어 재구성(Reconfiguration)을 할 수 있는 프로세서이다. 구조적인 측면에서 LEON 2는 코어 부분과 시스템버스 및 기본 하드웨어 모듈들(인터럽트 컨트롤러, 타이머, UART, I/O)로 구성되어있다. 그림 1은 LEON 2의 기본 구조를 보여준다.

LEON 2의 정수 유닛은 SPARC V8 구조에 기반한 5단계의 명령어 파이프라인을 가진다. 내부 캐쉬(cache)는 명령어와 데이터 캐쉬로 분리되어 있으며, 2-32 레지스터 집합을 지원하는 구조이다. FPU는 선택적으로 사용이 가능하다. 메모리 컨트롤러는 외부에 있는 메모리를 제어하는 유닛으로 8/16/32 비트 모드로 구성할 수 있으며, 최대 2Gbyte 주소공간을 관리할 수 있다. 구성요소간의 인터페이스를 위하여 LEON 2에서는 ARM에서 제공하는 표준의 AMBA AHB/APB 버스를 사용한다. 따라서 다른 IP를 첨가하거나 자체를 다른 프로세서와 결합하는 커스텀이징이 용이한 구조를 제공한다. AHB버스는 고속의 데이터 전송에 사용되며, APB버스는 주로 코어 외부에 있는 주변장치들의 온-칩 레지스터에 접근하는데 사용된다. AHB/APB 브릿지(Bridge)는 두 버스 시스템(AHB, APB)간의 인터페이스를 담당한다. 이들 장치들은 메모리 맵 방식으로 주소 공간에 매핑되어 있으며, LEON 2에서 기본적으로 할당된 주소영역은 표 1에서 확인 할 수 있다.

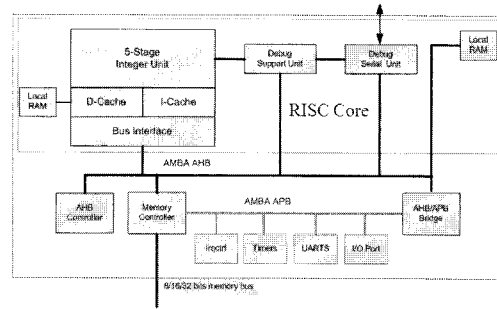


그림 1 LEON 2에 대한 블록도
Fig. 1 The block diagram of LEON 2

표 1 기본 주소 할당
Table 1 Default address allocation

주소 영역	크기	Mapping	모듈
0x00000000 - 0x1FFFFFFF	512M	PROM	Memory controller
0x20000000 - 0x3FFFFFFF	512M	I/O	
0x40000000 - 0x7FFFFFFF	1G	RAM	
0x80000000 - 0x8FFFFFFF	256M	on-chip registers	APB bridge
0x90000000 - 0x9FFFFFFF	256M	Debug support Unit	DSU

2.2 FPGA기반 개발 플랫폼 구축

LEON 2를 기반으로 한 영상처리용 SoC 개발을 효과적으로 진행하기 위해서는 적절한 개발 플랫폼의 활용이 매우 중요하며 개발 기간의 단축 및 검증의 편리성을 얻을 수 있다. 현재 일반적인 개발 플랫폼은 프로세서가 장착되고 메모리, 주변 제어 장치 및 관련 소프트웨어를 가지는 구조가 된다. 그러나 이런 일반적인 개발 플랫폼은 본 논문에서와 같이 범용 프로세서 코어의 구현까지를 포함하는 경우에는 적합하지 못하다. 따라서 기존의 IP들을 이용하여 새로운 기능의 SoC를 구현하는데 효과적으로 사용할 수 있도록 새로운 형태의 플랫폼을 제작하여 실제 개발과정에서 사용하였다. 구현된 개발 플랫폼은 설계한 전체 하드웨어를 (프로세서, 사용자 하드웨어 블록들) 단일 칩 영역의 형태로 개발 및 검증할 수 있도록 FPGA를 기반으로 구성되었다. FPGA는 특성상 LEON 2 코어를 비롯하여 설계되는 모든 하드웨어 블록들이 통합적으로 검증될 수 있도록 관련 하드웨어들이 한 곳에 매핑할 수 있기 때문이다[8]. 그림 2는 실제 구성된 개발 플랫폼의 블록도와 실제 사진을 보여준다.

개발 플랫폼에서 지원하는 외부 장치는 LCD와 카메라를 선택하였다. 그 이유는 목표로 하는 SoC가 영상처리/인식에 관련되기 때문에 목표 SoC 자체에 LCD와 카메라를 하드웨어적으로 제어할 수 있는 기능이 필요하기 때문이다. 또한 플랫폼에는 외부 데이터 및 프로그램 저장을 위하여 FPGA 외부에 SRAM기반의 메모리와 플래쉬 기반의 메모리를 가지고 있다. 부가적으로 LEON 2의 구현에 사용되어질 디버

경 인터페이스, 개발과정에서 사용하기 위한 개발서버와의 연결을 위한 시리얼 포트 등도 포함하고 있다. 표 2는 개발 플랫폼의 세부적인 사양에 대해서 나타낸 것이다.

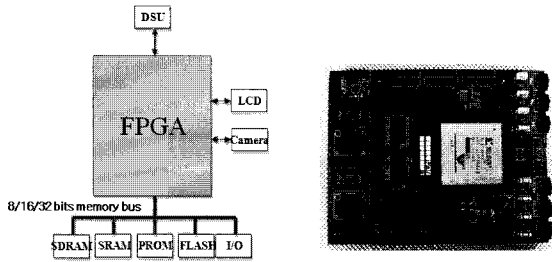


그림 2 개발 플랫폼
Fig. 2 The prototyping platform

표 2 구현된 개발 플랫폼의 주요 특징
Table 2 Major components of the platform

구성 요소	목적
FPGA(XILINX X2CV8000)	하드웨어 구현
PROM (XILINX XCF32PV048)	Fixed 로직 저장
SRAM(SAMSUNG, 128M)	데이터 저장
FLASH(Intel strata, 8M)	부트 프로그램, 응용프로그램 저장
카메라 (MICRON MT9V112)	영상 입력
LCD (SAMSUNG)	출력용
시리얼 인터페이스 (DSU)	다운로딩/디버깅

3. LEON2 기반 재구성 가능 영상처리 SoC 설계

이동형 기기에 영상 처리/인식 기능을 내장하는 방법은 해당 알고리즘에 대한 하드웨어/소프트웨어 통합설계를 통하여 단일 칩으로 구현하는 것이 효과적이다. 이런 개발 과정에서 소프트웨어 수행을 위한 기본 CPU 코어, 영상의 입력 및 출력 기능을 담당하는 디바이스 제어 블록, 내부 버스 인터페이스, 디바이스를 하나의 IP 블록으로 제공하고 제약 없이 활용 가능하다면 원하는 SoC 개발을 효율적으로 할 수 있다 [9]. 또한 중요한 프로세서 관련 IP의 국산화를 통하여 국내 지적재산권을 넓히는 계기가 된다. 이러한 요소들을 고려하여 구현된 LEON 2 기반의 영상처리용 SoC 코어의 설계 및 구현 과정을 기술한다.

3.1 목표 SoC의 하드웨어 구조

그림 3은 LEON 2 코어를 중심으로 설계된 영상처리 응용에 사용이 될 수 있는 재구성 가능 SoC의 블록 다이어그램이다. LEON 2의 코어부분은 Gaisler Research Group에서 제공하는 VHDL 소스를 기반으로 구현하였다. CPU외에 내부에 구현된 하드웨어 요소는 버스 시스템을 위한 AHB/APB 브릿지 (Bridge), 카메라 인터페이스 컨트롤러, LCD 컨트롤러 및 메모리 컨트롤러이다. CPU부분과 다른 하드웨어 블록간의 인터페이스를 위한 버스 시스템은 AMBA AHB/APB 표준 버스 시스템으로 구현하였다.

AHB 버스의 경우 마스터 (Master)와 슬레이브 (Slave)의 구성에 따라서 AHB_1과 AHB_2로 분리하여 구현하였다. 2층의 버스들은 디코더에 의해서 하나의 시스템 버스로 상호 연결이 되는 구조를 가진다. APB 버스 시스템은 내부 컨트롤러와 외부 장치의 연결을 위하여 구현되었다. APB버스에 연결된 I²C 마스터 및 SPI 마스터는 입/출력을 위한 외부 장치를 연결하는데 사용되어지는 인터페이스이다.

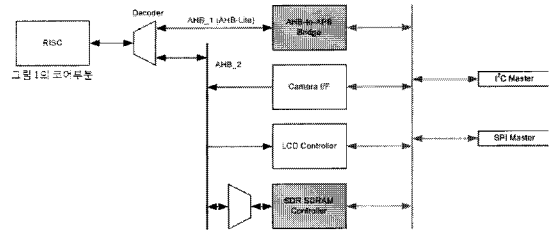


그림 3 구현된 내부 구조도
Fig. 3 The block diagram of the implemented architecture

AHB_1 버스는 하나의 버스 마스터 (LEON 2, 그림 3에서 RISC로 표시)와 하나의 슬레이브 (AHB/APB Bridge)를 가지는 구조이기 때문에 AHB_Lite 규약에 따라 설계한 버스 시스템이다. 이에 비해서 AHB_2 버스의 경우는 다수의 마스터들과 (CPU, 카메라 인터페이스, LCD 컨트롤러) 하나의 슬레이브 (메모리 컨트롤러)로 구성하였기 때문에 중개기(Arbitor)를 통해 버스의 점유와 해제를 중재하도록 하는 구조를 가진다. 표 3에서는 AHB/APB 브리지를 통하여 연결되어지는 APB 슬레이브와 AHB_2 버스 시스템에 존재하는 마스터에 대한 세부적인 내용을 보여준다.

표 3 버스 시스템상의 슬레이브와 마스터
Table 3 Slaves and Masters on bus systems

APB 슬레이브	베이스 주소
I ² C master	0x0000_0000
SPI Master	0x0000_1000
카메라 인터페이스	0x0000_2000
LCD 컨트롤러	0x0000_3000
메모리 컨트롤러	0x0000_6000

AHB_2 마스터	버스 우선 순위
LCD 컨트롤러	0
카메라 인터페이스	1
CPU (LEON 2)	2

LEON 2 코어외에 SoC내에 설계되는 주요 하드웨어 블록은 카메라 인터페이스 모듈, LCD 컨트롤러가 있다. 이들 하드웨어 블록은 각각 버스 시스템 상에서 마스터로 동작을 하기 위하여 내부에 버스 시스템에 접근하기 위한 블록 (AHB_DMA, 그림 4)을 가지도록 하였다. AHB_DMA는 버스 시스템 상에서 Burst/Single 데이터 전송을 주도하는 AHB_Master블록과 DMA 방식을 통하여 시스템 메모리에 있는 데이터에 직접 접근할 수 있도록 해주는 DMA 컨트롤러 등으로 구성된다. 이중 AHB_Master 블록은 SoC에 추가되는 마스터 블록들이 버스에 연결되는데 필요한 공통적인 내용을 정리하여 본 연구팀에서 설계한 것이다. 따라서 이

블록의 구조는 AHB Interface에 규정되어 있는 것이 아니고, AHB Master들을 설계 할 때 마다 고려해야 할 내용을 집약하여 쉽게 이용하도록 자체 개발한 회로이다. 따라서 이 블록을 사용하면 다양한 하드웨어 컨트롤러를 쉽게 기존의 컨트롤러와 결합이 가능하다. 다음에 나타날 LCD 컨트롤러와 카메라 인터페이스는 내부에 AHB_DMA를 포함하여 구현된 것이다.

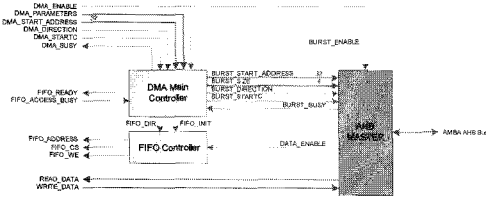


그림 4 AHB_DMA 내부 구조도
Fig. 4 The block diagram of the AHB DMA

카메라 인터페이스 모듈은 I²C 인터페이스에 연결된 물리적인 카메라로부터 CCI656 포맷의 320 X240 QVGA급 영상을 입력받은 후, CCI656 디코더(Decoder)를 통하여 Y 데이터를 추출한 후 이를 외부의 메모리 시스템 (SDRAM)에 저장하는 역할을 한다. 구현된 카메라 인터페이스 모듈은 I²C에 의해서 연결이 가능한 카메라에 대해서 공통적으로 사용이 가능하다. 그림 5는 구현된 카메라 인터페이스 모듈의 내부 구조를 보여준다. 그림에서 보듯이 내부에는 버스 시스템에 접근을 위하여 그림 4에서 보인 AHB_DMA를 사용하여 버스에 접근하는 구조로 설계되었다.

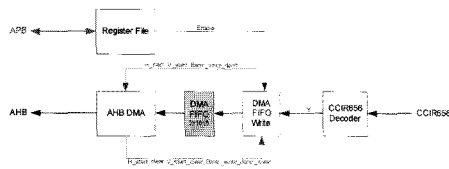


그림 5 카메라 인터페이스 내부 구조도
Fig. 5 The block diagram of the camera controller

LCD 컨트롤러는 영상의 입력 과정이나 처리 결과 등을 외부에 부착된 LCD로 출력하는 역할을 수행한다. 구현된 컨트롤러는 SPI (OpenCore에서 제공하는 IP)를 통하여 연결될 수 있는 다양한 LCD들을 제어할 수 있는데, 본 개발에서 사용한 LCD는 삼성 TFT-LCD LTV350Q-F06 모델이다. 구현된 LCD 컨트롤러 (그림 6)의 제어사양은 1) Sync 모드로 동작, 2) RGB 포트 중에서 Green 포트에만 Y 데이터를 출력하고 나머지 포트는 0으로 고정하는 방식이다. 사용된 프레임 (Frame) 주파수는 카메라와 동일하게 30Hz로 맞추었으며, Dot 주파수는 카메라의 픽셀 클럭 27MHz를 1/8 분주한 3.375MHz로 하였다.

3.2 운영 및 개발환경 구현

SoC는 하드웨어적인 요소와 소프트웨어적인 요소를 동시에 가지는 구조를 가지는 것이 일반적이다. 하드웨어의 경우는 실제 기능 블록을 구현하는 것이고, 소프트웨어의 경우

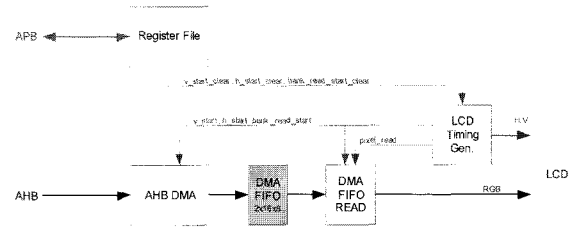


그림 6 LCD 컨트롤러 내부 구조도
Fig. 6 The block diagram of the LCD controller

는 주로 SoC에 펌웨어 (Firmware) 형태로 내부의 ROM이나 플래쉬 등에 저장되는 형태이다. 따라서 개발 플랫폼을 이용하여 SoC를 개발/검증하기 위해서는 FPGA에 구현되는 하드웨어를 초기화하고 작동시키는데 필요한 소프트웨어 요소를 개발하고 구현할 수 있는 개발환경은 SoC의 개발에 필수적인 요소이다. 이런 소프트웨어 요소 개발 환경을 구축하기 위하여 기존의 다른 개발 플랫폼의 경우에는 주로 범용 임베디드 운영체제인 Linux나 WinCE 같은 범용의 임베디드 운영체제를 개발 플랫폼에서 운영하고, 이 운영체제에서 소프트웨어 개발을 하도록 구성한다. 이러한 범용 운영체제를 사용할 경우 유연한 개발 환경을 제공한다는 장점이 있으나, 많은 하드웨어 자원을 필요로 한다는 단점이 있다. 이런 점을 고려하여 본 연구에서는 개발 플랫폼에 맞는 최적화된 부트로더 (boot loader, 그림 7)를 중심으로 소프트웨어 개발 환경을 지원한다. 구현된 부트로더는 보드에 전원이 들어왔을 때 제일 먼저 실행되는 프로그램으로 플래쉬 메모리에 저장된다. 주요기능은 FPGA내의 CPU를 비롯한 하드웨어를 초기화 한 후 버튼이 눌리거나 소프트웨어로 구현된 기능을 수행하고 다시 버튼 입력을 기다리는 과정을 반복하는 기능을 수행하도록 구현되었다. 구현된 SoC를 단일 칩으로 만들 경우, 이 부트로더는 펌웨어 형태로 SoC 내부에 고정되어 저장되고 칩의 기능을 제어하는 역할을 담당할 것이다.

구현된 SoC가 수행해야 할 사용자 정의 프로그램은 모니터 프로그램에 직접 삽입되어 구현되는데 구체적으로 부트로더에서 사용자 프로그램을 호출하는 방법을 사용한다. 영상처리 응용을 위하여 SoC를 사용할 경우, 원하는 기능을 수행하는 알고리즘을 구현한 후, 구현된 프로그램을 모니터 프로그램에서 호출하도록 모니터 프로그램을 수정하고 컴파일하여 링크를 시킴으로써 하나의 실행가능 모듈로 만들 수 있다. 이때 구현되는 알고리즘은 하드웨어/소프트웨어 co-design 기법을 이용하여 하드웨어 블록으로 구현되는 부분과 소프트웨어로 구현되는 부분이 나누어질 경우, 하드웨어는 VHDL를 이용하여 직접 FPGA내에 구현할 수 있으며 소프트웨어 부분은 부트로더의 사용자 함수 (User functions) 부분에 삽입되는 방법으로 구현되어진다. 프로그램의 개발을 지원하는 목적으로 다양한 입출력 기능은 그림 [7]에서 보듯이 사용자가 프로그램을 작성할 때 호출하여 사용할 수 있도록 미리 라이브러리로 만들어 제공한다. LEON 2에서 실행 가능한 코드를 만드는 방법은 LINUX 운영체제나 Cygwin상에서 운용되는 BCC-bare C 교차 컴파일러 툴체인을[7] 사용하여 구성할 수 있다.

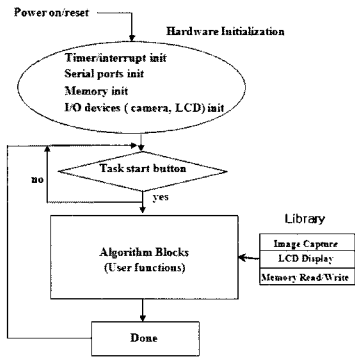


그림 7 운영 프로그램의 흐름도
Fig. 7 The flowchart of the monitor program

4. 구현 및 검증

목표 SoC에 포함되는 모든 하드웨어 요소들은 VHDL로 구현하였으며, XILINX 툴을 이용하여 FPGA에 매핑하였다. 그림 8에서 구현이 완료된 SoC의 내부 구조를 보여주고 있다. 점선으로 표현된 부분은 FPGA내부에 포함된 회로를 나타낸다. 내부에는 LEON 2 코어, 버스 시스템 그리고 새로이 설계된 하드웨어 모듈을 모두 포함하고 있다. 이렇게 모든 회로들이 모두 단일 FPGA상에 구현되었기 때문에, 단일 칩에서의 동작과 같은 효과를 낸다.

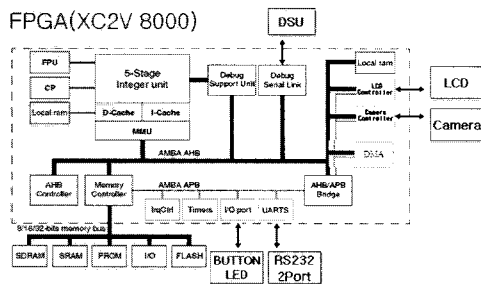


그림 8 운영 프로그램의 흐름도
Fig. 8 The flowchart of the monitor program

구현된 SoC의 동작을 검증하기 위하여 영상처리 알고리즘을 선택하여 실제 실행 여부를 검증하였다. 실행되는 알고리즘은 그림 7의 모니터 프로그램 내의 사용자 함수영역 (user functions) 부분에 삽입되어 하나의 프로그램으로 만들어진 후 컴파일되어 플래쉬 메모리에 저장된다. 사용한 알고리즘은 히스토그램 정보를 이용하여 입력 영상에 대해서 영역분할 (Segmentation)을 [10] 수행하는 것으로 실행 순서는 다음과 같다. 먼저 카메라를 이용하여 얻어진 후 메모리에 저장된 영상을 읽어와서 영상에 대한 히스토그램 정보를 추출한다. 추출된 히스토그램 정보를 바탕으로 영역분할을 수행한 후, 결과 영상을 다시 메모리에 저장하고 그 결과를 LCD에 출력하게 된다. 그림 9의 (a)는 그림 7의 사용자 영역user blocks영역에 삽입되는 히스토그램 기반 영역분할 알고리즘을 구성하는 함수들을 보여준다. 그림 9의 (b)는 플랫폼에 연결된 카메라에 의해서 캡처(capture)되어 플

랫폼의 SDRAM에 저장되어져 있는 입력영상을 보여준다. 그림 9의 (c)는 개발된 SoC에서 실행된 결과를 보여준다. 비교를 위하여 같은 알고리즘을 PC에서 소프트웨어로 구현하여 얻은 결과가 그림 9의 (d)에 나타나 있다. 이렇게 얻어진 두 개의 결과영상간의 차이를 알아보기 위하여 영상간의 차이를 구한 것이 그림 9의 (e)에 나타나 있다 (검은 부분 : 차이 없음, 흰 부분 : 차이 있음). 이러한 차이는 구현된 SoC의 경우, FPU가 없기 때문에 영상처리에 따른 실수연산을 모두 정수연산으로 변환하여 수행한 결과로 나타나는 것으로 판명되었다. 그러나 그림 9의 (e)에서 보듯이 차이는 매우 적은 것이기 때문에, 지금까지 기존의 PC 등에서 구현되던 영상처리 응용을 SoC에서 효과적으로 구현할 수 있음을 알 수 있다. 또한 하드웨어/소프트웨어 co-design에 의해서 시간이 많이 걸리는 함수들을 하드웨어 로직으로 구성할 경우 복잡한 알고리즘이라도 실시간에 처리가 가능하다. 실험을 통하여 구현된 SoC의 작동을 검증하였으며, 결과를 통하여 영상처리 관련 응용을 위한 새로운 IP 및 기반 SoC로 활용이 가능함을 보였다.

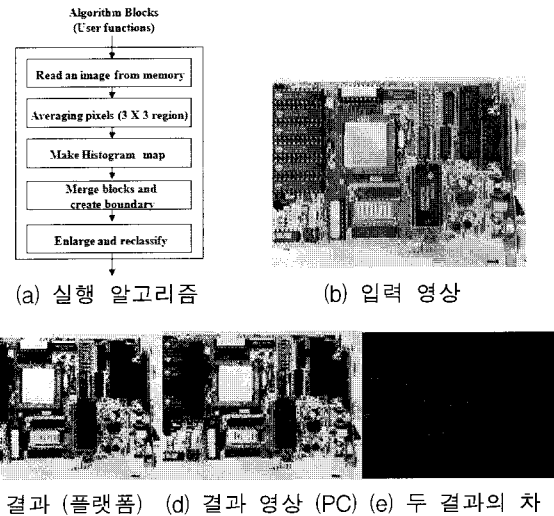


그림 9 실험 결과
Fig. 9 The experimental results

구현된 FPGA기반 플랫폼을 통한 LEON 2기반 SoC의 사용은 실제 응용 구현에 있어 다음과 장점을 가진다. 현재 주로 사용하고 있는 일반적인 개발 보드의 경우는 CPU 로직이 고정 (외부 혹은 내부)되어 있기 때문에 특정 응용에 맞추어 성능, 주변장치와의 인터페이스를 복합적으로 고려하기 어렵다. 또한 소스 수준에서 CPU로직에 대한 변경이 불가능하기 때문에 커스텀마킹 하드웨어 로직과의 인터페이스를 구현하는데 제약이 따른다. 이런 이유로 응용에 적합한 SoC를 만들 때 필수적인 성능개선, 저전력 사용, 칩 사이즈 최소화 및 비용 등에서 어려움을 겪게 된다. 그러나 본 연구에서 구현된 SoC의 경우는 FPGA에 모든 하드웨어 로직을 소스레벨에서 통합구현하고 이를 RTL에서 평가 및 검증이 가능하기 때문에 인터페이스 구현에 대한 제약이 없고, 목표로 하는 SoC의 초기 설계 단계에서 보드 레벨에서의 구현에서 생길 수 있는 여러 가지 문제점을 사전에 예방할 수 있어 효율적인 설계 및 구현을 할 수 있다.

5. 결 론

본 논문은 국내에서는 처음으로 LEON2 코어를 이용하여 영상처리/인식 기능의 구현에 IP로 사용이 가능한 새로운 형태의 SoC를 설계하고 구현하였다. 또한 목표 SoC를 설계/검증하기에 적합한 개발 플랫폼을 FPGA를 이용하여 구현하였으며, 이 플랫폼을 이용하여 실제 응용을 개발하는데 필요한 기본 소프트웨어 및 개발 환경을 구현하였다. 구현된 SoC는 개발 플랫폼에서 단일 칩으로 구현된 것과 같은 환경에서 검증이 되었으며, 다양한 영상처리/인식 기능을 임베디드 시스템에 구현하는데 사용되어 질 수 있음을 보였다. 구현된 SoC에 포함된 모든 하드웨어 요소들은 VHDL의 형태로 무료로 제공이 가능하기 때문에 다양한 상용 SoC의 개발에 효과적으로 사용이 가능하다. 본 연구팀에서는 계속적으로 다양한 영상처리/인식 관련 알고리즘을 하드웨어/소프트웨어 공동 설계 방법으로 구현된 SoC내에 탑재시킴으로써, 이동형 기기에서 실제로 요구되는 기능을 가지는 관련 SoC의 개발을 진행 중이다.

참 고 문 헌

- [1] J. Yang, X. Chen, W. Kunz, "A PDA-based Face Recognition System", Proceedings of WACV 2002, 2002.
- [2] Jong Bae Kim, "A personal identity annotation overlay system using a wearable computer for augmented reality," IEEE Transactions on Consumer Electronics, vol. 49, no. 4, pp. 1457-1467, Nov., 2003.
- [3] E. Foxlin and L. Naimark, "VIS-Tracker : A Wearable Vision-Inertial Self-Tracker," IEEE VR2003, March, 2003.
- [4] S. M. Shon, S. H. Yang, S. W. Kim, K. H. Baek and W. H. Paik, "Soc Design of An Auto Focus Driving Image Signal Processor for Mobile Camera Applications", IEEE Transactions on Consumer Electronics, vol. 52, no. 1, pp. 10-16, Feb., 2006.
- [5] T. Theocharides, G. Link, N. Vijaykrishnan, M. J. Irwin and W. Wolf, "Embedded Hardware Face

Detection", Proceedings of th 17th International Conference on VLSI Design (VLSID'04), Jan., 2004.

- [6] M. Shabiul Islam, M. Salim Beg, M. S. Bhuyan and Masuri Othman, "Design and Implementation of Discrete Cosine Transform Chip for Digital Consumer Products", IEEE Transactions on Consumer Electronics, vol. 51, no. 3, 998-1003, Aug., 2006.
- [7] LEON2 processor user's manual, Gaisler Research, <http://www.gaisler.com>
- [8] M. Brogatti, F. Lertora, B. Foret and L. Cali, "A reconfigurable system featuring dynamically extensible embedded microprocessor, FPGA, and customizable I/O", IEEE J. Solid-State Circuits, vol. 38, pp. 521-529, Mar. 2003.
- [9] "상위수준 시스템설계 및 프로토타이핑 기술 개발 - Reconfigurable디바이스 및 시스템구현," 국가 15대 기술 연구기획 최종보고서, 지식경제부, 2008
- [10] A. W. Naji and A. R. Ramli, "A Segmentation Algorithm based-on Histogram Equalizer For Fingerprint Classification System," Second International Conference on Electrical and Computer Engineering ICECE 2002, 26-28 December, 2002.

저 자 소 개



이 봉 규 (李 鳳 奎)

제주대학교 전산통계학과 교수

Tel : (064) 754-3593

E-mail : bklee@venus1.cheju.ac.kr

관심분야 : 영상처리 SoC 설계