

## Design and Implementation of Hardware for various vision applications

양 근 탁\* · 이 봉 규†  
(Keuntak Yang · Bongkyu Lee)

**Abstract** - This paper describes the design and implementation of a System-on-a-Chip (SoC) for pattern recognition to use in embedded applications. The target Soc consists of LEON2 core, AMBA/APB bus-systems and custom-designed accelerators for Gaussian Pyramid construction, lighting compensation and histogram equalization. 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, a pattern recognition application is performed.

**Key Words** : iSoC (System-on-a-chip), LEON2, Pyramid, Preprocessor, FPGA

### 1. 서 론

스마트폰이나 개인정보단말기(PDA)와 같은 개인용 이동 기기에 카메라(영상입력 장치)가 보편화되면서, 기존에 데스크 탑 PC나 서버 등에서 이루어지던 영상처리/인식 기능을 개인용 이동기기에 구현하려는 연구가 활발히 진행되고 있다. PDA를 이용한 얼굴인식 시스템 [1], 웨어러블(wearable) 컴퓨터에 개인 얼굴 인식시스템 구축 [2], self-tracker 시스템 [3] 그리고 모바일용 포터블 번역기 [4] 등이 실제적인 예이다. 이런 영상처리/인식 시스템은 개인의 상황인지 기능이나 실세계와의 연계성을 향상시키는 도구로 유용하게 사용이 될 수 있다 [1].

패턴/물체 인식 시스템은 일반적으로 1)전처리 2)영상내의 목표 대상 검출 3) 대상에 대한 개별 인식을 포함하는 후처리로 구성된다 [3]. 이런 단계 중에서 특히 목표대상을 효과적으로 검출하는 것은 전체 시스템의 성능을 좌우하는 핵심적인 역할을 담당하고 있는 매우 중요한 분야이다. 이런 이유로 자동 타겟 인식에서부터 산업용 자동검사기 등에 이르기까지 다양하게 적용되고 있다. 그러나 얻어지는 실세계 영상에는 빛의 영향이나 카메라의 특성으로 인하여 생기는 영상 픽셀 값의 편중에 따른 변이가 생길 수 있다. 이러한 변이들은 영상피라미드 (Gaussian Pyramid, GP) [5], 빛 보정 (Lighting Inhomogeneity Compensation, LIC) [6]과 히스토그램 평활화 (Histogram Equalization, HE) [7]의 조합을 통해 효과적으로 해결할 수 있다.

위에서 기술한 처리조합을 임베디드 시스템에 구현하는 방법

은 크게 소프트웨어적 구현 (Fully Software Implementation) 과 하드웨어적 구현 (Fully Hardware Implementation) 방법으로 분류할 수 있다. 소프트웨어적 구현방법은 비록 유연하고 구현이 용이하다는 장점이 있으나, 제한된 하드웨어 자원으로 구성된 임베디드 하드웨어에 적용하는 것은 무리가 있다. 일반적으로 영상처리 알고리즘은 부동소수점 연산을 포함한 많은 연산시간을 요구하기 때문이다. 따라서 알고리즘 자체를 전용 하드웨어로 구현하는 방법이 사용되어야 한다 [8]. 이런 이유로 영상에 대한 전처리 알고리즘에 대한 개별적인 하드웨어 설계 관련 연구가 꾸준히 진행되어 왔으며 다수의 결과가 보고되고 있다 [9,10,11]. 그러나 기존의 결과들은 대부분 패턴 인식 응용 시스템의 실제 구현을 위해서는 추가적인 자원을 요구하기 때문에 실제 임베디드 응용에 적용이 힘들다. 이런 문제점을 해결하기 위하여 실제 임베디드 시스템에 적용이 가능한 영상전용 SoC를 구현하는 것을 목표로 본 연구를 진행한다.

본 논문에서는 개인 이동형 기기를 포함한 다양한 임베디드 디바이스에 실시간 처리가 가능한 영상/패턴 인식 응용 시스템을 구현할 수 있는 Field Programmable Gate Array (FPGA) 기반의 SoC를 설계하는 것을 목적으로 한다. 구현 과정은 다음과 같다. 먼저 목표로 하는 SoC의 설계/구현/검증을 위하여 FPGA를 중심으로 하는 새로운 형태의 개발 플랫폼을 구현한다. 구현되는 플랫폼은 FPGA를 중심으로 외부 메모리 시스템, 디바이스 인터페이스, 버튼 및 LED 등으로 구성된다. 구성된 플랫폼에 있는 FPGA내에 메인 프로세서인 LEON2 [12], 영상처리 기능블록, 메모리 컨트롤러와 버스 시스템을 구현/매핑하고 합성 툴(Synthesize Tool)을 이용하여 Register Transfer Level (RTL)에서 기능을 검증한다. 구현된 SoC위에 패턴인식 시스템을 실제 실행하여 설계한 SoC가 패턴/영상 인식에 관련된 임베디드 응용에 효과적으로 사용될 수 있음을 보인다.

† 교신저자, 종신회원 : 제주대학교 전산통계학과

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

\* 비 회 원 : 제주대학교 전산통계학과 박사과정

접수일자 : 2010년 7월 26일

최종완료 : 2010년 11월 10일

## 2. LEON2 기본 구조 및 개발대상 알고리즘

### 2.1 LEON2 프로세서 구조 및 개발 플랫폼

LEON2는 코어 부분과 시스템버스 및 기본 하드웨어 모듈들(인터럽트 컨트롤러, 타이머, UART, I/O)로 구성되어 있다 (그림 1). 내부 구성요소간의 인터페이스를 위하여 LEON2에서는 표준의 AMBA AHB/APB 버스를 지원한다. AHB버스는 고속의 데이터 전송에 사용되어지며, APB버스는 주로 코어 외부에 있는 주변장치들의 온-칩 레지스터에 접근하는데 사용된다. AHB/APB 브리지(Bridge)는 두 버스 시스템 (AHB, APB)간의 인터페이스를 담당한다. 이들 장치들은 메모리 맵 방식으로 주소 공간을 할당받는 메모리 맵 (Memory Map)방식으로 수행된다.

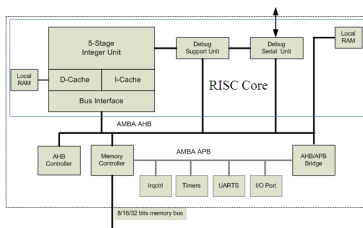


그림 1 LEON2에 대한 블록도  
Fig. 1 The block diagram of LEON2

LEON2를 기반으로 기존의 IP들을 이용하여 새로운 기능의 SoC를 구현하는데 효과적으로 사용할 수 있도록 새로운 형태의 FPGA기반 플랫폼을 제작하여 실제 개발과정에서 사용한다. 개발 플랫폼은 설계한 전체 하드웨어를 (프로세서, 사용자 하드웨어 블록들) 단일 칩 형태로 개발 및 검증할 수 있는 기능을 제공한다 [8]. 개발 플랫폼에서 지원하는 외부 장치는 LCD와 카메라이다. 그림 2는 실제 구성된 개발 플랫폼의 블록도와 실제 사진을 보여주며 표 1은 구현된 개발 플랫폼의 사양을 보여준다.

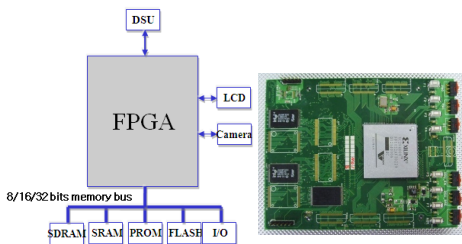


그림 2 개발 플랫폼  
Fig. 2 The prototyping platform

표 1 구현된 개발플랫폼의 주요 특징  
Table 1 Major components of the platform

구성 요소	목적
FPGA(XILINX X2CV8000)	하드웨어 구현
SRAM(SAMSUNG, 128M)	데이터 저장
카메라 (MICRON MT9V112 )	영상 입력
LCD ( SAMSUNG)	출력용
시리얼 인터페이스 (DSU)	다운로딩/디버깅

### 2.2 전처리를 위한 알고리즘 조합

본 논문에서 하드웨어로 구현할 알고리즘은 1) 입력 영상에 대한 GP 구축 2) 인식의 대상 영역에 대한 LIC 및 HE 수행으로 연결되는 3가지 단계로 구성된다.

GP 알고리즘은 입력 영상을 단계 0으로 하여 정해진 단계별로 일정 비율씩 축소되는  $N$ 개의 복수영상 집합을 만든다. 입력 영상  $g_0$  ( $C \times R$  크기)에 대해서 단계 1에 있는 영상  $g_1$ 은 일정 비율 (Factor)만큼 줄어든 크기의 영상으로 정의한다. 하위 단계의 영상으로부터 상위 단계의 영상을 구성하는 계산과정은 수식 (1)에 있는 REDUCE [10] 함수로 정의된다. 여기서  $N$ 은 목표 피라미드 단계의 수를 나타내며,  $M$ 과  $w$ 는 내부의 저대역 필터에서 사용되는 윈도우 크기 및 확률밀도함수이다. 그리고  $C_k, R_k$ 는  $k$ 단계에서의 영상의 크기를 나타낸다. 본 논문에서 구현되는 GP 구축에서 사용되는 축소비율은 1.2 (이전 단계의 83%에 해당)이다.

$$g_k(i, j) = REDUCE(g_{k-1}(i, j)) = \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{M}{2}}^{\frac{M}{2}} w(m, n) g_{k-1}(2i+m, 2j+n) \quad (1)$$

$0 \leq k < N, 0 \leq i < C_k, 0 \leq j < R_k$

LIC는 탐색 대상 영역에 분포하는 전체 밝기 분포를 추정할 수 있는 선형함수를 이용하여 구현된다. 구해진 선형함수에 의해서 추정된 전체 밝기 정보를 실제 영상에 적용함으로써 다양한 빛의 조건을 보상할 수 있다. 그 방법은 다음과 같다. 좌표  $(x, y)$ 에 있는 픽셀의 값을  $I(x, y)$ 로 가정하면, 픽셀 값들로부터 추정된 빛의 분포를 통하여 선형함수 계수  $(a, b, c)$ 를 얻은 후, 이 계수를 수식 (2)의 선형 방정식에 적용하여 새로운 픽셀 값들을 구한다. 이 기법은 탐색 영역에 있는 픽셀 값들에서 밝기 분포를 보상해주는 효과를 나타낸다. 그러나 LIC 연산에서 수식 (2)에 필요한 계수  $(a, b, c)$ 는 Pseudo-Inverse 방법을 통한 행렬식으로 구하기 때문에 상대적으로 많은 연산이 요구된다 [6].

$$I(x, y) = ax + by + c \quad (2)$$

HE는 탐색 대상 영역에 분포하는 픽셀 값들을 균일하게 재구성하는 방법을 통하여 영상을 보정하는 기법으로, 픽셀의 분포범위를 넓힘으로써 영상의 대비(Contrast)를 개선하는 방법이다. HE 처리가 된 영상은 히스토그램이 균일한 분포를 가지기 때문에 영상인식이나 패턴검출이 용이해진다. 픽셀이 가지는 범위가  $[0-L]$ 인  $M \times N$  크기의 영상에 대해서 HE 기법은  $O(M \times N)$ 의 복잡도를 가진다.

## 3. 목표 SoC 설계

### 3.1 개발 플랫폼

설계한 LEON2 중심의 SoC의 블록 다이어그램이 그림 3에 나타나 있다. CPU외에 내부에 구현된 하드웨어 요소는 버스 시스템을 위한 AHB/APB 브리지 (Bridge), 카메라 인터페이스 컨트롤러 및 메모리 컨트롤러이다. CPU부분과 다른 하드웨어 블록간의 인터페이스를 위한 버스 시스템은 AMBA AHB/APB를 사용한다. 실제 설계된 AHB 버스의

경우 마스터 (Master)와 슬레이브 (Slave)의 구성에 따라서 AHB\_1과 AHB\_2로 분리하여 구현한다. 이들 2층 버스는 디코더에 의해서 하나의 시스템 버스로 상호 연결이 되는 구조를 가진다. APB 버스 시스템은 내부 컨트롤러와 I<sup>2</sup>C 인터페이스를 지원하는 외부 카메라와의 연결을 위하여 사용되는 것으로 카메라로부터 들어오는 영상을 외부의 메모리 시스템 (SDRAM)에 저장하는 역할을 한다.

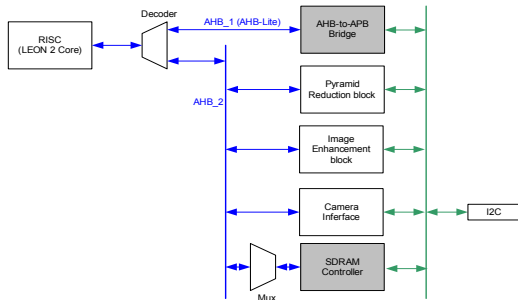
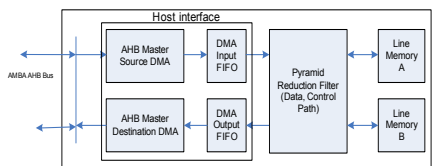
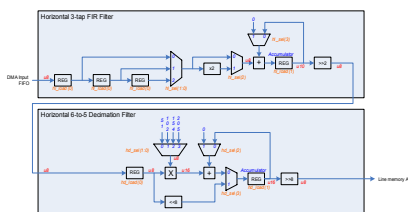


그림 3 구현된 내부 구조도  
Fig. 3 The block diagram of the implementation

그림 4는 설계된 GP 구축 알고리즘을 위한 Pyramid Reduction (PR) 블록의 하드웨어 구조와 핵심 요소인 Pyramid Reduction Filter (PRF) 모듈의 RTL 사양을 보여준다. PR 블록은 메모리 접근 및 버스 인터페이스에 사용되는 'Host interface' 부분과 실제 피라미드 연산을 수행하는 PRF 모듈로 구성된다. 'Host interface' 모듈은 PRF와 다른 컨트롤러 사이의 인터페이스를 제공하는 것으로, 직접 메모리 접근 (DMA) 관련 회로인 'Source DMA' 블록과 'Destination DMA'로 구성된다. 'Source DMA' 블록은 외부 SDRAM메모리에 있는 영상 데이터를 PRF에 전달하는 기능을 수행한다. PRF는 입력으로 받은 2차원 영상에 대해서 3-taps 저대역 필터 (low-pass filter)로 컨볼루션을 실행함으로써 피라미드 연산을 수행한다.



(a) 전체 블록도



(b) 수평(수직) 1차원 필터 RTL 블록도

그림 4 피라미드 연산블록  
Fig. 4 The pyramid reduction block

전체적인 PRF의 데이터 흐름 (Data Flow)은 다음과 같다. 'Source DMA' 블록은 SRAM에 있는 입력 영상에서 매번 16 픽셀을 'Input-FIFO' 버퍼로 읽어온다. PRF는 버퍼에 있는 픽셀을 대상으로 수평(Row)방향의 컨볼루션 (Convolution)과 데시메이션 (Decimation)을 수행하여 얻어진 결과영상을 내부 메모리인 'A'에 저장한다. 이런 후, PRF는 'A'에 저장된 픽셀을 대상으로 수직(column) 방향으로 컨볼루션과 데시메이션을 수행한 후, 결과를 내부 메모리 'B'에 저장한다. 내부 메모리 'B'에 저장된 픽셀들은 'Output-FIFO' 버퍼로 이동된 후, DMA에 의해서 최종적으로 SRAM에 저장된다. 이런 연산 과정은 목표 단계인 N번 반복되는데, 반복횟수는 영상의 크기와 패턴 탐색을 위한 부영상의 크기에 따라 가변적으로 정해진다.

그림 5는 전체 영상에서 패턴 탐색 대상이 되는  $w \times w$  영역 ( $w$  : 윈도우 크기)에 대해서 LIC 및 HE를 수행하는 영상개선 (Image Enhancement, IE) 블록의 내부 구조를 보여준다. 'Host interface'는 그림 4의 피라미드 연산 블록 내부에 있는 것과 같은 회로이며, DMA를 통하여 외부 메모리에 있는 영상데이터를 Read/Write하는 기능을 수행한다. 'Source DMA' 블록은 먼저 외부 메모리로부터 탐색에 사용할  $w \times w$  영역을 읽어온 후, 내부의 입력버퍼 (2K X8 bits)에 저장한다. 버퍼에 픽셀들은 저장되면 실제 LIC 및 HE 연산을 담당하는 'Preprocessing' 모듈에 전달된다. 효과적인 영상 입출력 수행을 위하여 IE 블록에는 2개의 내부 버퍼를 구현한다. 먼저 하나의 버퍼에 데이터가 들어온 후, 'Preprocessing' 모듈을 통하여 전처리가 이루어지고 있는 동안, DMA는 새로운 탐색 영역을 다른 버퍼에 저장하는 파이프라인 방식을 처리를 위한 설계로써, 전체 수행시간을 줄이는 역할을 한다. 전처리가 완료된 결과는 CPU에서 사용이 가능하도록 'Destination DMA'에 위해서 외부 메모리에 저장된다. 그림 6에서는 LIC와 HE의 데이터 경로 및 제어신호를 포함한 RTL 블록도를 보여준다.

구현되는 SoC를 운영하기 위해서는 소프트웨어가 필요하다. 또한 SoC에 새로운 기능을 소프트웨어적으로 추가할 경우, 이를 개발할 수 있는 환경을 만들어 주어야 한다. 이런 점을 고려하여 개발 SoC에 맞는 최적화된 운영 환경과 소프트웨어 개발 환경을 동시에 지원하도록 하기 위해 그림 7과 같은 부터로더 기반의 소프트웨어 개발체계를 구축한다.

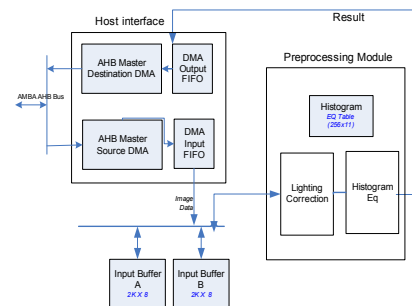
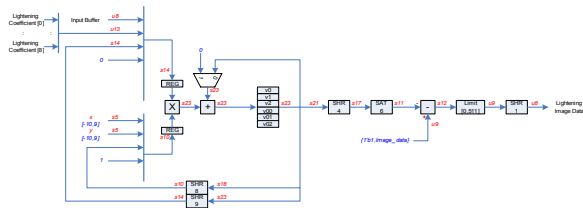
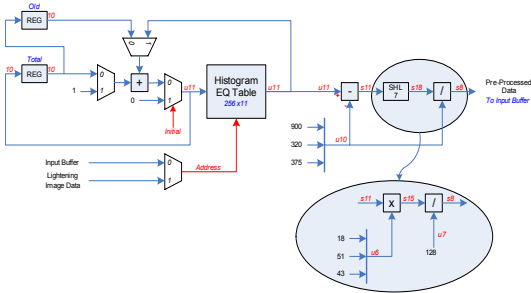


그림 5 영상개선 블록도  
Fig. 5 The block diagram of the image enhancement



(a) LIC 회로도



(b) HE 회로도

그림 6 LIC와 HE의 RTL 블록도  
Fig. 6 RTL diagrams of LIC and HE

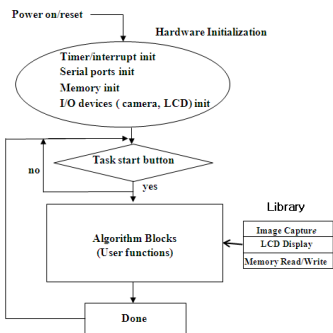


그림 7 응용 시스템 흐름도  
Fig. 7 The flowchart of the application system

#### 4. 구현 및 검증

목표 SoC에 포함되는 모든 하드웨어 요소들은 VHDL로 구현하였으며, XILINX사의 개발 툴을 이용하여 FPGA에 매핑 (Mapping)된다. 이런 과정을 통하여 LEON2 코어, 버스 시스템 그리고 설계된 영상처리용 하드웨어 모듈을 모두 단일 FPGA상에 구현되었기 때문에, 단일 칩에 구현된 여러 하드웨어의 연계 동작과정을 검증할 수 있다. 실제적인 유용성 및 효율성 평가를 위하여 패턴인식 관련 응용 시스템을 직접 SoC상에서 구현하고 실행한다.

구현할 영상 응용 시스템은 그림 8에서 보듯이 3개의 주요 스텝 (steps)으로 구성된다. 먼저 이미지 센서 (마이크론사의 MT9V112 [13])를 이용하여 다양한 크기의 차량패턴을 1개 이상 포함한 320X240크기의 영상을 획득한다 (그림 8). 입력된 영상에 대해서 13단계 피라미드 구성이 끝나면 피라미드내의 각각의 영상들에 대해서 차량 패턴을 찾는 과정이

진행된다. 먼저 입력되는 피라미드 영상에서 30X30 크기의 부영상을 추출한 후, 추출된 부영상에 대하여 LIC와 HE를 수행하여 얻어진 결과를 소프트웨어로 구현된 공분산 기반 Template 매칭기법 [14]을 통하여 차량 패턴인지 아닌지를 판별한다. LIC와 HE가 인식 성능에 미치는 영향을 알아보기 위해 추출된 부영상을 LIC와 HE 처리하지 않고 매칭하는 실험을 함께 진행한다. 피라미드를 적용하지 않는 실험은 고려하지 않는다. 그 이유는 크기 변이를 소화할 없기 때문에 상호 비교가 불가능하기 때문이다.

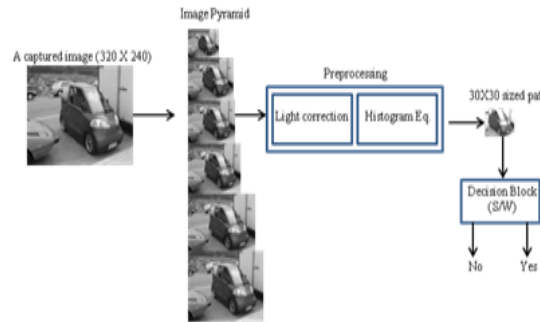


그림 8 구현 시스템 사양  
Fig. 8 The specification of the implemented application

크기가 다른 자동차 영상들을 포함한 30개의 320X240영상을 대상으로 한 실험결과가 표 2에 나타나 있다. 실험 결과 LIC 및 HE를 하지 않은 경우에는 대상 자동차 영상 65개 중 32개만을 검출하였다. 그러나 LIC 및 HE를 한 경우에는 61개의 대상을 정확히 검출하는 결과를 보였다. 이런 결과를 통하여 실제 영상관련 응용의 인식 및 검출 성능을 높이기 위해서는 영상의 개선이 반드시 필요함을 확인함으로써 본 연구에서 구현한 전처리가 응용 시스템 구축에 유용하게 사용될 수 있음을 알 수 있다.

표 2 검출율 비교표

Table 2 The comparison of detection rates

전처리 실행 방법	검출갯수	검출율(%)
Pyramid LIC HE 모두 사용	61	93
Pyramid만 사용	32	49

검출율과 더불어 구현된 SoC에 대한 성능 척도로 사용한 것은 실행속도이다. 구현된 시스템을 이용하여 320X240 크기의 입력 영상에 대해서 각 모듈별로 실행시간을 측정된 결과가 표 3에 나타나 있다. 최종 소요되는 시간은 수식 (3)에 의해서 계산할 수 있다. 수식에서  $P_{time}$ 은 영상 피라미드를 구성하는데 필요한 시간을 나타내며,  $L_{time}$ 과  $H_{time}$ 은 빛 보정 및 HE에 소요시간을 나타내며  $M_{regions}$ 은 실제 탐색이 되는 30X30 크기의 영역 수를 나타낸다.  $D_{time}$ 은 판단 루틴의 수행시간을 나타내며  $I_{time}$ 은 영상입력 소요시간을 나타낸다.

$$P_{time} + M_{regions} \times (L_{time} + H_{time} + I_{time} + D_{time}) \quad (3)$$

**표 3** 구성요소별 소요시간

**Table 3** Required times of processing modules

알고리즘 단계	구현	소요시간(msec)
Pyramid구축( $P_{time}$ )	H/W	160
30X30영상 로딩( $I_{time}$ )	H/W	0.0127
빛보정 ( $L_{time}$ )	H/W	0.0911
HE ( $H_{time}$ )	H/W	0.0521
Decison ( $D_{time}$ )	S/W	0.129
대상 영역수 ( $M_{regions}$ )		2569개
총 소요 시간		892

계산 결과, SoC에 구현된 시스템은 320X240 영상을 처리하는데 0.892초를 소요하였다. 하드웨어화로 인한 성능향상 정도를 측정하기 위해 LEON2 CPU에 전체 알고리즘을 순수 소프트웨어로 구현하고 동일한 작업을 실행한 결과는 191.5초가 걸렸다. 이 비교실험을 통하여 3가지 전처리의 하드웨어 구현으로 기존의 소프트웨어로의 구현보다 221배의 속도향상이 이루어 졌음을 확인하였다.

### 5. 결 론

본 논문에서는 낮은 하드웨어 사양을 가지는 스마트 디바이스에 영상인식 기반 응용 시스템을 효과적으로 구현할 수 있는 SoC를 제안하고 설계하였다. 또한 실제 개인용/모바일 기기에 활용이 가능한 ‘스마트’ 응용 중 하나인 패턴검출 시스템 전체를 SoC에 구현하고 성능을 측정하는 실험을 통하여 구현된 SoC가 하드웨어를 통한 알고리즘의 계산시간 감소 효과와 영상관련 응용에 필수적인 전처리 기능의 제공을 동시에 달성하였음을 보였다.

### 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] H. Nakajima, Y. Matsuo, M. Nagata and K. Saito, "Portable Translator capable of Recognizing Characters on Signboard and Menu Captured by built-in camera," Proc. of the ACL Interactive Poster and Demonstration Sessions, pp. 61 - 64, June, 2005.

[5] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," IEEE Transactions on communications, vol. 31, no.4, pp. 532 - 540, April, 1983.

[6] Kah-Kay Sung. Learning and Example Selection for Object and Pattern Detection. PhD thesis, MIT AI Lab, January 1996. Available as AI Technical Report 1572.

[7] J. Y. Kim, L. S. Kim, S. H. Hwang, "An Advanced Contrast Enhancement Using Partially Overlapped Sub-Block Histogram Equalization," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no.4, pp. 475-484, 2001.

[8] P. G. D. Valle, D. Atienza, G. Paci and F. Poletti, "Application of FPGA Emulation to SoC Floorplan and Packaging Exploration," XXII Conference on Design of Circuits and Integrated System, pp. 236 - 240.

[9] Theocharides, G. Link, N.Vijaykrishnan, M. J. Irwinand W. Wolf, "Embedded Hardware Face Detection", Proceedings of th 17thInternational Conference on VLSI Design (VLSID'04), Jan., 2004.

[10] O. Sims and J. Irvine, "An FPGA implementation of pattern-selective pyramidal image fusion," Proceedings of 2006 international conference of field programmable logic and application, Madrid, Spain, August 2006.

[11] Z. Salcic, J. Sivaswamy, "IMECO: A Reconfigurable FPGA-based Image Enhancement Co-Processor Framework," Real-Time Imaging, vol. 5, no. 6, pp. 385~395, 1999.

[12] LEON2 processor user's manual, Gaisler Research, <http://www.gaisler.com>

[13] MT9V112 manual, Micron Technology Inc., <http://www.micron.com>

[14] A. Kumar and B. Yegnanaarayana, "Template Matching Approach for Pose Problem in Face Verification," MRCS 2006, LNCS 4105, pp. 191-198, 2006

### 저 자 소 개



#### 양근탁 (梁根倬)

E-mail : doongul@gmail.com  
 제주대학교 전산통계학과 박사과정  
 관심분야 : 패턴인식, SoC



#### 이봉규 (李鳳奎)

제주대학교 전산통계학과 교수  
 Tel : (064)754-3593  
 E-mail : bklee@venus1.cheju.ac.kr  
 관심분야 : 영상처리 SoC 설계