

## A SoC Based on a Neural Network for Embedded Smart Applications

이 봉 규\*  
(Bongkyu Lee)

**Abstract** - This paper presents a programmable System-On-a-chip (SoC) for various embedded smart applications that need Neural Network computations. The system is fully implemented into a prototyping platform based on Field Programmable Gate Array (FPGA). The SoC consists of an embedded processor core and a reconfigurable hardware accelerator for neural computations. The performance of the SoC is evaluated using a real image processing application, an optical character recognition (OCR) system.

**Key Words** : SoC (System-on-a-chip), FPGA, Neural Networks

### 1. 서 론

휴대폰이나 개인정보단말기(PDA)와 같은 개인용 이동기에 카메라(영상입력 장치)가 보편화되면서, 기존에 데스크탑 PC나 서버 등에서 이루어지던 영상처리/인식 등의 '스마트' 응용시스템을 임베디드 기기에 구현하려는 요구가 증가하고 있다[1]. 이런 '스마트' 디바이스는 주변의 상황에 맞추어 개별화되고 다양한 서비스를 제공하기 위한 주변 상황의 센싱 (Context sensing)과 인지기능 (Context awareness)을 제공할 수 있는 방향으로 발전하고 있다. 인공 신경망 (Artificial Neural Networks, ANN)은 학습 및 구조적인 특징으로 인하여 '스마트' 디바이스의 개발에 이상적인 계산 모델로 인식되고 있으며, 특히 다층신경망 (Multilayer Perceptron, MLP)은 가장 폭넓게 쓰이는 모델로써 광학문자 인식 (Optical Character Recognition, OCR), 데이터 마이닝 (data mining), 영상 처리/인식 등의 '스마트' 시스템 설계에 성공적으로 적용되고 있다 [2]. 그러나 이런 장점에도 불구하고 실제 MLP를 기반으로 한 '스마트' 응용을 개인용 기기와 같은 임베디드 환경에 구현하는 것은 한계를 보이고 있다. 그 이유는 MLP의 경우 많은 계산량을 요구하기 때문에 계산능력에 한계를 가지는 임베디드 하드웨어에는 데스크톱 PC와 같은 S/W 형태로의 구현이 힘들기 때문이다. 이런 이유로 MLP모델을 H/W로 구현하여 임베디드 시스템에 적용하는 연구가 많이 진행되고 있으며 가장 대표적인 방법으로는 ASIC 디자인 기법을 통한 완전 H/W 구현이 있다 [3,4,5].

그러나 순수 H/W로의 구현 방법은 구현 비용이나 복잡도

측면에서 효율적이지 못하고 구조의 변경이 어려운 단점이 있다. 이에 따라서 최근에는 재구성 가능한 FPGA를 통한 구현 방법이 주목을 받고 있다. 이 방법을 통하여 많은 구현 사례가 최근에 발표되고 있다 [6,7,8,9]. 이 방법을 통하여 MLP에 대한 H/W적인 구현방법에 있어서 가장 중요한 파라미터인 정확도, 메모리 요구량과 계산속도 등을 동시에 고려한 설계가 가능해지고 있는 것이다. 그러나 현재까지의 재구성 가능 디바이스에 기반을 둔 MLP H/W 설계방법은 2가지의 문제점이 있다. 일반적으로 MLP의 경우, 적용하려는 응용에 따라서 각기 다른 전처리 및 후처리 과정과 결합하여 기능을 수행하는 것이 일반적이기 때문에 구현 H/W는 동적으로 재구성이 가능해야 한다. 또한 다양한 MLP의 구조를 H/W에 대한 재 디자인 없이 구현된 H/W에서 바로 수용할 수 있는 구조를 가지는 것이 바람직하며 실제 응용에서 필수적인 요소이다 [10].

본 논문에서는 한정된 자원을 가지는 개인용/모바일 임베디드 시스템에 신경망을 이용한 영상처리/인식 관련 '스마트' 응용을 구현하는데 사용이 가능한 단일 칩 형태의 MLP\_SoC를 Co-design [11] 방법으로 설계한다. 일반적으로 MLP를 이용한 영상 처리/인식은 전처리, 신경망 연산과 후처리 3단계로 구성이 된다. 이들 중에서 전처리 및 후처리는 응용분야에 따라서 달라지고 많은 계산량을 필요로 하지 않은 반면, 신경망 연산의 경우는 응용에 따른 변화가 없는 부분이면서 계산량이 많다. 이런 점을 고려하여 신경망 연산 단계는 H/W로 구성하고, 나머지 모듈은 S/W로 구현할 수 있는 형태의 MLP\_SoC를 설계한다. 제안하는 SoC에 대한 Register Transfer Level (RTL)에서의 테스트와 디버깅을 위하여 FPGA에 기반을 둔 새로운 형태의 프로토타입 플랫폼을 구현하여 사용한다. 개발 플랫폼에서 사용하는 메인 프로세서 및 영상관련 H/W 컨트롤러는 기존 IP를 활용하여 FPGA에 구현한다. 신경망 관련 H/W 역시 같은 FPGA내에 구현되기 때문에 모든 하드웨어 블록들을 설계

\* 종신회원 : 제주대학교 전산통계학과  
E-mail : bkleee@cheju.ac.kr  
접수일자 : 2009년 7월 11일  
최종완료 : 2009년 9월 7일

단계에서부터 통합 검증할 수 있는 새로운 플랫폼이다. 구현된 SoC 구성 블록들은 합성 툴(Synthesize tool)에 의해서 개발 플랫폼에 있는 FPGA에서 실현되어 Register Transfer Level (RTL)에서 검증된다. 구현된 플랫폼에 대한 개발 환경 구축을 위하여 플랫폼에 대한 초기화 및 구동용 모니터 프로그램을 함께 구축한다. 실험에서는 설계/구현된 SoC에 실제 OCR (Optical Character Recognition) 시스템을 구현하는 실험을 통하여 제안하는 SoC의 유효성을 보인다.

2. 설계 조건 및 개발 플랫폼 구현

본 장에서는 MLP\_SoC를 효과적으로 구현하기 위한 플랫폼 활용방법, H/W구현을 위한 데이터 정밀도 및 표현방법과 SoC를 구성할 하드웨어 요소 등에 대해서 기술한다.

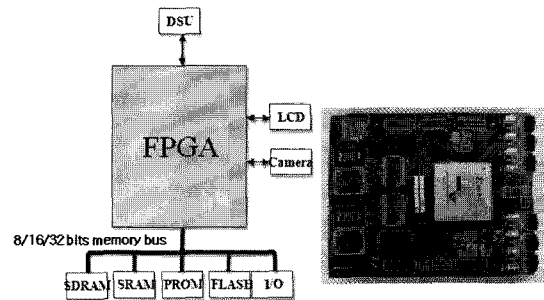
2.1 개발 플랫폼

FPGA에 기반을 둔 플랫폼은 타깃 SoC에 대한 빠른 프로토타입 제작과 테스트에 일반적으로 널리 사용되는 기본적인 요소로서 SoC의 모든 디자인 요소들을 FPGA에 실현하여 테스트해 봄으로써, 정확하고 빠른 구현에 활용 가능하다 [12]. 현재 사용되고 있는 개발 플랫폼은 프로세서가 장착되고 메모리, 주변 제어 장치 및 관련 소프트웨어를 가지는 구조이다. 그러나 이런 형태의 고정된 형태의 플랫폼은 본 연구에서 목표로 하는 CPU를 내부에 포함하는 단일 칩 형태의 SoC를 위한 개발에는 적합하지 못하다. 또한 SoC의 설계에 기존의 다양한 IP를 활용하기 위해서는 사용할 IP를 구현되는 SoC에 쉽게 포함시킬 수 있어야 한다. 따라서 본 연구에서는 기존의 IP들을 전체 설계단계에서 필요할 때 효과적으로 사용하기 위하여 FPGA에 기반을 둔 새로운 형태의 플랫폼을 제작하여 사용한다. 개발 플랫폼의 FPGA에는 구현될 MLP뿐만 아니라 프로세서 및 기본 하드웨어 블록들이 모두 포함되기 때문에 설계부터 단일 칩 영역의 형태로 개발 및 검증이 가능한 장점이 있다. FPGA에 MLP관련 부분 외에 구현되는 요소는 CPU, 버스 시스템, 외부 센서제어용 콘트롤러 등이다.

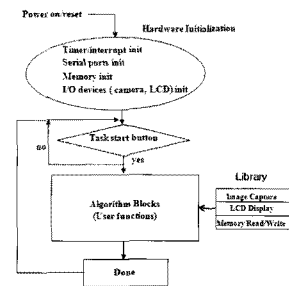
목표 SoC의 메인(main) CPU로는 LEON 2를 [13] 채택하였으며, CPU와 다른 H/W 요소들 간의 연결을 위하여 온칩 (on-chip) AHB/APB AMBA 버스 시스템을 FPGA내부에 구현한다. 또한 영상 응용을 위하여 외부에 연결될 영상센서를 작동할 수 있는 카메라 인터페이스가 구현된다. 이러한 모든 요소들은 실제 신경망 연산을 담당할 MLP H/W와 같은 FPGA 상에 구현이 되기 때문에 MLP H/W를 구축한 후, 전체 SoC에 대하여 하나의 칩에서 동작하는 것과 같은 검증은 신속하고 정확하게 할 수 있다.

플랫폼에는 FPGA이외에도 128Mbytes의 SDRAM기반 외부 메모리와 8 Mbytes Flash 메모리를 외부에 가지고 있다. SDRAM기반 메모리는 외부에서 들어오는 데이터를 저장하기 위한 것이며, 플래시 메모리는 플랫폼을 운영하는 모니터 프로그램과 MLP S/W를 저장하는 기능을 수행한다.

그림 1에서는 실제 구현이 완료된 프로토타이핑 플랫폼과 해당 플랫폼에 대한 소프트웨어 개발 환경인 모니터 프로그램의 흐름을 보여주고 있다. 플로차트에 나타난 블록 중에서 사용자 함수 블록에 MLP를 구성하는 S/W 모듈이 삽입된다. 표 1에서는 플랫폼의 주요 구성요소를 보여준다.



(a) 프로토타이핑 플랫폼



(b) 운영프로그램 흐름도

그림 1 프로토타이핑 플랫폼 및 운영프로그램 구조  
Fig. 1 The prototyping platform and monitor program

표 1 구현된 개발플랫폼의 주요 특징

Table 1 Major components of the platform

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

2.2 영상처리/인식을 위한 MLP 설계 고려사항

영상 처리/인식 응용에 적용되는 MLP는 층 (Layers) 별로 구분되는 많은 수의 처리요소인 뉴런 (neuron)들로 구성된다. 일반적으로 MLP는 하나의 입력 층과 하나의 출력 층 그리고 하나 이상의 은닉층 (Hidden layer) 등 3개 이상의 층으로 구성되며 각 층의 뉴런들은 자신이 속한 층의 아래층 및 위층에 있는 뉴런들과 연결 (fully/partially connected) 되어있다. 이런 층간의 모든 연결에는 서로 다른 가중치 (weights)가 주어져 있으며, 이 가중치에 의해서 특정 MLP의 동작이 정의된다. MLP는 대상 문제에 대해서

적절한 가중치를 찾음으로써 해당 문제에 대한 해결 능력을 가지게 되는데 이런 가중치를 찾는 것을 학습이라 한다. 일반적으로 MLP는 오류 역전파 학습규칙 (Error Backpropagation)에 의하여 학습되어 진다 [2].

특정 대상에 대해서 학습이 완료된 가중치를 이용한 인식과정은 외부로부터 얻어지는 입력 영상이 입력 층의 뉴런에 들어오면서 시작된다. 입력 층에 들어온 영상은 은닉 층과 출력 층의 방향으로 진행하면서 각 층에서의 출력 값을 계산한다. 수식 (1)에서는 은닉/출력 층에 있는 뉴런  $i$ 가 자신과 연결된  $n$ 개의 하위 층 노드들로부터 받는 입력들에 대한 가중치 합을 구하는 과정을 보여준다.  $w_{ij}$ 는 뉴런  $i, j$  간의 가중치를 나타내며  $out_j$ 는 하위 층의 뉴런  $j$ 로부터 들어오는 값을 나타낸다. 최종적인 노드  $i$ 의 출력값은 (1)에서 얻어진 값에 시그모이드(sigmoid)와 같은 비선형 함수  $f$ 를 처리하여 수식 (2)와 같이 계산된다.

$$input_i = \sum_{j=1}^n w_{ij} out_j \quad (1)$$

$$out_i = f(input_i) \quad (2)$$

신경망연산에 사용되는 중요 데이터(가중치, 입력값, 출력값)는 대부분 부동소수점 (float point)으로 처리된다. 따라서 별도의 FPU (Floating Point Unit)이 없는 임베디드 환경을 위해서는 부동소수점 연산을 고정소수점 (fixed point) 연산으로 바꾸어야 한다. 본 연구에서도 입력/출력 값과 가중치에 대한 표현을 고정소수점으로 구현한다. 또한 출력값 계산에 사용되는 활성화함수를 테이블 룩업 (Table LookUp) 방식으로 구현하여 H/W 오버헤드를 줄이면서, 다양한 활성화함수를 표현할 수 있도록 한다. 표 2에서는 이런 설계를 만족시키기 위하여 정해진 값을 보여준다.

표 2 구현된 개발 플랫폼의 주요 특징

Table 2 Major components of the platform

| 주요 파라미터             | 설정 값          |
|---------------------|---------------|
| 최대 구현가능 입력 노드 수     | 1,000         |
| 최대 구현가능 은닉 노드 수 및 층 | 128/2         |
| 가중치 표현 비트 수         | 12비트 (signed) |
| 활성함수 출력 비트 수        | 9비트 (signed)  |
| 입력 데이터 값 범위         | 0 ~ 255       |
| 출력값의 범위             | -255 ~ 255    |

### 3. MLP\_SoC

이동형 기기에 영상 처리/인식 기능을 내장하는데 효과적으로 사용될 수 있는 신경망 기반 SoC의 단일 칩 구현을 위해서는 소프트웨어 수행을 위한 기본 CPU 코어, 영상의 입력 및 출력 기능을 담당하는 디바이스 제어 블록, 내부 버스 인터페이스 등을 포함하는 구조를 가져야 한다. 이러한 요소들을 고려하여 본 연구에서 구현하려는 LEON 2 기반의 영상처리용 SoC의 설계 및 구현 과정을 기술한다.

#### 3.1 목표 SoC의 하드웨어 구조

LEON 2를 중심으로 설계한 MLP\_SoC의 블록 구조가 그림 2에 나타나있다. LEON 2의 코어부분은 Gaister

Research Group [13]에서 제공하는 VHDL 소스를 기반으로 구현하였다. 내부에 구현된 다른 하드웨어에는 버스시스템을 위한 AHB-APB 브리지(Bridge), 카메라 인터페이스 컨트롤러 및 메모리 컨트롤러가 있다. 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 연결에 사용되는데 I<sup>2</sup>C 마스터는 외부 카메라와의 인터페이스에 사용된다. MLP\_SoC의 가장 핵심적인 부분인 신경망 연산을 담당하는 'MLP Computation Coprocessor'의 내부 구조는 다음 절에서 상세히 기술한다.

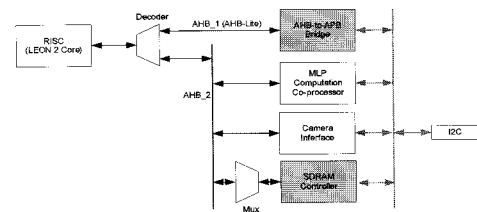


그림 2 구현된 내부 구조도

Fig. 2 The block diagram of the implemented architecture

#### 3.2 MLP Computation Coprocessor (MLPCC) 블록

설계된 MLPCC의 내부구조가 그림 3에 나타나 있다. MLPCC는 메모리 접근 및 버스 인터페이스에 사용되는 'Host interface' 부분과 실제 신경망 계산을 담당하는 'MLP' 모듈로 구성된다. 'Host interface' 블록은 MLPCC와 다른 컨트롤러 사이의 인터페이스를 담당하는 회로로서 직접 메모리 접근 (DMA) 관련 회로인 'Source DMA'와 'Destination DMA'로 구성된다. DMA의 사용은 MLPCC가 CPU를 거치지 않고 직접 메모리에 접근이 가능토록 한다. 'Source DMA' 블록은 외부 SDRAM메모리에 있는 영상 데이터를 MLPCC 내부에 있는 2개 입력 버퍼 (2K X8 bits) 중 하나로 가지고 오는 기능을 수행한다. 입력버퍼가 2개로 구성되는 이유는 하나의 버퍼에 들어온 입력을 신경망이 처리하는 동안 다른 버퍼에 새로운 입력을 미리 읽어 놓음으로써 성능을 향상시킬 수 있기 때문이다. 'Source DMA' 블록은 버퍼에 입력이 완료되면 신경망 연산을 요구하는 신호를 연산 담당블록에 준다. 버퍼의 입력에 대한 신경망연산 결과는 'Destination DMA' 블록이 SDRAM에 저장한다.

실제 신경망 연산을 담당하는 'MLP' 블록은 내부의 데이터 저장용 레지스터 집합과 계산 모듈로 구성이 된다. 데이터 저장용 메모리는 활성화함수 테이블, 은닉 노드 저장 파일과 가중치 테이블로 구성이 된다. 테이블을 이용한 활성화함수의 구현은 하드웨어 변경없이 다양한 함수를 MLP\_SoC에 구현하는 것을 가능토록 한다. 표 1에 나타난 설계조건을 만족하기 위하여 가중치 테이블은 128K \* 19비트의 크기로 구현되며, 19비트 중에서 12비트는

가중치를 고정소수점으로 표현하는데 사용되고 나머지 7비트는 가중치에 관련된 노드의 인덱스를 표현한다. 은닉 노드 저장파일은 은닉 층과 출력 층 노드들의 임시값을 저장하기 위한 공간으로 표 1의 설계조건을 고려하여 128 \*24비트로 구성된다.

그림 4는 MLPCC에서 연산을 담당하는 'Computation Module'의 Register Transfer Level (RTL) 블록도와 데이터의 하드웨어 표현을 보여준다. 이 모듈은 입력버퍼 ('A' 혹은 'B')에 있는 데이터를 입력 층에 있는 노드로 가져와서 최종 출력이 계산될 때까지 각 층의 뉴런 출력값을 계산한다. 계산된 출력 뉴런의 값들은 앞 절에서 기술한 것과 같이 SDRAM에 저장된다.

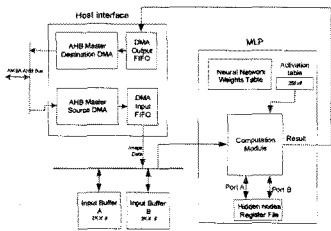


그림 3 MLP연산을 위한 H/W 구조  
Fig. 3 Architectural overview of the MLPCC

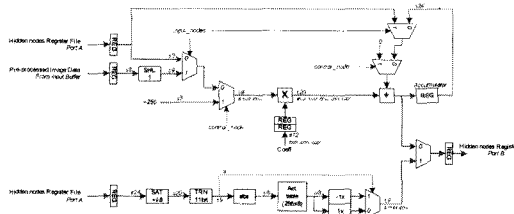


그림 4 'Computation Module'의 RTL 블록도  
Fig. 4 The RTL diagram of the computation module

4. 구현 및 검증

설계된 하드웨어 요소들은 VHDL로 구현하였으며, XILINX 툴을 이용하여 FPGA (XILINX X2CV8000)에 구현된다. 구현된 SoC의 동작을 검증하기 위하여 영상 인식 관련 응용인 OCR 시스템을 직접 SoC에 구현하고 검증한다.

OCR은 컴퓨터로 하여금 디지털 문자 이미지를 텍스트 형태로 변환하는 과정을 수행토록 하는 영상 인식시스템의 한 종류로서 휴대용 번역기 (portable translators), 전자사전 (electronic dictionaries) 및 PDA (Personal Data Assistant) 등에 적용이 가능하다 [14]. 구현할 OCR 시스템은 그림 5에서 보듯이 3개의 단계로 구성된다. 이미지 센서 (마이크로사의 MT9V112 [15] 센서를 사용)를 이용하여 인식 대상이 되는 320X240크기의 영상 2개 (720 문자, 대/소 타임스 뉴로만체)를 획득한다. 이 영상들에 포함된 각각의 문자들은 문자단위로 분리되어 전처리 (왜곡, 크기)를 통하여 신경망의 입력 벡터 (30X24 크기)로 정규화 된다. 이 입력벡터를 학습된 가중치를 가지는 MLP\_SoC가 입력받아 인식을 수행한다. 구현된 SoC가 H/W 변경없이 다양한 형태의 MLP를

구현할 수 있음을 보이기 위하여 먼저 은닉 노드의 수가 각각 20개/32개인 MLP를 각기 구현한 후, 학습 데이터를 통하여 인식실험을 수행하였다. 그 결과 은닉노드가 32개인 MLP가 좋은 인식율을 보였기 때문에 OCR시스템에는 32개의 은닉노드로 구성되는 MLP를 사용하였다. 실제 테스트 데이터를 통한 실험결과 구현된 OCR시스템은 720개의 분리된 알파벳 영문자 중 716개를 정확히 인식하는 결과를 보였다. 표 3에는 구현된 OCR시스템을 위하여 사용된 MLP의 사양과 최종 인식결과가 나타나 있다.

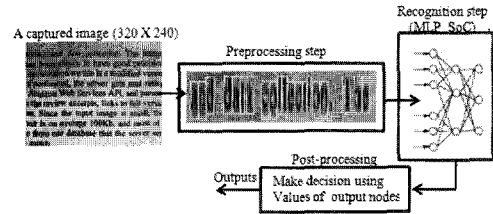


그림 5 구현된 OCR 흐름도  
Fig. 5 The processing flow of the OCR system

표 3 사용된 MLP의 사양  
Table 3 Configurations of the implemented MLP

| 주요 파라미터       | 설정 값      |
|---------------|-----------|
| 입력/은닉/출력 노드 수 | 720/32/26 |
| 활성함수          | 시그모이드     |
| 층간 연결방식       | 전체 연결     |
| 학습규칙          | 오류 역전파    |
| 인식율           | 96%       |

인식율과 더불어 구현된 SoC에 대한 성능 척도로 사용한 것은 인식 속도이다. 구현된 OCR시스템을 이용하여 320X240크기의 260개의 문자를 가지는 영상이미지에 대해서 각 모듈별로 실행시간을 측정한 결과가 표 4에 나타나 있다. 구현된 OCR시스템은 6.03초 (초당 43개의 문자영상 처리)에 작업을 완료하였으며, 전체 소요시간 중에서 MLP H/W가 사용한 시간은 3.9초로 나타났다. 비교를 위하여 LEON2 CPU에 같은 형태의 MLP를 S/W적으로 구현하고 동일한 작업을 실행한 경우에는 신경망 연산에 869초가 소요되었다. 비교실험을 통하여 MLP H/W를 사용할 경우 기존의 S/W적인 구현에 비교하여 200배 이상의 속도향상이 가능해진 것으로 판명되었다. 이런 결과는 구현된 MLP\_SoC가 임베디드 시스템에 실시간 영상 처리/인식 관련 '스마트' 응용을 효과적으로 구현하는데 효과적으로 사용되어질 수 있음을 보여준다.

표 4 OCR 구성요소별 소요시간  
Table 4 Speed of each OCR processing module

| 알고리즘 단계 | 구현 방법       | 소요시간(msec) |
|---------|-------------|------------|
| 문자분리    | S/W (LEON2) | 690        |
| 전처리/정규화 | S/W (LEON2) | 1,240      |
| 신경망 연산  | MLP H/W     | 3,980      |
| 후처리     | S/W (LEON2) | 120        |
| 총 소요시간  |             | 6,030      |

## 5. 결 론

본 논문에서는 작은 크기의 스마트 디바이스에 사용이 적합한 신경망 SoC를 제안하고 설계하였다. 또한 FPGA를 기반으로 한 플랫폼을 이용하여 실제 SoC를 구현한 후, RTL에서 테스트 및 작동을 검증하였다. 제안된 SoC는 H/W의 변경이 없이도 재구성을 통하여 서로 다른 형태를 가지는 MLP를 구현할 수 있음을 실험을 통하여 보였다. 또한 실제 개인용/모바일 기기에 활용이 가능한 '스마트' 응용 중 하나인 OCR 시스템 전체를 단일 SoC로 구현됨을 보임으로써 MLP\_SoC가 지능적 영상 처리/인식을 필요로 하는 다양한 임베디드 시스템에 적용이 가능함을 증명하였다.

## 참 고 문 헌

- [1] Pentland and T. Choudhury, "Face recognition for smart environments," IEEE computer, vol. 33, no. 2, pp. 50 - 55, Feb., 2000.
- [2] J. M. Zurada, *Introduction to Artificial Neural Systems*, PWS publishing company, 1992.
- [3] T. Schoenauer, A. Jahnke, U. Roth and H. Klar, "Digital Neurohardware : Principals and Perspectives," Neural Networks in Applications (NN'98), Magdeburg, pp. 101 - 106, 1998.
- [4] K. Mathia, J. Clark, B. Colbert and R. Saeks, "Benchmarking and MIMD Neural Network Processor," WCNN'96, San Diego, California, Sep., 1996.
- [5] Theocharides, G. Link, N. Vijaykrishnan, M. J. Irwin and W. Wolf, "Embedded Hardware Face Detection", Proceedings of th 17<sup>th</sup> International Conference on VLSI Design (VLSID'04), Jan., 2004.
- [6] 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.
- [7] E. M. Ortigosa, A. Canas, E. Ros, P. M. Ortigosa, S. Mota and J. Diaz, "Hardware description of multi layer perceptrons with different abstraction levels," Microprocessors and Microsystems, vol. 30, pp. 435 - 444, 2006.
- [8] S. Vitabile, V. Conti, F. Gennaro and F. Sorbello, "Efficient MLP Digital Implementation on FPGA," Proceedings of the 8<sup>th</sup> Euromicro conference on DSD, 2005.
- [9] A. Rosado Munoz, E. Soria Olivas, L. Gomez Chova and J. V. Frances, "An IP Core and GUI Implementing Multilayer Perceptron with a Fuzzy Activation Function on Configurable Logic Devices," Journal of Universal Computer Science, vol. 14, no. 10, pp. 1678 - 1694, 2008.
- [10] M. Pormann, M. Franzmeier, H. Kalte, U. Witkowski and U. Ruckert, "A Reconfigurable SOM Hardware Accelerator," ESANN'2002 proceedings, pp. 337 - 342, 2002.
- [11] M. Shabiul, M. S. Beg, M. S. Bhuyan and M. Othman, "Design and Implementation of Discrete Cosine Transform Chip for Digital Consumer Products," IEEE Transaction on Consumer Electronics, vol. 52, no. 3, pp. 998 - 1003, 2006.
- [12] 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, 2007.
- [13] LEON2 processor user's manual, Gaisler Research, <http://www.gaisler.com>
- [14] 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.
- [15] MT9V112 manual, Micron Technology Inc., <http://www.micron.com>

## 저 자 소 개



### 이 봉 규 (李 鳳 奎)

제주대학교 전산통계학과 교수

Tel : (064)754-3593

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

관심분야 : 영상처리 SoC 설계