재구성 가능한 신경회로망의 하드웨어 구현
Hardware Implementation of Neural Network Reconfigurable Architecture

동성수 (디지털전자정보과)
Sung-Soo Dong (Dept. of Digital Electronics & Information)

Key Words : Neural Network(신경회로망), Hardware Implementation(하드웨어 구현).

ABSTRACT : Extension of the synaptic connection is one of the desirable and difficult aspects in hardware implementation of neural networks. This paper proposes a new architecture that processes synaptic computations in separate modules in order to realize scalable architecture. Conventional digital neural networks have to change their design or provide external memory when the number of required synaptic connection exceeds than each neuron holds. By using multiple neurons in cooperative way, the proposed neural network can process synaptic computations in various conditions without altering its original design. Better precisions were achieved in the simulation result and effectiveness of the proposed method was validated.

1. 서론

신경회로망을 하드웨어로 구현하려는 노력은 꾸준히 있어왔고, 최근에는 디지털기술과 ASIC기술의 발전에 힘입어 아날로그 신경회로망보다는 디지털 신경회로망 쪽에 활발한 연구가 있었다.[1] 그러나 디지털 신경회로망의 구현에 있어서 다음과 같은 문제점들이 제기되어 왔다.

1.1 면적의 효율성

디지털 신경회로망을 구현하는 방법은 크게 DSP 또는 CPU를 이용한 마이크로프로세서 기반의 신경망과, 시냅스, 뉴런 등의 회로를 병렬 구성하여 ASIC으로 구현하는 방법이다. 마이크로프로세서 기반의 신경망은 ASIC 방식에 비해 재구성이 용이 하며, 구현이 쉽다는 장점이 있는 반면, 신경망 크기가 커지면 버스와 연산기의 병목현상으로 인해 전체회로망을 구현하기 어려우며, 큰 구현 면적을 차지한다는 단점이 있다. 이를 극복하기 위한 방법으로 곱셈기가 없는 디지털 신경망[2], 혹은 곱셈기의 크기를 줄이기 위한 직렬 데이터 방식의 디지털 신경망[3]과 같이 구조를 그대로 두고 곱셈기의 크기를 줄이는 연구들이 이루어졌다. 그리고 다른 방법으로는 SIMD(Single Instruction Multiple Data) 형태와 같이 버스 구조를 통해서 곱셈기를 재사용하는 디지털 신경망[4]과 사슬 배열 구조(Systolic Array Architecture)를 채용하여 곱셈기의 수를 선형적으로 늘이는 신경회로망[5]과 같은 연구를 통해서 곰셈기의 증가 문제를 해결하는 노력들이 이루어졌다.
SIMD구조는 단일명령 특성으로 인해서 디지털 신경회로망의 하드웨어 구현 방법으로 적합하며 다중프로세서 구조로 인해서 뉴런의 병렬성은 구현하기가 쉬우나, 각각의 프로세스 단위의 문제점 때문에, 하나의 프로세서 단위가 처리할 수 있는 시냅스의 크기는 제한된다. 따라서 시냅스 병렬성은 구현하기가 어렵다. Fig. 1 은 SIMD구조를 이용한 신경회로망의 예를 보여준다.[6]

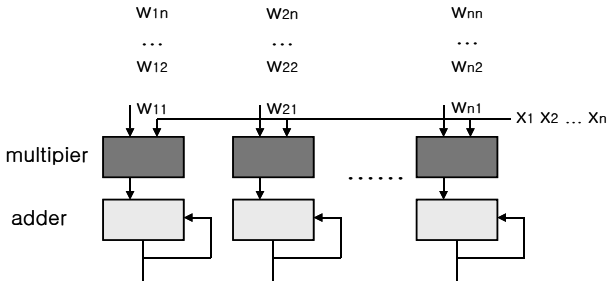


Fig. 1 SIMD architecture

1.2 재구성 능력

신경망 하드웨어의 처리속도 장점을 살리면서 범용성 확보를 위해서는 다양한 종류의 신경회로망을 하드웨어로 재구성 할 수 있어야 한다. 재구성 방법에 대해서 다양한 연구가 있어왔으나 주로 범용 프로세서를 이용한 신경회로망을 재 프로그래밍 하는 방법과[7] FPGA처럼 간단한 재구성 비트를 사용하여 기본 프로세스 소자의 기능과 버스 구조를 변경함으로써 목적에 맞는 신경회로망으로 구성하는 방법[8] 등이 제시되어 왔다.

범용 프로세서를 이용한 방법은 재구성 능력은 우수하지만 앞에서 말한 바와 같이 속도와 면적에서 취약하다는 단점을 가지고 있으며, 반면에 재구성형 하드웨어 신경회로망은 제한된 재구성 능력을 갖지만, 속도가 빠르고 적은 면적에 구현 가능하고 단일 칩으로 제작이 가능하다는 장점을 가지고 있다.

1.3 확장성

확장성 문제에 대한 연구사례로 B. Pino는 각각의 프로세스 소자의 시스템버스를 이웃의 소자 사이만 연결시킴으로써 프로세스 소자를 직렬로 확장하는 방법을 채택했으며, 또한 이를 칩 외부까지 연결시킴으로써 칩 간의 연결도 가능하게 하였다. 이러한 구조는 여러 개의 프로세스 소자를 통해서 뉴런 및 시냅스를 구성함으로써 뉴런, 시냅스의 크기가 프로세스 소자의 한계에 대한 제한을 받지 않는 장점을 가진다. 그러나 인접유닛간의 연결에 의해서만 데이터의 전송이 이루어지므로, 프로세스 소자 간에 데이터의 병목이 생김으로서 느려지는 속도가 전체 시스템 능력을 저하시키는 문제를 야기할 수 있다.

B. Girau는 FPNA라는 개념을 사용해서 FPGA의 이웃간 연결을 이용하여 버스를 구성함으로써 뉴런의 개수를 확장하는 방법을 도입하였다. 이는 뉴런의 개수를 확장하는 데는 유효하지만 각각의 프로세스 소자가 뉴런에 상응할 경우 뉴런 자체의 크기를 확장할 수 없는 단점이 있다.

2. 제안된 구조

2.1 MPU : Modular Processing Unit

재구성 능력과 면적 문제 해결에 초점을 맞추어, 마스터-슬레이브 구조를 응용한 새로운 모듈러 신경망 구조를 제안한다. 대부분의 하드웨어 신경회로망에 있어서 뉴런의 시냅스의 개수는

내부 메모리 크기에 의해서 뉴런의 확장 및 레이어의 확장에 제한을 받게 된다.
이러한 결점을 보완하기 위하여 망을 구성하는 기본단위인 프로세스 단위(PU : process unit)를 뉴런에 대해 확장가능 하도록 재설계 했다. 본 논문에서는 하나의 계층(layer)을 구성하는 한개 또는 여러 개 의 뉴런을 구현할 수 있는 독립된 회로 블록을 모듈러 프로세스 단위(MPU : modular processing unit)이라고 지칭하였다.

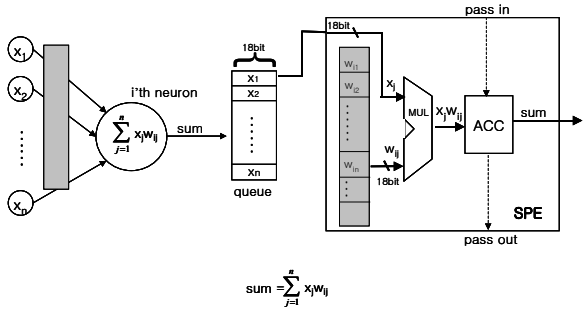


Fig. 2 Neural model and SPE architecture

프로세싱 모듈은 Fig. 2 처럼 기본적으로 입력노드에 대한 가중치를 곱하는 시냅스 기능과 각 시냅스의 누적 합 기능을 가지고 있는 시냅스 프로세스 소자(SPE : synapse processing element)와 누적된 시냅스의 결과에 대한 활성화 함수 값을 출력하거나 MPU간의 데이터 전송 방향을 제어하는 계층 프로세스 소자(LPE : layer processing element)로 구성되어 있다. Fig. 3 은 네 개의 SPE와 한 개의 LPE로 구성된 경우이다.

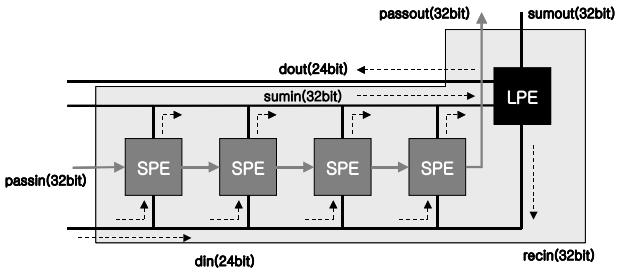
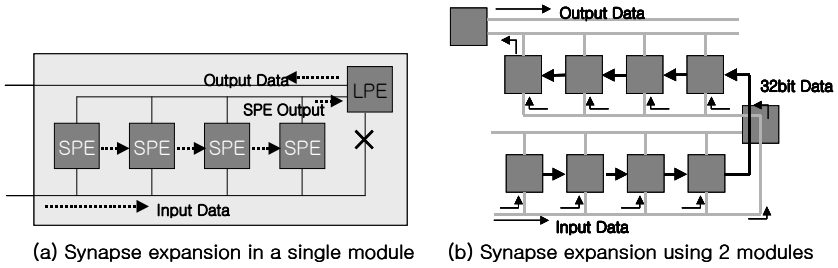


Fig. 3 MPU architecture

기존 신경망의 문제로 나타났던 시냅스 개수가 많은 뉴런의 구현은 내부연결 버스를 사용하여 SPE에서 누적된 시냅스 값을 인접 SPE로 전달하는 방법으로 해결한다. 이것은 단지 MPU 내부 뿐 아니라 MPU 내부에 국한되지 않고 모듈을 초과하는 양의 시냅스도 하나의 뉴런에서 처리하는 것이 가능하다. Fig. 4는 내부 연결버스를 사용해서 시냅스를 확장하는 방법을 보여주고 있다. (a)는 하나의 모듈 내에서 시냅스를 확장하는 형태를 보여주고 있고, (b)는 두개의 모듈을 이용하여 시냅스를 확장하는 방법을 보여주고 있다.



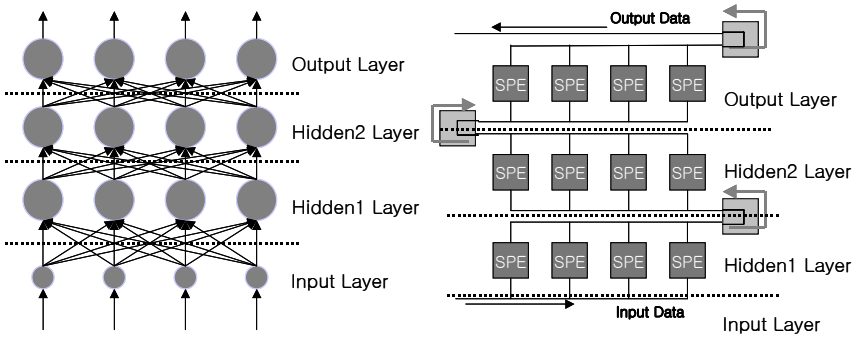
(a) Synapse expansion in a single module (b) Synapse expansion using 2 modules

Fig. 4 Synapse expansion

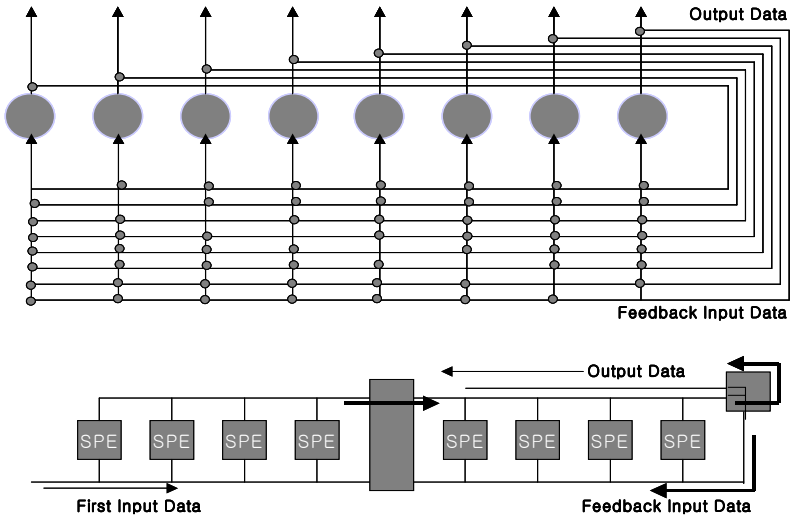
각각의 모듈은 마스터-슬레이브 형태로 연결된다. 뿐만 아니라 재구성 가능한 구조를 사용함으로서 다양한 형태의 신경회로망을 구성하는 것이 가능하다. 이렇게 고안된 구조를 확장성을 가진 재구성 가능한 신경망 구조(ERNA : extendable reconfigurable neural network architecture)라 이름 붙였다.

2.2. 다양한 신경망 구성

MPU로 구성된 신경망은 모듈간의 연결 구조를 바꿈으로써 여러 가지 신경회로망을 구현할 수 있도록 되어 있다. LPE에 저장되는 구성비트에 따라서 버스의 방향 및 활성 함수 (AFU : activation function module)의 기능이 결정된다. 구성 비트의 설정에 의해서 외부 혹은 같은 다른 모듈에서의 입력이 아닌 자신의 모듈, 혹은 같은 계층상에 존재하는 다른 모듈로부터의 출력 값을 입력으로 사용하는 것이 가능하다. 이 기능으로 홉필드(Hopfield) 같은 되먹임 신경망(feedback neural network)을 구현하는 것이 가능하다.
이러한 구성들 가운데, 실제적으로 자주 쓰이는 신경망인 다층 퍼셉트론(multi layer perceptron)에 대한 구성의 예를 Fig. 5(a)에 나타내었고, 마찬가지로 되먹임 신경망의 대표적인 홉필드 네트워크(Hopfield network)를 Fig. 5(b)에 나타내었다.



(a) MLP architecture(4x4x4)



(b) Hopfield network(8 neuron)

Fig. 5 Neural network architecture using ERNA

2.3 신경망 학습

제안한 모듈러 구조는 칩의 크기를 자유롭게 변경하고 칩과 칩 간에도 동일한 학습규칙을 적용해야 하기 때문에 모듈 자체에 학습기능을 내장하고 있지 않으며, 망을 구성할 때는 외부에 별도의 학습기능 칩(on-board learning chip)이 복수의 ERNA 칩으로 구성된 신경망의 학습을 총괄하게 된다. Fig. 6 은 복수개의 칩을 사용하여 망을 구성했을 경우 학습기능 칩과의 연결을 보여준다.

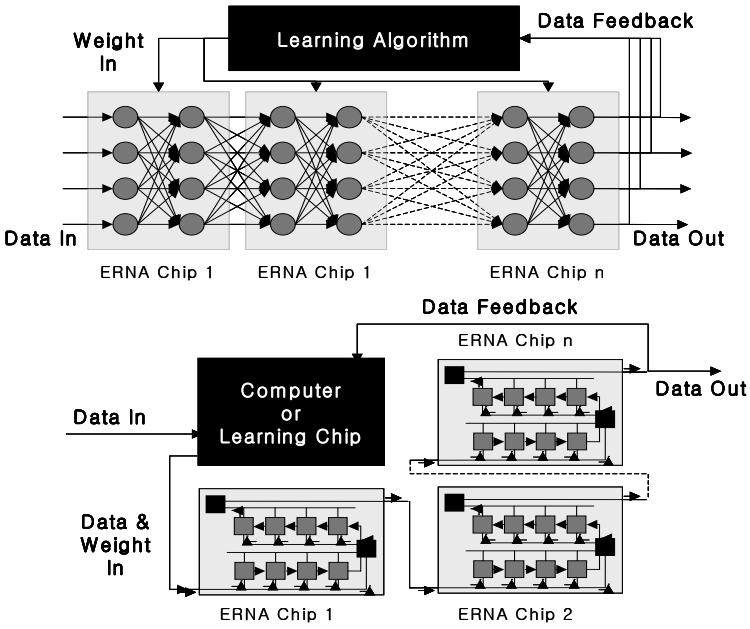


Fig. 6 Configuration with ERNA and learning chip

3. 실험방법

3.1 FPGA 구현

위에 제안된 구조는 검증을 위해 Verilog HDL을 이용하여 설계하고 Leonardo Spectrum을 사용하여 합성하였으며, 시뮬레이션은 Modelsim을 사용했고, FPGA(VertexII 6000)로 구현하였다.

3.2 구성의 정의

구성 형태와 확장 형태를 정의하기 위해서는 먼저 구성 비트에 대해 정의해야 한다. SPE의 경우는 Table 1. 과 같은 구성비트 형식을 사용하고 LPE의 경우는 Table 2. 와 같은 구성비트 형식을 사용한다.

Table 1. Configuration bits of SPE

b22	b21	b20	b19
Enable/ Disable	Inc/ Dec	Start	End

Enable/Disable : enable SPE or disable SPE
Inc/Dec : increment or decrement acc
Start : start block of neuron
End : end block of neuron

Table 2. Configuration bits of LPE

b22	b21
Mode Select	AFU On/Off

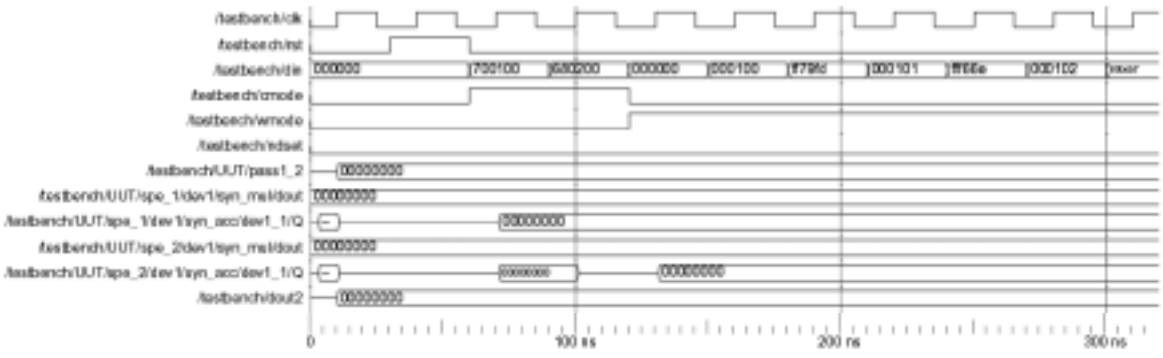
00 : feed forward mode and AFU on
01 : feed forward path route
10 : feedback mode and AFU on
11 : feedback path route

구성 가능한 모듈의 형태는 기본적으로 구성 비트들과 LUT에 의존하게 된다. 그 외의 구성 형태에 미치는 요소는 내부연결 버스 형태와 시냅스의 완전연결, 부분연결 등이 영향을 미치게 된다. 이와 같은 요소에 의해 구성 가능한 형태는 SPE 구성비트(24), LPE 구성비트 (22), 내부연결 버스형태 (21),시냅스의 완전연결, 부분연결 (21)로 최소 28=256가지가 되며, 실제적으로 모듈 내부의 SPE 개수와, 비선형 함수의 출력, 부분 되먹임(feedback)에 의한 TDNN(time delayed neural network) 구성 등에 의해서 수많은 형태의 신경회로망으로 조합이 가능하다.

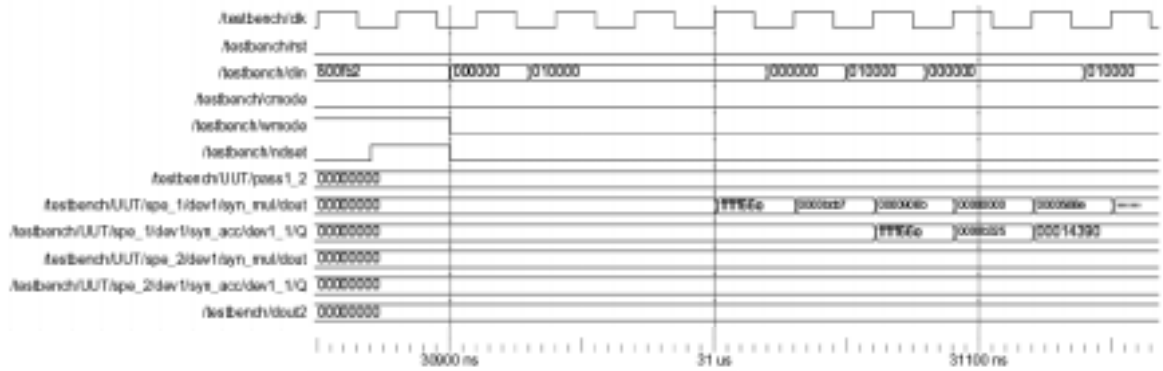
4. 실험결과

4.1 SPE 확장실험

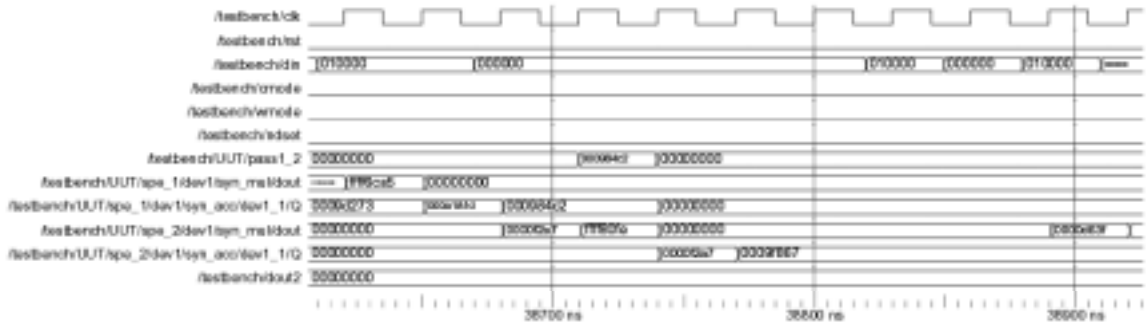
제안된 구조의 확장성을 증명하기 위해서 SPE 두개를 이용한 시냅스의 확장실험을 수행하였다. 실험에서는 두개의 SPE모듈을 연결하여 시냅스를 512개로 확장한 후, 무작위 값을 입력하고 이를 소프트웨어로 예측한 결과와 비교함으로써 그 정확도 및 동작 여부를 증명하였다. Fig. 7은 시냅스 연결에 대한 시뮬레이션 파형이다.



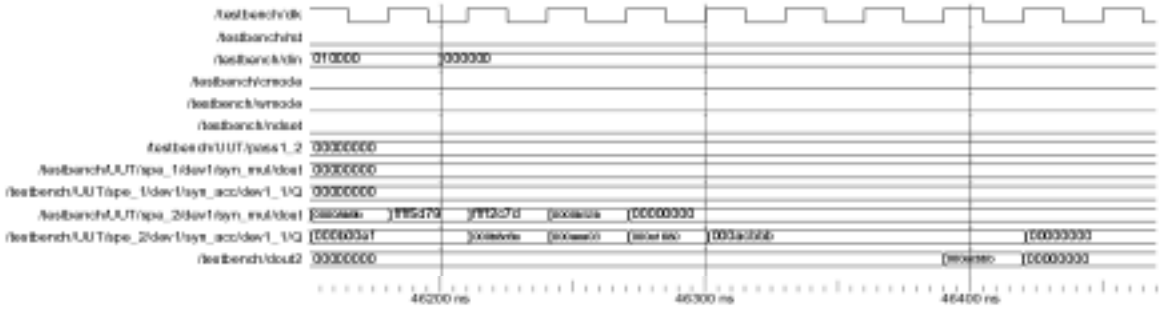
(a) Weight save operation



(b) Working mode



(c) Transfer 1st SPE to 2nd SPE



(d) Result

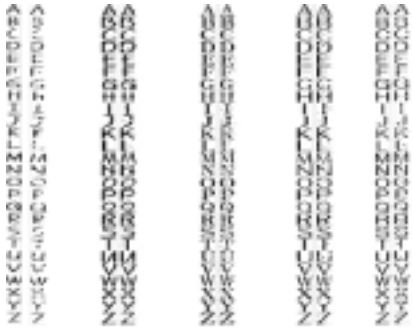
Fig. 7 Synapse expansion using SPE

4.2 정밀도 실험

본 논문에서는 제안된 하드웨어의 유용성을 증명하기 위해서 알파벳 영문자 인식 실험을 수행하였다. 사용된 신경망은 MLP 구조를 사용했으며 학습에는 역전파 알고리즘을 사용하였다. 실험에 사용된 구조에는 학습 모듈이 내장되어 있지 않기 때문에, PC상에서 C언어로 구현된 소프트웨어 시뮬레이터를 통해서 학습된 가중치를 다운로드해서 사용하는 선 학습 방법(pre-trained model)을 사용하였다. Fig. 8은 학습에 사용된 문자 패턴 및 테스트용으로 사용된 손상된 문자 패턴이다. 입력 패턴으로써는 영어 대문자 폰트 26개에 대해서 5가지 폰트를 사용했으며 각각의 왼쪽은 정상 패턴, 오른쪽은 손상된 패턴을 나타낸다.

Fig. 8 Test patterns

손상된 패턴은 신경망의 적응성을 테스트하기 위하여 사용되었다. 구성된 신경회로망은 뉴런 256개 (16X16 폰트사용)의 입력층, 뉴런 32개의 은닉층 1개, 뉴런 5개의 출력층을 사용하였으며, 각 네트워크에 저장된 가중치는 소프트웨어 학습 알고리즘으로 구했으며, 오류 역전파 알



고리즘을 사용하였다. 학습률은 0.05 이며, 에러 임계값 보다 에러가 작아지면 학습을 중지 시키는 방법을 사용하였다. 학습에 걸린 반복 회수는 약 2000회 이며, 출력 형태는 그레이 코드(Gray Code)형태로 학습 시켰다.

하드웨어 시뮬레이션 실험은 구현된 신경회로망 모델을 통해서 256 * 32 * 5 의 MLP구조를 구성한 뒤, 입력 패턴과 가중치를 하드웨어 사양에 맞게 24비트 데이터 포맷으로 변환하여 입력시키고, 출력된 데이터를 소프트웨어 결과의 오차와 비교하였다. 하드웨어의 연산 결과와 소프트웨어의 연산 결과를 비교했을 때, 시냅스 연산에 대해서 2% 이내의 오차율을, 비선형 함수에 대해서 3%의 오차율을 보였다.

5. 결론

제안된 신경회로망의 구조는 하드웨어로 구성 시 재구성이 용이하고 확장 가능하도록 고안되었음을 보였다. 하드웨어 상에서 신경망의 토폴로지와 활성함수를 바꿀 수 있도록 하여 실시간으로 동작하는데 많은 이점을 가지도록 하였다. 또한 모듈러 구성으로 인해서 다양한 형태의 신경망을 조합할 수 있도록 하였고, 이는 캐스케이드 연결 기법을 이용함으로써 더욱 많은 능력을 보여준다. 향후의 더 나은 연구를 위해서는 학습을 칩 안에서 이루어지도록 해야 하며 칩의 확장성을 보장하는데 많은 실험이 필요하리라 여겨진다.

참고문헌

(1) Robert J. Schalkoff, 1997, *Artificial neural networks*, McGraw-Hill, pp. 1~2, pp.411.

(2) Miroslav Skrbek, 1999.5, "Fast neural network implementation", *Neural Network World*, pp. 357~391.

(3) Tams Szab, Lrinc Antoni, Gbor Horvth, Bla Fehr, 2000, "A full-parallel digital implementation for pre-trained NNs" , *Proceedings of the IEEE-INNS-ENNS International Joint Conference on 2000*, pp.49~54.

(4) B. Pino, F.J.Pelayo, J. Ortega and A. Prieto, 1999, "Design and Evaluation of a Reconfigurable Digital Architecture for Self-Organizing Maps", *Proceedings of the Seventh International Conference on 1999*, pp.395-402.

(5) Mancia Anguita, Francisco I. Pelayo, Ignacio Rojas, and Alberto Prieto, 1998, "Area Efficient Implementation of Fixed-Template CNN's", *IEEE Transactions on circuits and systems - 1. Fundamental theory and applications*, Vol.45, No. 9.

(6) Ral Rojas, 1996, *Neural Networks : A Systematic Introduction*, Springer-Verlag, pp.452.

(7) Masahiro Murakawa, Shuji Yoshizawa, Isamu Kajitani, Xin Yao, Nobuki Kajihara, Masaya Iwata and Tetsuya Higuchi, 1999, "The GRD Chip : Genetic Reconfiguration of DSPs for Neural Network Processing", *IEEE Transaction on computers*, Vol.48, No.6.

(8) Bernard Girau, 2000, "Digital hardware implementation of 2D complatible neural networks", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on 2000*, Vol.3, pp.506~511.