論文94-31A-10-20

On-chip 학습기능을 구현한 최소 광역 제어 신경회로망 칩의 코어 설계

(Design of a Neurochip's Core with on-chip Learning Capability in Hardware with Minimal Global Control)

裵 寅 浩*, 黃 善 泳*

(In Ho Bae and Sun Young Hwang)

要約

본 논문에서는 on-chip 학습기능을 갖춘 병렬 처리 구조 신경회로망 칩의 구현에 대하여 기술한다. 본 논문에서 구현한 신경회로망 칩은 다수의 프로세스 소자를 이용하여 신경회로망의 많은 계산 요소를 병렬 처리함으로써 빠른 속도로 동작하도록 설계 하였으며 오류 역전파 알고리듬을 이용한 on-chip 학습 기능을 구현하였고 전체 프로세스 소자의 제어를 위한 광역 제어 신호를 최소화 하였다. 신경회로망 칩의 입력과 출력 및 연결 가중치의 표현에 많은 비트를 할당하여 정밀한 신경회로망 시뮬레이션이 가능하도록 설계하였으며, 각 기능이 독립적으로 동작하도록 설계하여 확장이 용이하도록 구현하였다.

Abstract

This paper describes the design of a neurochip with on-chip learning capability in hardware with multiple processing elements. A digital architecture is adopted because its flexiblity and accuracy is advantageous for simulating the various application systems. The proposed chip consists of several processing elements to fit the large computation of neural networks, and has on-chip learning capability based on error back-propagation algorithm. It also minimizes the number of global control signals for processing elements. The modularity of the system makes it possible to build various kinds of boards to match the expected range of applications.

I. 서 론

신경회로망의 고속처리를 위한 분산병렬화의 필요성 증가에 따라 신경회로망의 하드웨어적 구현을 위한 많은 연구가 진행되고 있다.[1][2] MIMD나 SIMD 형의 병렬컴퓨터를 이용한 신경회로망 시뮬레이션은 소프트웨어에 의한 구현과 같이 융통성이 크면서도 비교적 큰 규모의 신경회로망 시뮬레이션에서 요구하는 속도를 낼수 있는 장점이 있으나 많은 통신경로를 필요로 하는 신경회로망 시뮬레이션의 특성으로 인해 상당히 긴 통신 시간을 소비하게 하는 단점이 있다.

이러한 단점은 신경회로망의 적합한 하드웨어를 구현함으로써 해결할 수 있으며 뉴로 컴퓨터와 신경회로망 칩이 그 예이다.[3] 신경회로망 칩의 구현은 아날

*正會員, 西江大學校 電子工學科
Dept. of Elec. Eng., Sogang Univ.
接受日字 : 1994年 1月 24日

로그를 이용한 방식과 디지탈을 이용한 방식 또는 이 둘을 혼합하여 구현하는 방식이 연구되고 있다. AT&T에서 개발된 ANNA[4] 칩은 디지탈 컴퓨터와의 인터페이스를 위해 입출력은 디지탈을 이용하였으며 내부는 아날로그 회로를 이용하여 병렬처리라는 신경회로망의 본질적인 문제을 해결하였으나 아날로그 회로의 이용으로 on-chip 학습기능이 제공되지 않으며 유연성과 정밀도 측면에서 단점을 보인다. 디지탈로 구현된 TInMAMM[5] 칩 이나 Lneuro[6] 칩은 on-chip 학습기능이 제공되며 디지탈 컴퓨터와의 인터페이스 및 유연성, 확장성이 뛰어나다. 그러나 병렬적인 신경회로망의 많은 계산 요소들로 인해 비교적 느린 속도로 동작하며 0과 1 또는 몇단계의 값만이 사용되어 일반적으로 발생 할 수 있는 지속적인 값에 대한 처리가 미흡하여 정밀한 신경회로망의 시뮬레이션에는 취약하다는 단점이 있다.

본 논문에서 구현된 신경회로망 칩은 디지탈로 구현되어 디지탈 컴퓨터와의 인터페이스가 용이하고 다치논리 (multiple - valued logic)를 이용한 입력으로 지속적인 값에 대한 처리가 강화되었다. 학습 알고리듬으로는 오류 역전파 알고리듬을 이용하여 on-chip 학습기능을 구현하였으며 높은 해상도 (Resolution)의 연결 가중치를 이용하여 정밀한 학습과 신경회로망 시뮬레이션이 가능하도록 설계하였다.

디지탈 회로의 느린 연산속도의 단점을 보완하기 위하여 여러개의 프로세스 소자를 사용한 병렬처리가 가능하도록 신경망 알고리듬을 병렬 알고리듬으로 전환하였으며 각 프로세스 소자가 독립적으로 동작하도록 설계하여 광역 제어 신호를 축소시켰다. 각 프로세스 소자는 링 구조로 연결되어 있으며 NCU(Neuron Control Unit)를 삽입하여 광역 제어 신호의 발생을 전담하게 함으로써 신경회로망에 필요한 연산과 제어를 독립적으로 분할하여 전체 프로세스 소자 개수의 확장이 용이하도록 설계하였다.

본 논문의 Ⅱ장에서는 신경회로망 알고리듬에 대해 설명하고 Ⅲ장에서는 병렬 알고리듬의 전환과 아키텍처의 설정에 대해 설명한다. Ⅳ장에서는 설정된 아키텍처를 이용하여 설계된 신경회로망 칩과 합성 결과에 대해 기술하고 Ⅴ장에서는 구현된 신경회로망 칩의 실험결과에 대하여 기술하고 Ⅵ장에서는 결론 및 추후과제에 대하여 기술한다.

Ⅱ. 신경회로망 알고리듬

신경회로망의 기본 구조는 그림 1에 제시하였다. 신경회로망의 기본 동작은 입력 자극에 따라 뉴런의 출력을 생성하고 이는 다음 레이어의 자극으로 사용된다. 이와 같이 뉴런이 연쇄적으로 반응하여 분류, 인식 등의 지적 기능을 수행하며 전체적인 동작은 출력 생성 과정과 학습 과정으로 나눌 수 있다.

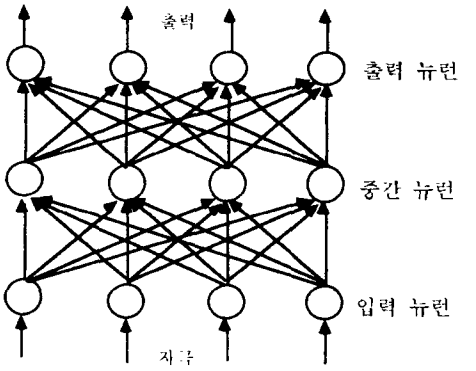


그림 1. 신경회로망의 기본 구조
Fig. 1. Basic structure of neural network.

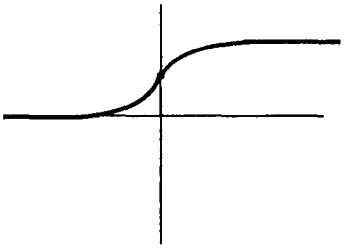


그림 2. 활성화 함수
Fig. 2. Activation function.

1. 출력 생성 과정

출력 생성 과정은 입력되는 자극에 대해 저장되어 있는 연결 가중치를 승산하여 총 자극을 구한 다음 활성화 함수를 이용하여 뉴런의 출력을 생성하는 과정이다. 본 논문 에서 구현된 칩은 역전파 delta학습 알고리듬에 적합한 sigmoid 함수를 사용하였으며 그 함수는 그림 2에 제시하였다.

각 뉴런의 동작은 다음과 같다. 입력신호 X_j(j = 0,1,…,n-1)에 대한 중간 레이어의 출력신호 T_i(i = 0,1,…,n-1)는 식(1)로 주어지며 이 식에서 사용된 활성화 상수 λ는 학습시간, 안정성(stability) 등에 영향을 끼친다.

$$T_i = f(net_i) = \frac{1}{1+\exp(-\lambda\, net_i)}, \quad net_i = \sum_{j=0}^{n-1} w_{i,j} * X_j \qquad (1)$$

출력 레이어의 출력은 중간 레이어의 출력을 입력

으로 사용하고 연결 가중치 W_{ki}를 이용하여 식 (1)과 같은 방식으로 최종출력 $O_k(k = 0,1,\cdots,n-1)$를 구한다.[7]

2. 학습 과정

학습 과정은 입력패턴에 대해 출력된 결과와 주어진 지도출력과의 오차를 최소화하기 위해 연결가중치를 교정하는 과정이다. 최종출력 $O_k(k = 0,1,\cdots,n-1)$와 지도 출력 $D_k(k = 0,1,\cdots,n-1)$와의 오차 E를 구하는 함수와 오차 그라디엔트값을 이용하여 연결 가중치의 교정값을 구하는 과정은 식(2)로 주어진다.

$$\Delta W_{ki} = -\eta \frac{\partial E}{\partial W_{ki}}, \quad E = \frac{1}{2} \times \sum_{k=0}^{n-1} (D_k - O_k)^2, \quad \eta : \text{학습상수} \quad (2)$$

출력 뉴런에 입력되는 총자극은 출력 레이어의 입력과 연결 가중치를 곱하여 적산한 것이며 오차 신호에 대한 δ_{ok}값은 각 뉴런 k에 대해 식 (3)로 주어진다.

$$\delta_{ok} \equiv -\frac{\partial W_{ki}}{\partial (net_k)} \quad (3)$$

체인규칙을 이용하여 식 (2)을 전개하면 식 (4)가 주어지며 net_k는 연결 가중치와 중간 레이어의 출력의 곱과 합 $(W_{ko} * T_0 + W_{k1} * T^{1+} \cdots + W_{kn\,1} * T_{n-1})$으로 이루어져 있으므로 뉴런 i에 대한 연결 가중치의 편미분은 식 (5)으로 주어진다.

$$\frac{\partial E}{\partial W_{ki}} = \frac{\partial E}{\partial (net_k)} \times \frac{\partial (net_k)}{\partial W_{ki}} \quad (4), \qquad \frac{\partial (net_k)}{\partial W_{ki}} = T_i \quad (5)$$

식 (3)는 체인 규칙에 의해 식 (6)로 전개되며 각 부분의 의미는 식 (7)로 주어진다.

$$\delta_{ok} = -\frac{\partial E}{\partial O_k} \times \frac{\partial O_k}{\partial (net_k)} \quad (6)$$

$$\frac{\partial E}{\partial O_k} = -(D_k - O_k), \quad f'(net_k) \equiv \frac{\partial O_k}{\partial (net_k)} \quad (7)$$

$f'(net_k)$는 활성화 함수의 net_k에 대한 편미분으로 식 (8)로 주어진다. 그러므로 출력 레이어의 연결 가중치 변화량은 식(9)으로 표현된다.

$$f'(net_k) = \frac{1}{1+\exp(-net_k)} \times \frac{1+\exp(-net_k)-1}{1+\exp(-net_k)} \quad (8)$$

$$= O_k \times (1.0 - O_k)$$

$$\Delta W_{ki} = \eta (D_k - O_k) O_k (1.0 - O_k) T_i \quad (9)$$

중간 레이어의 연결 가중치 교정 과정에서 변화량과 정의된 δ_{xi} 값은 식 (10)로 주어 진다.

$$\Delta W_{ij} = -\eta \times \frac{\partial E}{\partial W_{ij}} = -\eta \times \frac{\partial E}{\partial (net_i)} \times \frac{\partial (net_i)}{\partial W_{ij}} = \eta \delta_{xi} X_j \quad (10)$$

체인 규칙에 의해 전개된 δ_{xi}값과 의미는 식 (11)과 (12)로 주어진다.

$$\delta_{xi} \equiv -\frac{\partial E}{\partial T_i} \times \frac{\partial T_i}{\partial (net_i)} \quad (11)$$

$$\frac{\partial E}{\partial T_i} = \frac{\partial}{\partial T_i}\left[\frac{1}{2}\sum_{k-0}^{n-1}\left[D_k - f(net_k)\right]^2\right], \quad \frac{\partial T_i}{\partial (net_i)} = X_j \quad (12)$$

식 (11)를 이용하여 중간 레이어의 연결 가중치 변화량을 유도하면 식 (13), (14)로 전개되며 중간 레이어의 연결 가중치 변화량은 식 (15)으로 주어진다.[7]

$$\frac{\partial E}{\partial T_i} = -\sum_{k=0}^{n-1}(D_k - O_k)\frac{\partial}{\partial T_i}\left[f(net_k)\right]$$

$$= -\sum_{k=0}^{n-1}(D_k - O_k) f'(net_k)\frac{\partial (net_k)}{\partial T_i} = -\sum_{k=0}^{n-1} d_{ok} W_{ki} \quad (13)$$

$$\delta_{xi} = f'(net_i)\sum_{k=0}^{n-1}\delta_{ok} W_{ki}, \text{ for } i = 0,1,2\cdots,n-1 \quad (14)$$

$$\nabla W_{ij} = \eta\ f'(net_i)\sum_{k=0}^{n-1}\delta_{ok} W_{ki}, \text{ for } i,j = 0,1,2\cdots,n-1 \quad (15)$$

Ⅲ. 아키텍쳐 설정

1. 병렬 알고리듬으로의 변환

병렬 알고리듬으로의 변환은 순차적인 구조로 기술되어 있는 알고리듬을 여러개의 프로세스 소자를 이용하여 병렬처리를 할 수 있도록 변형하는 과정으로 각 프로세스 소자간의 연결구조및 기능이 결정된다. 본 연구에서 채택한 병렬처리 구조는 하나의 프로세

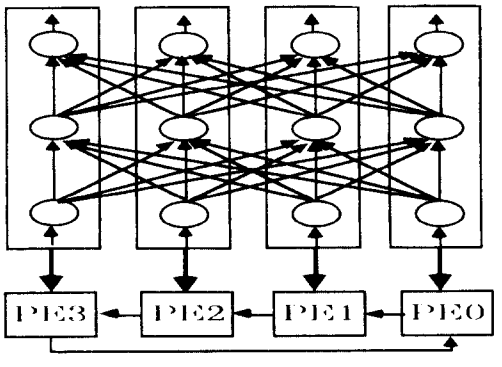


그림 3. 뉴론의 PE매핑

Fig. 3. Mapping the neural network onto PE.

스 소자가 각 레이어에 속한 하나의 뉴런 기능을 수행하는 것이다. 그림 3은 신경회로망에서 프로세스 소자로의 매핑 관계를 나타낸 것이다.

그림 3에서 하나의 프로세스 소자는 중간 레이어와 출력 레이어의 뉴런 기능을 수행하므로 중간 레이어의 연결 가중치와 출력 레이어의 연결 가중치를 저장할 수 있는 기억 용량이 필요하며 연결 가중치의 저장은 스큐메모리[13] 형태를 취하여 모든 과정에서 메모리를 순차적으로 엑세스하도록 하여 입력 어드레스 없이 읽기, 쓰기 신호로 동작하도록 설계하였다.

1) 출력생성 과정

출력생성 과정에서 필요한 연산을 독립적으로 각 프로세스 소자에 균일적으로 분산 하여 처리하였으며 병렬 알고리듬으로 변환된 출력생성 과정은 그림 4에 제시하였다. 출력 생성 과정이 시작되면 각 프로세스 소자는 저장되어 있는 입력 신호를 다음 프로세스 소자로 전송하게 된다. 그림 4의 각 과정에서 프로세스 소자는 전송되는 입력과 연결 가중치를 곱하여 적산한 후 입력된 신호를 다음 프로세스 소자로 전송하는 과정을 반복한다. 이러한 과정을 거쳐 적산된 결과는 프로세스 소자 내부에 있는 활성화 함수 블럭을 이용하여 자극에 대한 출력을 생성한다.

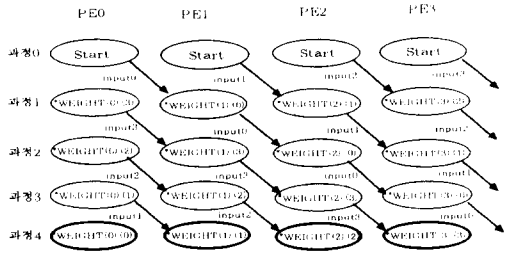


그림 4. 출력 생성 과정
Fig. 4. Output generation process.

2) 학습 과정

역전파를 이용한 방식은 역방향으로 교정이 이루어지므로 출력 레이어의 연결 가중치를 교정한 후 중간 레이어의 연결 가중치를 교정한다. 출력 레이어의 학습은 중간 레이어의 출력 및 출력 레이어의 출력과 지도 출력을 이용하여 이루어지며 교정하기 위해 필요한 식 (6)의 δ_k 값은 NCU(Neuron Control Unit)의 중간 생성 모듈에 의해 생성된다. 출력 레이어의 각 뉴런에서 생성된 최종 출력은 링 구조의 연결을 따라 NCU의 내부에 있는 중간 변수 생성 모듈을 통해 생성되며 이 값은 다시 프로세스 소자로 전송되어 저장된다. δ_{ik} 값이 저장된 후 각 프로세스 소자는 저장된 중간 레이어의 출력을 다음 소자로 전송하여 출력 레이어의 교정 과정이 시작된다. 입력되는 데이타는 출력 레이어의 교 정에 이용된 후 다음 소자로 전송된다.

중간 레이어의 연결 가중치 교정 과정은 교정된 출력 레이어의 연결 가중치와 δ_k 값과 입력 뉴런의 출력을 이용하여 이루어진다. 이 과정에서 식 (14)의 δ_{xi} 값이 이용되고 이 값은 각 프로세스 소자에서 연산한다. 구하는 과정은 그림 4의 형태를 이용한다. 과정 1에서 PE0는 $\delta_{00} \times weight_{ki}(0)(3)$이 생성되고 과정 2에서 이 값을 PE1으로 전송하여 $\delta_{00} \times weight_{ki}(0)(3) + \delta_{1} \times weight_{ki}(1)(3)$을 생성하여 과정 4에서 $\Sigma \delta_{ik} \times W_{ki}$ 값이 생성된다. 이 값과 NCU에서 생성된 $(D_k - O_k) \times O_k$ 값을 곱하여 식 (14)의 δ_{xi} 값을 구하게 된다. 위의 과정으로 δ_{xi}가 생성되면 중간 레이어의 연결 가중치 교정 과정이 시작되며 각 프로세스 소자는 출력 생성 과정과 같은 방식으로 저장되어 있는 입력 뉴런의 출력을 다음 프로세스 소자로 전송하여 중간 레이어의 연결 가중치를 교정한다.

2. 데이타의 표현

본 연구에서 구현한 신경회로망 칩내의 데이타는 signed magnitude 방식을 이용한 고정 소수점 값으로 표현된다. 연결 가중치는 19 비트, 뉴런의 입출력은 6비트, 승산 기의 출력은 25 비트, 적산기의 출력은 33 비트, 그라디엔트 연산기의 출력은 9 비트로 구성되어 있으며 있으며 뉴런의 입출력을 제외한 각각의 MSB는 부호에 해당한다.

Ⅳ. 신경회로망 칩의 구현

1. 전체 구조

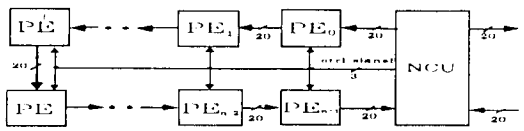


그림 5. 전체 구조
Fig. 5. Overview of neurochip.

전체 구조는 그림 5에 제시하였다. 각 프로세스 소자간의 버스는 20 비트로 구성되 어 있고 NCU에서 발생되는 광역 제어 신호는 3 비트로 구성되어 있으며 전체 프로세스 소자의 동작을 제어하는 정보를 전

송한다. 외부와의 데이타 입출력은 각각 입력 20 비트, 출력 20 비트를 이용하며, NCU는 프로세스 소자와 외부와의 입출력을 연결하거나 프로세스 소자간을 연결하는 기능을 수행한다.

2. NCU

NCU는 입출력 제어기, 출력 확인 모듈, 중간 변수 생성 모듈, 광역 신호 제어 모듈로 구성되며 자세한 블럭 구조는 그림 6에 제시하였다. NCU의 내부 콘트롤러는 모드와 상태에 따라 NCU 내부의 각 블럭의 구동과 연결을 제어한다. 외부로부터 입력되는 명 령어에 의해 전이되는 모드는 대기, 학습, 분류, 연결 가중치 적재, 연결 가중치 저 장, 입력 적재 모드가 있으며 이들 모드하에서 카운터의 출력에 의해 상태가 전이되어 그 과정에 적절한 제어 신호를 발생하게 된다.

입출력 제어기는 외부 입력, 프로세스 소자의 출력과 NCU 내부의 각 모듈의 출력을 입력으로 받아들여 프로세스 소자 또는 외부 출력 포트로 출력한다. 입출력 제어기는 내부 콘트롤러에서 전송되는 콘트롤 신호에 의해 제어되며 모드와 상태에 맞게 입력과 출력을 연결하는 기능을 수행한다.

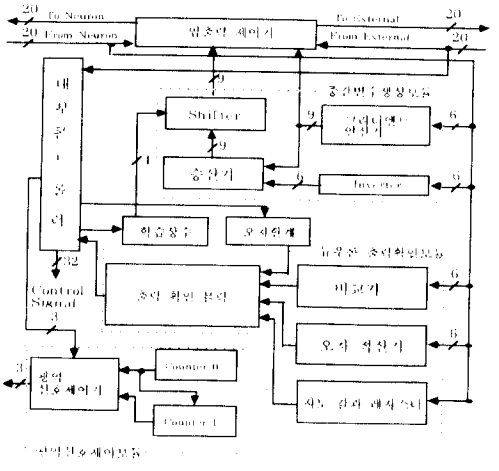


그림 6. NCU의 내부 블럭 구성도
Fig. 6. Overview of NCU.

광역 신호 제어 모듈은 전체 프로세스 소자를 제어하는 광역 신호를 전송하는 기능을 수행한다. 이 광역 신호로 전송하는 정보는 프로세스 소자의 모드와 상태의 전이에 필요한 내용으로 NCU의 내부 콘트롤러에서 발생되는 신호와 카운터에서 발생되는 신호를 이용하여 광역 제어 신호를 생성하여 전송한다. 광역 신호에 사용되는 카운터는 한 레이어의 뉴런수 N에 따라 결정되며 두개의 카운터가 필요하다. 각각의 카운터는 $\lceil \log_2 N \rceil$ 비트와 $\lceil \log_2 N \rceil + 1$ 비트의 크기를 가지고 있다. 전자의 카운터는 한 레이어의 뉴런수 만큼의 클럭 주기가 경과했음을 의미하며, 후자의 카운터는 전자의 카운터의 출력을 입력으로 사용하며 $2N^2$만큼의 클럭이 경과했음을 의미한다. 각 프로세스 소자는 카운터의 신호에 의해서 다음 동작을 결정하므로 NCU 내의 카운터의 확장만으로 전체 칩의 확장이 가능하게 된다.

중간 변수 생성 모듈은 학습 과정에서 필요한 중간 계산값을 생성하는 기능을 수행하며 승산기, 그라디엔트 연산기로 구성된다. 출력 레이어의 교정에 사용되는 식 (6)의 δ_{ik} 값을 생성하는 과정에서는 승산기의 출력이 이용되며 중간 레이어의 교정에 사용되는 식 (14)의 δ_{ij} 값을 생성하는 과정에서는 그라디엔트 연산기의 출력이 이용된다.

이 모듈에서 사용한 승산기는 9 비트와 6 비트 두개의 입력에 대해 15 비트의 출력을 생성하는 것으로 56개의 Full Adder로 구성된 Wallace Tree를 이용하여 설계하였다. 그라디엔트 연산기는 오차를 구하는 함수의 미분값을 구하는 과정에서 이용되며 6 비트의 입력을 받아들여 9 비트를 출력한다. 쉬프터는 저장되어 있는 학습 상수에 따라 승산기의 출력을 쉬프트하는 기능을 수행하여 δ_{ik}값을 조정함으로써 전체 신경회로망의 학습 속도의 조절이 가능하도록 한다.

뉴런 출력 확인 모듈은 출력 뉴런에서 생성된 출력을 이용하여 학습과정을 계속할 것인지 중지할 것인지를 결정하는 모듈로 비교기, 오차 적산기, 지도 결과 레지스터 및 출력 확인 블럭으로 구성된다. 이 모듈은 신경회로망의 최종 출력을 확인하여 학습과정의 계속유무와 신경망의 결과를 생성하는 과정에서 이용된다.

3. 프로세스 소자

그림 3에 제시한 바와 같이 하나의 프로세스 소자는 각 레이어에서 하나의 뉴런의 기능을 수행한다. 중간 레이어의 출력 생성 과정에서 각 프로세스 소자는 입력 신호와 저장되어 있는 연결 가중치를 곱하여 적산한 후 출력을 생성하며 출력 레이어의 출력 생성 과정에서는 중간 레이어의 출력을 이용하여 결과를 생성한다. 학습 과정에서는 중간 변수와 입력 및 중간 레이어의 출력을 이용하여 저장되어 있는 연결 가중치를 출력 생성 과정의 역방향으로 교정하여 저장한다. 이와 같은 기능을 수행하기 위한 프로세스 소자의 블럭 구조는 그림 7에 제시하였다.

PE의 내부 콘트롤러는 NCU에서 전송되어 오는 광역 제어 신호에 의해 제어되며 모드와 상태의 전이는 NCU의 내부 콘트롤러와 동기되며 매 클럭마다 입력되는 광역 제어 신호를 확인하여 모드와 상태를 전이한다. 내부 콘트롤러에서 발생된 신호는 입출력 제어기, 메모리, 가산기 등 프로세스 소자 내부의 각 모듈을 제어하는 기능을 한다.

뉴런 입출력 제어기는 각 레지스터 값, 활성화 함수 출력, 외부 입력, 메모리의 출력을 입력으로 사용하고 다음 프로세스 소자로의 데이타 전송 포트를 출력으로 사용한다. 입출력 제어기는 각 상태에 따라 입력과 출력을 연결하여 데이타를 다음 프로세스 로 전송하는 기능을 수행한다.

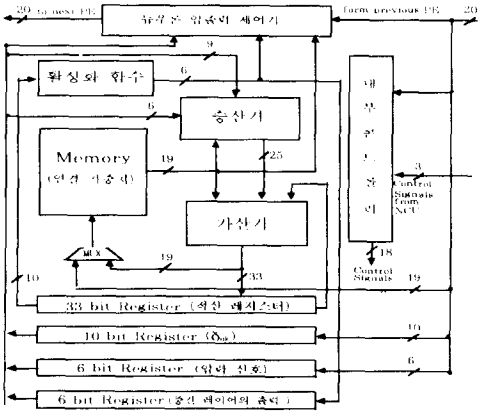


그림 7. 프로세스 소자의 내부 구성도
Fig. 7. Overview of Process Element.

프로세스 소자 내부에서 사용하는 승산기는 최대 19 × 6 비트의 연산을 수행해야 하기 때문에 승산기 내부에 가산기를 두므로써 두번의 승산을 통해 생성된 15 비트의 결과들을 합산하여 25 비트의 결과를 출력한다. 승산기 내부의 구조는 그림 8에 제시 하였으며 여기에서 사용된 Wallace Tree는 호스트에서 설계된 것과 일치한다. 그 외의 입력이 사용되는 경우(δ_k값과 중간 레이어의 출력이 입력으로 사용되는 경우)에는 한번 승산한 결과를 출력하며 승산기 앞단의 MUX/Alignment는 여러 입력 가운데 각 모드와 상태에 적합한 것을 선택하여 승산기로 입력하는 기능을 수행한다.

가산기는 출력 생성 과정에서 입력되는 자극의 적산과 학습 과정에서 연결 가중치 의 교정과정에서 사용되며 입력으로는 19 비트의 연결 가중치, 33 비트의 적산 레지스 터의 출력, 25 비트의 승산기 출력이 이용되며 MUX/Alignment는 이러한 입력가운데 각 과정에서 적절한 것을 콘트롤 신호에 의해 선택하여 가산 하도록 하는 기능을 수행한다. 가산기의 내부블럭 구조는 그림 9에 제시하였다.

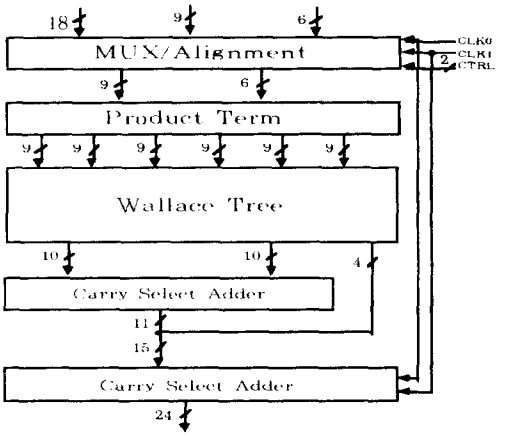


그림 8. 승산기의 내부 구조
Fig. 8. Overwiew of multiplier.

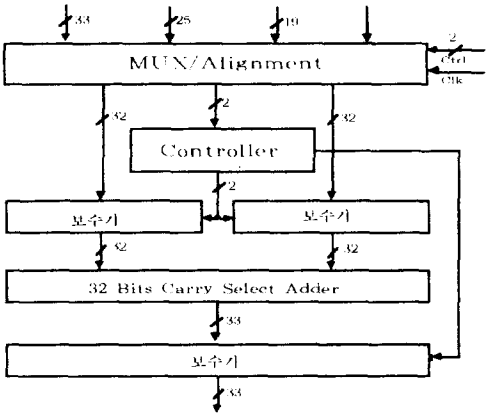


그림 9. 가산기의 내부 구조
Fig. 9. Overview of adder.

활성화 함수는 각 뉴런에 입력되는 입력 신호와 저장된 연결 가중치를 곱한 것을 적산하여 구한 총 자극에 대해 출력을 구하는 비선형 함수이다. 활성화 함수의 선택과 설계는 안정성, 학습 속도, 정밀도 등의 전체 시스템 성능에 중대한 영향을 끼치므로 학습 알고리듬에 적합한 뉴런의 출력을 생성하도록 모델링

되어야 한다. 본 논문에서 사용된 Sigmoid 함수를 하드웨어로 구현하기 위해서 부분적으로 직선화 하는 근사화 과정이 필요하였다.

4. 합성 결과

하드웨어 설계는 VHDL을 이용한 계층적 설계방식을 채택하였으며 효율적인 설계를 위하여 각 하드웨어 콤포넌트를 행위 기술, 데이타 흐름 기술, 구조 기술을 이용하여 기술하였다. 활성화 함수, 비교기, 그라디엔트 연산기 등은 다단 논리 합성 시스템을 이용하여 설계하였으며, 승산기 및 가산기 등의 연산기는 전가산기 및 반가산기들이 연결된 구조 기술로 설계하였다. NCU의 입출력 제어기와 뉴런의 입출력 제어기는 데이타 흐름기법으로 기술하였으며 콘트롤러부분은 행위 수준에서 기술하였다. 합성은 RT 수준 VHDL 합성 시스템인 VSYN[16]을 이용하였고 논리 최적화는 다단 논리 최적화 시 스템인 Smile-II[14]를 이용하였다.

그림 10는 활성화 함수 부분을 구현한 결과이며 11비트의 입력에 대해 6비트의 결과를 5×(AND _gate _dalay)이내에 생성하며 45개의 Gate로 구성된 다단 논리로 설계되었다.

그림 11은 그라디엔트 연산기를 합성한 결과이며 이 부분은 오차 측정함수의 그라 디엔트를 구하는 과정에서 필요한 함수로 "(1.0 - 입력) × 입력" 의 연산 결과를 출력한다. 입력은 활성화 함수의 출력이 이용되며 입력은 6 비트로 구성되며 9비트의 결 과를 출력한다. 그림 12는 승산기에서 사용된 Wallace Tree를 합성한 결과이며 각각의 컴포넌트는 전가산기와 MUX로 구성되어 있다.

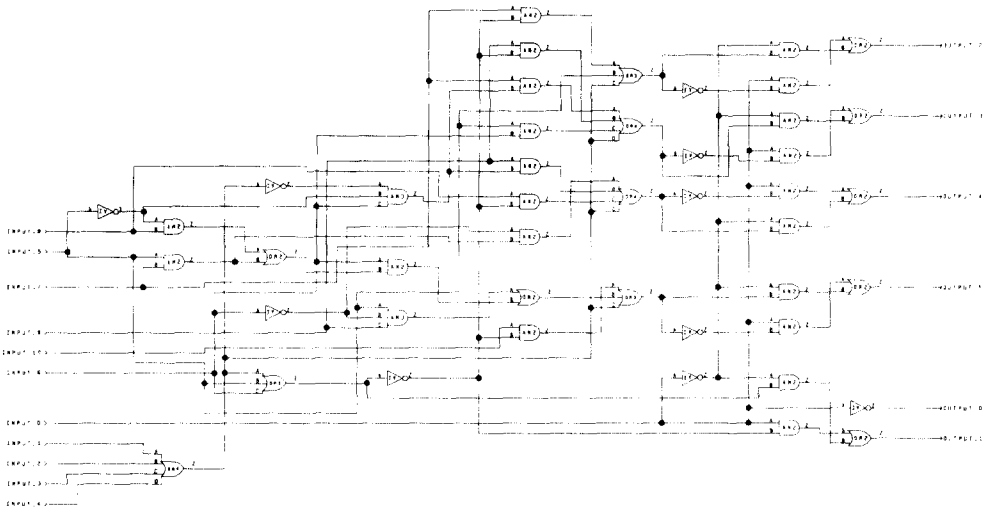


그림 10. 합성된 활성화 함수
Fig. 10. Synthesized activation function.

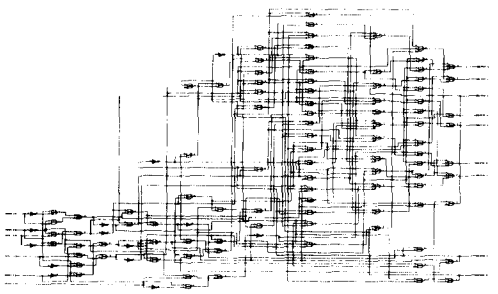


그림 11. 합성된 그라디엔트 연산기
Fig. 11. Synthesized gradient function

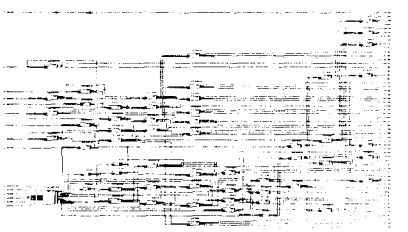


그림 12. Wallace tree
Fig. 12. Wallace tree.

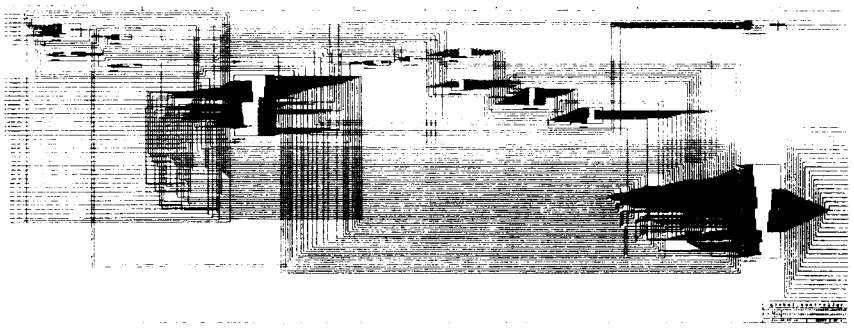


그림 13. NCU의 합성 결과
Fig. 13. Synthesized NCU.

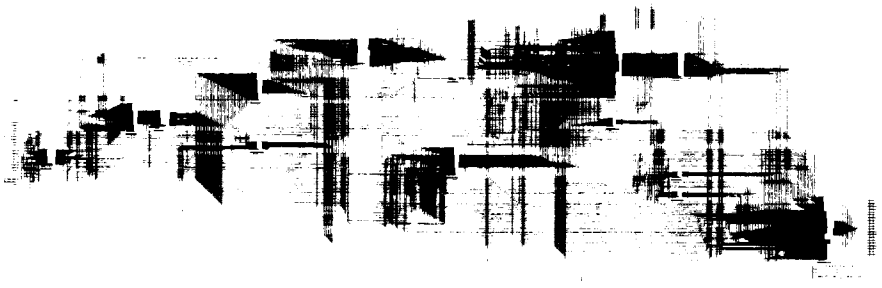


그림 14. PE의 합성 결과
Fig. 14. Synthesized PE.

그림 13은 전체 프로세스 소자를 관리하고 외부와의 입출력을 제어하는 NCU를 합성한 결과이며 그림 14는 프로세스 소자를 합성한 결과이다.

V. 실험 결과

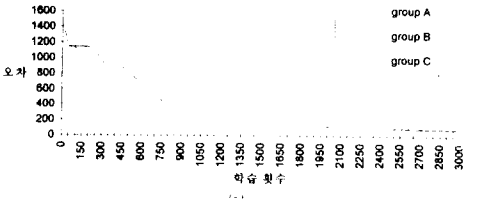
본 연구에서 설계된 신경회로망 칩의 학습 능력과 인식 능력을 평가하기 위하여 각 레이어당 뉴런수를 4개, 8개, 16개로 변화시켜 실험을 수행하였다. 이러한 실험은 설 계된 PE의 정밀도와 학습 능력 및 안정성을 측정하기 위한 것으로 각 레이어당 뉴런수가 증가할수록 정밀도와 안정성 문제는 중요한 요인으로 대두된다. 안정성 문제를 해결하기 위하여 구현된 신경회로망 칩은 학습 상수를 조절하는 방법을 채택하였으며 각 레이어당 뉴런수가 증가할수록 학습 상수의 적절한 조절이 중요하게 된다. 이러한 실험을 위한 테스트 데이타로는 얼굴 인식과 사진에서의 기호 인식을 위한 입력 패턴을 이용하였다.

1. 레이어당 4개의 뉴런

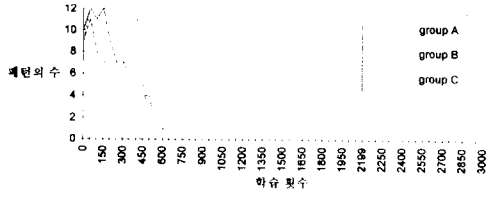
4개의 프로세스 소자를 이용하여 중간 레이어, 출력 레이어의 뉴런수를 4개로 고정하고 각 패턴에서 추출한 4개의 입력 데이타를 이용하여 4명을 분류하는 실험을 하였다. 입력 데이타로는 얼굴 인식에서 일반적으로 통용되는 이목구비에 대한 비를 이용 하였으며, 각 그룹별로 1인당 3장씩 총 12장의 사진에서 특징을 추출하여 4명을 하나의 그룹으로 하여 그룹 A, B, C에 대하여 인식 실험을 수행하였다. 입력 신호로는 양쪽 눈길이의 평균값을 미간의 길이로 나눈 값, 입의 길이를 미간의 길이로 나눈 값, 눈에서 코까지의 길이를 미간의 길이로 나눈 값과 코에서 입까지의 거리를 미간의 길이로 나눈 4개의 데이타를 이용하여 실험하였으며 실험 결과는 그림 15에 제시하였다.

그림 15(a)에 제시한 오차적산기의 오차는 뉴런의 출력 6비트를 정수로 전환하여 측정한 것으로 뉴런의 최대 출력을 63으로 하고 최소 출력을 0으로 환산한

것이다. 2500회의 학습을 하였을 때 그룹 A의 오차는 90이었다. 12개의 실험 패턴에 대해 실험 하였으므로 하나의 출력 뉴런에 대한 평균 오차는 90(오차 적산기의 오차) / 12(실험 패턴의 수) / 4(출력 뉴런 수) 가되어 1.875가 된다. 즉, 지도 출력이 0으로 주어졌을 때 출력이 약 2가 되는 것을 의미하며 지도 출력이 63으로 주어졌을 때 61의 출력을 생성함을 의미한다. 그림 15(b)는 학습 횟수에 따라 인식하지 못하는 패턴의 수이다.



(a)



(b)

그림 15. PE가 4개일 경우 얼굴인식 실험 결과
(a) 오차적산기의 오차
(b) 인식하지 못한 패턴의 수

Fig. 15. Experimental results of face recognition with 4 PEs.
(a) The value of error accumulator,
(b) unrecognizable patterns.

표 1. 소프트웨어를 이용한 실험 결과

Table 1. Software simulation.

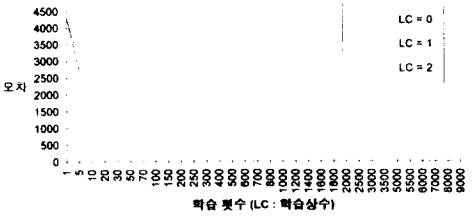
	그룹 A	그룹 B	그룹 C
인식 완료 학습 횟수	2093 회	1295 회	7595 회
인식 완료시 평균 오차	0.0908635	0.1141428	0.1007763
오차 함수	$0.5 \times (D_k - O_k)^2$/패턴수/한레이어의 뉴런수		

D_k : 지도 출력 O_k : 출력 레이어의 출력

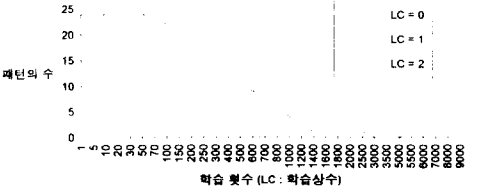
뉴런이 4개인 경우에 대한 실험은 사용된 세 그룹의 실험 패턴에 대해 모두 인식하 였으며 3000회의 학습 후 각 출력 뉴런의 평균 오차는 그룹 A인 경우 1.667, 그룹 B인 경우 1.813, 그룹 C인 경우 8.958 이었다. 소프트웨어를 이용하여 학습 상수를 0.4, 활성화 상수를 0.2로한 시뮬레이션 결과는 표 1에 제시하였으며 본 연구에서 구현한 신경회로망 칩은 한 레이어의 뉴런수가 4개인 경우에 대해서는 소프트웨어에 비견할 만한 인식 능력 보였으며 학습 상수의 변화없이 안정성 및 정밀도를 유지하였다.

2. 레이어당 8개의 뉴런

8개의 프로세스 소자를 이용하여 중간 레이어와 출력 레이어의 뉴런수를 8개로 하여 인식 실험을 수행하였다. 실험 데이타는 1절에서 사용한 얼굴 인식을 위한 입력 패턴중 그룹 A와 그룹 B를 한번에 인식하는 실험을 수행하였으며 학습상수에 따른 실험 결과는 그림 16에 제시하였으며 입력은 1절에서 사용한



(a)



(b)

그림 16. PE가 8개인 경우의 얼굴 인식 실험 결과
(a) 오차 적산기의 오차
(b) 인식 하지 못하는 패턴의 수

Fig. 16. Experimental results of face recognition with 8 PEs.
(a) The value of error accumulator,
(b) The number of Unrecognizable patterns.

입력을 두번 사용하여 24개의 입력 패턴에 대해 각각 8개의 입력값을 생성하였다.

그림 16(b)에 보였듯이 실험에 사용된 입력 패턴을 모두 인식할 수 있었으며 9000 회의 학습 후 각 출력 뉴런의 평균 오차는 학습 상수가 '0'인 경우1.146이 되었으며 학습 상수가 '1'인 경우 1.932가 되었고 학습 상수가 '2'인 경우는 3.412가 되었다.

실험 결과 한 레이어의 뉴런수를 8개로 확장한 경우에도 모든 실험 패턴에 대해 학 습 상수의 조정없이 인식이 가능하였으며 학습상수가 '0'인 경우에 가장 정밀하게 인식하였고 뉴런수가 4개인 경우와 비교하여 평균 오차율을 유지하였다. 소프트웨어를 이용한 시뮬레이션은 학습 상수를 0.4, 활성화 상수를 0.2로 하였을때 2445회에 실험 패턴을 모두 인식하였으며 인식 완료시 오차는 1절의 오차 함수를 이용하여 평균 오차를 측정한 결과 6.629597/24/8가 되어 0.03453이 되었다. 실험 결과 한 레이어의 뉴런수를 8개로 하였을 경우에 대해서도 안정한 인식 능력을 보였다.

3. 레이어당 16개의 뉴런

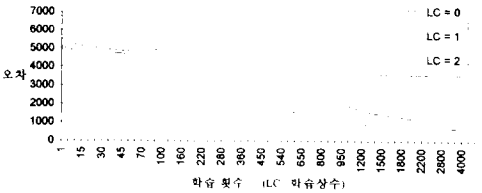
16개의 프로세스 소자를 이용하여 중간 레이어와 출력 레이어의 뉴런수가 16개인 경우에 대하여 실험하였다. 실험 데이타는 5.1절에서 사용한 얼굴 인식을 위한 입력 패턴과 사진에서의 기호 인식을 위한 입력 패턴을 이용하였다. 레이어당 뉴런수가 증가함에 따라 학습 상수의 조절이 필요하였으며 각 실험은 학습 상수의 변화에 따른 결과를 추출하여 안정성 및 인식능력에 대하여 측정하였다.

1) 얼굴 인식

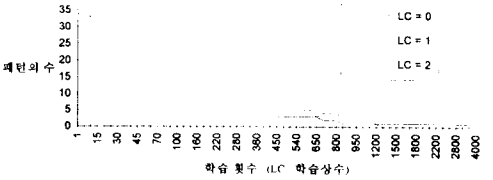
얼굴 인식에 대한 실험은 1인당 3장씩 36장의 사진에서 특징을 추출하였으며 추출된 데이타는 5.1절에서 사용한 데이타를 4번씩 입력하여 16개의 입력데이타를 생성하였다. 실험은 학습상수 (Learning Constant:LC)에 따라 실험하였으며 결과는 그림 17에 제시하였다.

실험에 사용한 입력 패턴에 대해 학습 상수가 '1', '2'인 경우에 대해서는 모든 실험 패턴에 대해 인식하였다. 4500회의 학습을 한 후 평균 오차는 학습 상수가 '1'인 경우 1.675가 되었으며 학습 상수가 '2'인 경우 3.226이 되었다. 학습 상수가 '0'인 경우 4500회의 실험으로는 하나의 입력 패턴에 대해서 인식하지 못하였다.

실험 결과 한레이어의 뉴런수가 16개인 경우에 대해서는 학습 상수의 조절이 필요하였으나 실험 패턴에 대해 모두 인식하였다. 소프트웨어를 이용한 실험 결과 6493회에 모든 실험 패턴에 대해 인식하였으며 평균 오차는 0.009159가 되었다.



(a)



(b)

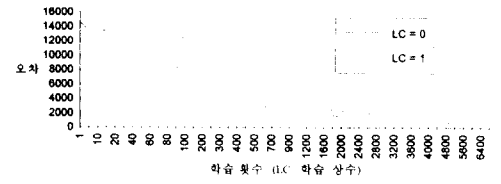
그림 17. PE가 16개인 경우의 얼굴 인식 실험 결과
(a) 오차적산기의 오차
(b) 인식하지 못하는 패턴

Fig. 17. Experimental results of face recognition with 16 PEs.
(a) The value of error accumulaotr
(b) The number of unrecognizable patterns.

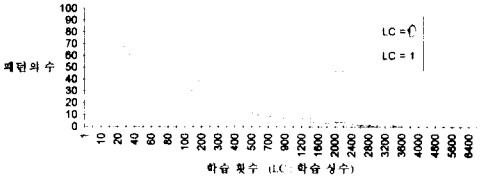
2) 기호인식

기호 인식에 대한 실험은 지도에서 빈번하게 나타나는 심볼 16개에 대하여 60 × 60 화소의 영상을 이용하였으며 각각의 영상은 60°씩 회전하여 얻은 6개의 영상에 대하여 정규화된 7차 모멘트[15]와 이 모멘트를 이용하여 추출한 9개의 데이타를 사용하였다. 16개의 기호에 대해 96개의 특징벡터를 추출하였으며 결과는 그림 18에 제시하였다.

기호 인식에 대한 실험에 대해서는 7100회의 학습으로 모두 인식할 수 있었으며 출력 뉴런의 평균 오차는 학습 상수가 '0'인 경우 0.367, 학습 상수가 '1'인 경우 0.666이 되었다. 소프트웨어 실험 결과 51455회에 모든 입력 패턴을 인식하였으 며 평균 오차는 0.0014565이었다. 실험 결과 실험 패턴에 대한 인식은 본 논문에서 구현한 칩이 빠른 학습 능력을 보였으며 정밀도는 소프트웨어 시뮬레이션이 우수하였다.



(a)



(b)

그림 18. 뉴런이 16개인 경우의 기호 인식 실험 결과
(a) 오차적산기의 오차값
(b) 인식하지 못하는 패턴의 수

Fig. 18. Symbol recognition simulation with 16 PEs
(a) The value of error accumulator
(b) The number of unrecognized patterns

4. 실험 결과 분석

본 논문에서 설계된 회로는 제한된 Resolution을 갖는 하드웨어로 구현되었으므로 입력 데이타의 정밀도에 제한을 갖는다. 즉, 하나의 입력 데이타가 6 비트로 정의됨에 따라 인식하여야 할 입력데이타가 (상한값 - 하한 값)/64보다 작은 차이를 보이는 경우 이진값으로 변환된 입력값이 같게 되어 인식이 불가능하게 된다. 실험에서 이용된 입력 데이타는 상한 값과 하한 값을 고려한 전처리 과정을 통하여 신경회로망의 입력으로 사용되었다. 본 논문에서 설계된 회로는 전처리 과정을 제외한 신경회로망의 코어 부분으로 적용분야가 결정되어지면 그에 따른 적절한 전처리 블럭을 설계하여 상한 값과 하한 값의 전범위에 입력 데이타가 고루 분포하도록 하여 입력 값의 한계를 최소화 하는 과정이 필요하다.

제한된 연결가중치의 값이 허용 범위를 초과할 경우 한계값으로 고정하여 발산하는 경우를 방지하였으나 한계 값을 갖는 경우 더이상 학습이 진행되지 않는 경우도 발생 가능하다. 그러나 위의 실험에서 일반적으로 얼굴 인식과 기호 인식에 사용되는 입력 패턴을 이용한 학습능력에 대한 실험 결과에서 안정적으로 학습되어짐을 알 수 있었다.

Ⅵ. 결론 및 추후 과제

본 논문에서는 기본적인 신경회로망 동작을 수행하는 신경회로망 칩의 구현에 대하여 기술하였다. 구현된 칩은 프로세스 소자의 확장이 용이하도록 설계되어 NCU의 카운터 신호를 변형함으로써 확장된 프로세스 소자들의 콘트롤이 가능한 구조를 이루고 있으며 각 프로세스 소자는 NCU에서 발생되는 광역 제어 신호에 의해 구동되므로 프로세스 소자의 수에 대해 독립적으로 동작하도록 설계되었다.

학습 과정은 학습횟수 모드, 제한 에러 모드, 구분 모드로 분류되어 각각 정해진 학습횟수에 도달할 경우, 정해진 에러 이하일 경우, 모든 학습 데이타의 분류가 정확할 경우 학습을 중단할 수 있게 설계하여 융통성 있는 학습이 가능하도록 구현하였으며 연결가중치의 값을 제한하여 안정성을 유지하도록 설계하였다. 또한 학습 상수를 이용한 학습속도의 조절로 정밀도의 조절이 가능하도록 구현되었다.

본 논문에서 설계된 NCU와 PE는 여러개의 단위 모듈로 구성되어 있으며 가장 큰 Delay를 갖는 임계 부분은 승산기로 8 × FA_delay의 임계 경로를 가진다. 설계된 회로에서 하나의 패턴에 대해 학습을 수행하는 시간은 한 레이어의 뉴런수가 N일 경우 6N+4 클럭 주기가 소요된다.

추후과제로는 학습 기능의 고려로 일반적인 분류동작시 지연되는 시간에 대한 연구가 진행되어야하며 프로세스 소자가 증가함에 따라 발생되는 활성화 함수의 동작, 최적의 연결가중치 정밀도, 학습을 위한 중간 변수의 정밀도 등에 대한 실험과 연구가 진행되어야 할 것이다.

參 考 文 獻

[1] C. Mead, Analog VLSI and Neural Systems, Addison-Wesley Pub.: Reading, MA, 1989.

[2] M. Nelson and W. Illingworth, A Practical Guide to Neural Networks, Addison-Wesley Pub. : Reading, MA, 1991.

[3] R. Hodges, C. Wu, and C. Wang, "Parallelizing the Self-Organizing Fracture Map on Multi-Processor

Systems," Proc. Int. Conf. on Neural Network, Vol. Ⅱ, Washington, DC, pp. 141-144, Jan. 1990.
[4] W. Lin, V. Prasana, and K. Przytula, "Algorithmic Mapping of Neural Network Models onto Parallel SIMD Machine," IEEE Trans. Computers, Vol. 40, No. 12, pp. 1390-1401, Dec. 1991.
[5] B. Lee and B. Sheu Hardware Annealing in Analog VLSI Neurocomputing, Kluwer Academic Publishers, 1991.
[6] E. S ckinger, B. Boser, J. Bronley, Y. LeCun and L. Jackel "Application of the ANNA Neural Network Chip to High Speed Character Recognition," IEEE Trans. on Neural Networks, Vol. 3, No. 3, pp. 498-505, May 1992.
[7] M. Melton, T. Phan, S. Reeves and D. Van den Bout,"The TInMANN VLSI Chip," IEEE Trans. on Neural Networks, Vol. 3, No. 3, pp. 375-383, May 1992.
[8] N. Mauduit, M. Duranton, J. Gobert and J. Sirat, "Lneuro 1.0: A Piece of Hardware LEGO for Building Neural Network Systems," IEEE Trans. on Neural Networks, Vol. 3, No. 3, pp. 414-421, May 1992.
[9] DARPA Neural Network Study, AFCEA International Press : Fairfax, VA, 1988.
[10] R. Lippmann "An Introduction to Computing with Neural Nets," IEEE ASAP Magazine, Vol. 3, No. 4, pp. 4-22, 1987.
[11] J. Zurada, Introduction to Artificial Neural Systems, West Pub., 1992.
[12] D. Burr, "Experiments on Neural Net Recognition of Spoken and Written Tex t," IEEE Trans. on Acoustics Speech and Signal Processing, Vol. 36, No. 7, pp. 1162-1168, July 1988.
[13] K. Hwang and F. Briggs, Computer Architecture and Parallel Processing, McGraw Hill: NY, 1992.
[14] 임 춘석, 황 선영, "Fanin 제약 하의 다단 논리 최적화 시스템의 설계," 대한 전자 공학회 논문지, 29-A권, 4호, pp. 689-692, 1992년 4월
[15] M. Hu, "Visual Pattern Recognition by Moment Invariants.", IRE Trans. Info. Theory, Vol. IT-8, pp. 179-187, 1962.
[16] 현 민호, 황 선영, "레지스터 전송 수준에서의 VHDL 순서문 합성에 관한 연구," 31-A권, 5호, pp.324-332, 1994년 5월

著 者 紹 介

裵 寅 浩(正會員)

1992年 서강대학교 전자공학과 학사. 1994年 서강대학교 전자공학과 대학원 석사. 1994年 ~ 현재 삼성 전자 마이크로 사업부 연구원. 주관심 분야는 신경회로망, CAD 시스템 Computer Architecture 및 System Design 등임.

黃 善 泳(正會員)

1976年 2月 서울 대학교 전자공학과 졸업. 1978年 2月 한국과학원 전기 및 전자 공학과 공학 석사 취득. 1986年 10月 미국 Stanford 대학 공학박사 학위 취득. 1976年 ~ 1981年 삼성 반도체 주식회사 연구원. 1986年 ~ 1989年 Stanford 대학 Center for Integrated Systems 연구소 연구원. Fairchild Semiconductor Palo Alto Research Center 기술 자문. 1989年 2月~ 현재 서강 대학교 전자 공학과 부교수. 주관심 분야는 CAD 시스템, Computer Architecture 및 Systems Design, VLSI 설계 등임.