

# 수정된 하니발 구조를 이용한 신경회로망의 하드웨어 구현

論文  
45~3~17

## A Hardware Implementation of Neural Network with Modified HANNIBAL Architecture

李 範 燁\* · 鄭 德 鎮\*\*  
(Bumyoub Lee, · Duckjin Chung)

**Abstract** - A digital hardware architecture for artificial neural network with learning capability is described in this paper. It is a modified hardware architecture known as HANNIBAL(Hardware Architecture for Neural Networks Implementing Backpropagation Algorithm Learning)<sup>[2]</sup>. For implementing an efficient neural network hardware, we analyzed various type of multiplier which is major function block of neuro-processor cell. With this result, we design a efficient digital neural network hardware using serial/parallel multiplier, and test the operation. We also analyze the hardware efficiency with logic level simulation.

**Key Words** : neural networks, neuro processor, multiplier, backpropagation algorithm, multi-layer perceptron

### 1. 서 론

신경회로망의 응용 가능성은 이미 통신, 제어, 신호처리등 광범위한 분야에 걸친 연구 결과들로부터 충분히 입증되고 있다. 그러나 신경회로망이 기본적으로 고등동물의 뇌구조를 기초로 하여 출발한 것이어서 단위시간당 막대한 수치연산을 요구하게 된다. 따라서 실시간 처리가 요구되는 분야에 있어서 신경회로망의 적용은 그러한 막대한 연산을 짧은 시간에 처리할 수 있는 전용 하드웨어를 필요로 한다. 이러한 전용 하드웨어의 구현 방법에는 크게 세 가지로 구분될 수 있는데 아날로그 방식과 디지털 방식, 그리고 아날로그와 디지털을 혼용하는 하이브리드 방식으로 구분된다. 이중 아날로그 방식은 속도와 비용의 측면에서 많은 이점을 제공하긴 하지만 재현성 정밀도의 제한 및 하드웨어의 융통성이 결여되어 있는 단점을 가지고 있다. 반면 디지털 방식의 접근은 프로그래머블하고 쉽게 확장할 수 있으며 정밀도의 제한이 없고, 다른 신호처리 시스템과 쉽게 결합할 수 있다는 장점을 가지고 있다. 디지털 방식의 가장 큰 문제점으로는 회로의 크기, 즉 많은 비용이 든다는 것이었지만 현대의 디지털 기술의 발전에 힘입어 비용 면에서도 아날로그 방식에 접근하고 있다[1]. 본 논문에서는 다층퍼셉트론 network(MLP)을 수용하며 오류 역전파알고리즘(errorbackpropagation algorithm)을 사용하여 학습될 수 있는 디지털 방식의 신경망 하드웨어 구조를 제안한다. 이는 HANNIBAL[2]구조를 수정하여 재구성한 것이므로 HANNIBAL과 제안된 구조를 정량적으로 비교 분석하여 주어진 하드웨어 자원의 이용 효율을 개선하는 방법을 제시한다.

### 2. 하드웨어의 구조

다층퍼셉트론 network을 하드웨어로 구현하기 위해서는 망의 recall 을 위한 자료의 순방향 전파와 학습 알고리즘(오류역전파 알고리즘)을 위한 자료의 역방향 전파를 수용할 수 있어야 한다. 이러한 정보흐름의 반전은 하드웨어 효율의 감소를 초래할 수 있으며 제안된 구조는 이러한 알고리즘의 수행시 하드웨어 효율의 감소를 최소화 할 수 있는 방법을 제시한다. 이는 HANNIBAL구조에서 효율 감소의 원인을 제거함으로써 이루어 졌다.

#### 2.1 신경망의 하드웨어의 mapping법 및 정보 전송방식

그림 1과 그림 2에 HANNIBAL과 제안된 구조의 신경망 mapping 방법과 정보전송 방식이 각각 나타나 있다. 이 두 구조는 하드웨어적인 측면에서 모두 단위연산소자(Node Processor : NP)의 1 차원 array를 이루고 있다. 그러나 HANNIBAL의 경우 층간의 명확한 경계를 가지고 있으며, 정보전송을 위한 2개의 양방향 pipeline을 구성하여 정보를 전송하는 반면, 제안된 구조의 경우 하나의 파이프라인을 사용하여 순방향 전파과정을 수행하고, 역방향 전파의 수행에 있어서는 정보를 broadcasting함으로써 양방향 파이프라인에서 발생될 수 있는 하드웨어 효율의 감소를 최소화 하였다. 양방향 파이프라인의 경우 정보의 흐름이 반전되는 경우, 파이프가 모두 비워질 때까지 모든 연산소자가 대기하여야 하는 문제가 발생하여 하드웨어의 효율이 급격히 감소하게 된다. 이에 대한 자세한 논의는 다음절에서 다루기로 한다.

#### 2.2 하드웨어 속도, 크기 및 효율

일반적으로 하드웨어의 연산 속도와 크기는 서로 상반되는 특성을 가지고 있어 이들 사이의 trade-off가 필연적이다. 그러나 하드웨어 자원의 이용 효율을 높일수록 속도의 현격한 감

\*正會員 : 仁荷大 工大 電子材料工學科 碩士課程

\*\*正會員 : 仁荷大 工大 電子材料工學科 副教授 · 工博

接受日字 : 1995年 10月 23日

最終完了 : 1996年 2月 10日

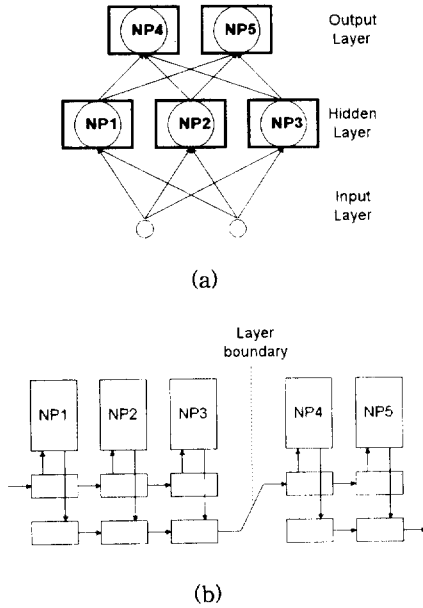


그림 1 HANNBAL구조의 신경망 매핑법 및 정보 전송방식  
 Fig. 1 Mapping neuron(a) and data transfer method(b) of HANNBAL architecture

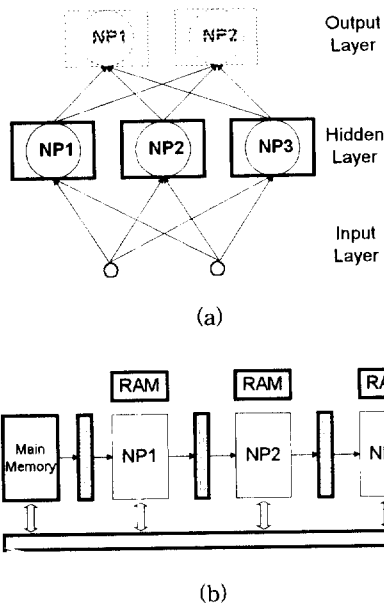


그림 2 제안된 구조의 신경망 매핑법 및 정보 전송방식  
 Fig. 2 Mapping neuron(a) and data transfer method(b) of proposed

소를 초래하지 않으면서 회로 크기를 줄일 수 있다. 이러한 특성을 분석하기 위하여 본 논문에서는 process-time graph를 사용하기로 한다[3, 4]. 그림 3과 그림 4는 HANNBAL 및 제안된 구조의 process-time graph가 나타나 있다. graph의 가로축은 하드웨어에 mapping된 신경망의 neuron을, 세로축은 시간을 나타내며 각 연산소자의 동작 상태는 크게 두 가지 형태로 분류하였다.

- Computation Mode (COMP) : 곱셈과 accumulation등 알고리즘 자체의 연산을 수행하며 단위 연산소자의 주요기능 block들이 연산동작을 수행하고 있는 경우
  - Idle or Communication (IDLE) : 단위연산소자가 정보를 전송하거나 대기하고 있는 상태
- 따라서 하드웨어의 효율이 증가하기 위해서는 COMP mode가 차지하는 비율이 커야 한다.

Communication의 경우 비록 연산소자가 대기상태인 것은 아니지만 주요 기능 block들은 아무런 동작도 하고 있지 않으므로 IDLE mode에 포함시켰다.

실제 분석을 위해 입력층이  $N_1$ , 중간층이  $N_2$ , 출력층이  $N_3$  개의 neuron을 가지는 3-layer 신경망을 사용하기로 한다. 좀 더 간단한 분석을 위하여 각층의 크기가 충분히 크다고 가정하면 그림 3과 그림 4는 그림 5와 같이 다각형 형태로 표시할 수 있다. 또한 구조적 측면의 분석을 위하여 하드웨어의 구성 조건

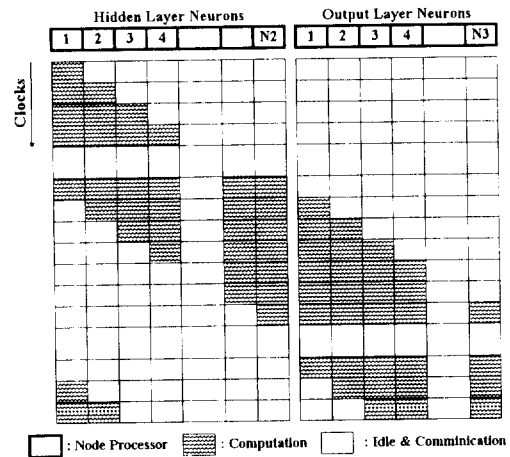


그림 3 NP들의 1차원 Array를 사용한 HANNBAL 구조의 Process Time Graph  
 Fig. 3 Process-time graph of HANNBAL architecture

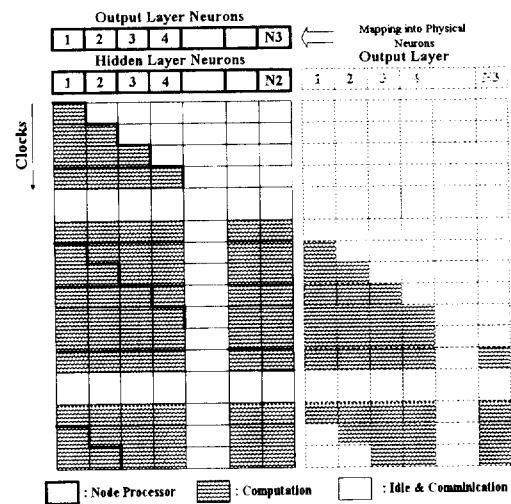


그림 4 NP들의 1차원 Array를 사용한 제안된 구조의 Process Time Graph  
 Fig. 4 Process-time graph of proposed architecture

및 중간층과 출력층의 크기는 동일한 것으로 간주한다.

두 구조의 연산 속도를 측정하기 위하여 latency 및 throughput 을 다음과 같이 정의하기로 하자.

- Latency : 하나의 입력 vector를 받아 완전한 출력 vector를 생성하는 데 소요되는 clock 수
- Throughput : 단위시간당 처리할 수 있는 입력 vector의 개수

그림 5를 토대로 HANNIBAL과 제안된 구조의 latency 및 throughput 그리고 필요한 단위연산소자의 개수가 표 1에 나타나 있다.

Recall과정(순방향 전파)의 throughput을 제외한 다른 특성들이 제안된 구조에서 우수함을 보이고 있는데, 이는 HANNIBAL의 양방향 파이프라인에서 각 연산소자의 출력자료의 loading 순서가 항상 동일해야 하고 정보흐름의 전환시 파이프가 모두 비워질 때까지는 다른 입력정보를 받아 처리할 수 없어 NP들이 대기하고 있어야 하기 때문이다.

또한 그림 5로부터 하드웨어의 효율을 간단히 계산할 수 있는데, 그림의 전체면적은 주어진 시간동안의 하드웨어 자원으로 볼 수 있고, COMM mode가 차지하는 면적은 하드웨어를 사용하는 양으로 볼 수 있다. 따라서 하드웨어 효율은 식(1)과 같이 정의될 수 있다.

$$\begin{aligned} \text{Hardware Efficiency} &= \frac{\# \text{ of Computing Node Processor}}{\# \text{ of Total Node Processor}} \\ &= \frac{\text{Area of computing mode}}{\text{Total Area}} \end{aligned}$$

(1)

표 2에 HANNIBAL과 제안된 구조의 하드웨어 자원의 이용 효율이 나타나 있다. 그림 5에서 볼 수 있듯 속도의 분석에서와 동일한 이유로 인하여 HANNIBAL은 learning과정에서의 효율이 급격히 감소하게 된다.

표 1 HANNIBAL과 제안된 구조의 Latency 및 Throughput

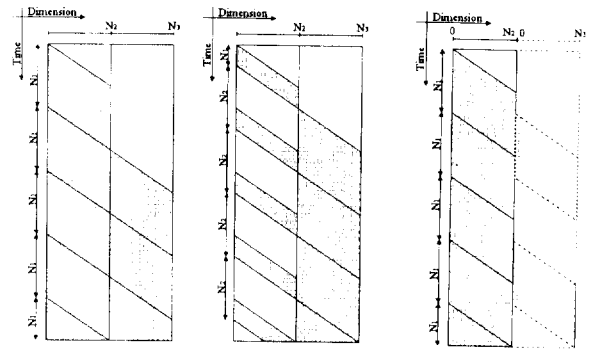
Table 1 Latency and throughput of HANNIBAL and proposed architecture

구분		HANNIBAL	제안된 구조
Latency	recall	$N_1+2N_2+N_3$	$N_1+2N_2$
	learning	$2N_1+5N_2+2N_3$	$2N_1+4N_2+N_3$
Throughput	recall	$1/N_1 (N_1 \geq N_2)$ $1/N_2 (N_1 < N_2)$	$1/(N_1+N_2)$
	learning	$1/(2N_1+5N_2+2N_3)$	$1/(2N_1+4N_2+N_3)$
Number of NP		$N_2+N_3$	$N_2$

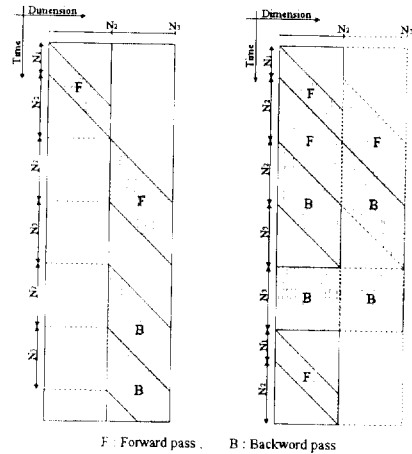
표 2 HANNIBAL과 제안된 구조의 하드웨어 효율

Table 2 Hardware efficiency of HANNIBAL and proposed architecture

Process	HANNIBAL	제안된 구조
Recall	$\frac{N_2N_3}{N_1(N_2+N_3)} \quad N_1 \geq N_2$	≈ 100 %
	$\frac{N_1N_2}{N_2(N_2+N_3)} \quad N_1 < N_2$	
Learning	$\frac{2N_1N_2+3N_2N_3}{(N_2+N_3)(2N_1+5N_2+2N_3)}$	$\frac{2N_1+2N_2+N_3}{2N_1+4N_2+N_3}$



(a) HANNIBAL(N<sub>1</sub>=N<sub>2</sub>) (b) HANNIBAL(N<sub>1</sub>=N<sub>2</sub>) (c) Proposed(any N<sub>1</sub>, N<sub>2</sub>)



(d) HANNIBAL (e) Proposed

그림 5 NP들의 1차원 Array를 사용한 HANNIBAL 및 제안된 구조의 간략화된 Process Time Graph

Fig. 5 Simplified process-time graph of HANNIBAL and proposed architecture:(a), (b), (c) recall, (e), (f) learning

### 3. 곱셈기의 선택

신경망 전용 하드웨어 설계에 있어서 가장 많은 면적을 차지하게 되는 부분은 가중치 저장을 위한 memory와 곱셈기 그리고 활성화 함수 등이다. 이중 memory가 가장 넓은 면적을 차지하게 되지만 이는 설계시에 한 neuron이 얼마나 많은 connection을 수용하게 할 것인가에 따라 그 용량과 면적이 결정되므로 구조적 관점에서는 논의의 대상이 되지 않는다. 실제로 신경망 하드웨어의 성능은 각 연산소자가 가지고 있는 곱셈기에 의해서 좌우된다. 따라서 본 장에서는 곱셈기를 형태별로 분석하고 신경회로망의 하드웨어 구현시 가장 적합한 곱셈기를 선택한다.

곱셈기의 형태는 크게 3종류로 분류될 수 있는데 병렬, 직렬, 직렬등으로 구분할 수 있다. 이들은 곱셈의 두 operand의 입력 형태에 따라 구분된 것인데 직렬의 경우 두 입력 모두를 1bit 씩 직렬로 입력해야 하므로 부수적인 제어회로가 필요하고, 연산속도에서의 불이익을 감수해야 하므로 실제응용에 있

표 3 대표적 곱셈기의 면적과 연산속도 (16bit X 16bit)

Table 3 Size and speed of multipliers

구분	Gate count	Speed (x10 <sup>7</sup> )	Normalized speed	Type
Wallace <sup>[4]</sup>	1756	2.19	1.00	P/P
Booth <sup>[4]</sup>	2405	2.32	0.77	P/P
1D systolic <sup>[5]</sup>	252	0.38	1.21	S/P
FSP <sup>[9]</sup>	370	0.38	0.82	S/P
Moh <sup>[7][8]</sup>	347	0.26	0.60	S/P
bit-level systolic <sup>[10]</sup>	266	0.33	0.99	S/P

어서는 별 이득이 없다. 따라서 본 논문에서는 병렬곱셈기와 직병렬 곱셈기에 대해서만 비교 분석하기로 한다. 일반적으로 병렬 곱셈기 중 속도와 크기면에서 가장 우수한 성능을 보이는 것이 Wallace tree를 사용한 것과 modified Booth's algorithm을 사용하는 것들이다[4]. 전자의 경우 CSA(Carry Save Adder)를 사용하여 carry propagation delay를 억제하고 partial product를 재배치하여 성능을 증가시키는 방법을 사용하고 있으며, 후자의 경우는 승수를 주어진 알고리즘에 따라 coding하여 partial product의 수를 줄이는 방식을 사용한다. 실제로 이들 병렬곱셈기의 연산속도는 직·병렬의 경우보다 훨씬 우수하다. 반면 이러한 병렬곱셈기의 경우 직·병렬의 경우보다 회로의 크기가 5~10배정도 커서 신경회로망과 같이 단위연산소자의 개수가 많아질 경우 하드웨어 구현에 큰 문제점을 낳게 된다. 따라서 좀더 작은 크기의 곱셈기가 요구되는데 실제로 직·병렬 곱셈기의 경우 병렬곱셈기보다 크기는 적고 효율이 좋아 많은 연산소자의 집합체인 신경회로망과 같은 분야에서 많은 이점을 제공하게 된다. 다음 표 3에 대표적인 곱셈기들의 속도와 gate수 등을 비교하였다. 여기서의 모든 수치는 2입력 NAND gate의 delay와 면적을 1로하여 계산된 결과이다.

표 3에서 speed는 1개의 곱셈 연산에 필요한 전체 gate delay의 역수를 취한 것이고, normalized speed는 모든 곱셈기의 size를 동일하게 유지한 후 Wallace Tree를 기준으로 normalize한 speed이다. 따라서 normalized speed가 높을수록 동일한 면적을 사용하여 하드웨어를 구현했을 경우 더 많은 곱셈연산을 수행할 수 있음을 의미한다. 따라서 가격 대 성능비가 우수한 시스템을 구현하고자 할 경우 normalized speed가 우수한 곱셈기를 선택하여야 한다. 그러나 실제의 응용에 있어서 가격 또는 성능을 더 중시하여야 하는 경우가 있으며 이러한 경우에 대한 분석은 본 논문의 범위를 벗어나므로 이에 대한 자세한 논의는 생략하기로 한다.

이상의 곱셈기에 대한 분석 결과로부터 본 연구에서는 가격 대 성능비가 우수하고 회로의 regularity가 뛰어난 1-D serial/parallel systolic multiplier[5]를 사용하였다.

### 4. 직병렬 곱셈기를 이용한 신경망 하드웨어의 설계

제안된 구조의 성능을 분석하기 위하여 직병렬 곱셈기를 이용한 신경망 하드웨어를 설계했다. 실험을 위해 설계된 신경망 하드웨어는 4개의 단위연산소자를 가지며, 프로그래밍을 통해 3 layer의 신경망을 수용할 수 있고, 자체 학습이 가능하도록 설계되었다.

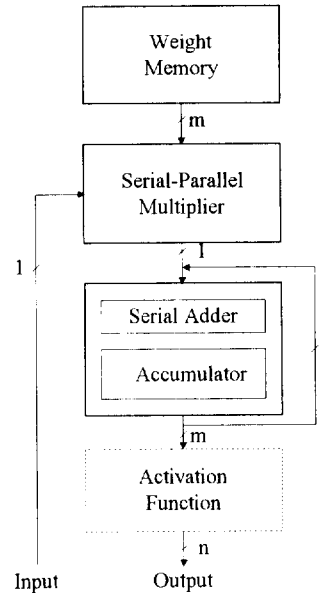


그림 6 기본 연산소자  
Fig. 6 Basic Node processor

#### 4.1 Recall과정을 위한 구조

그림 6의 기본 구조는 식(2)에 표현된 신경망의 recall 과정을 수행하기 위한 것으로서 가중치를 저장하기 위한 memory와 곱셈기, accumulator 그리고 활성화 함수를 가지고 있어야 한다.

$$OUT_j = f\left(\sum_i W_{ji} I_i + \theta_j\right) \tag{2}$$

Recall과정 즉 순방향 전파의 과정은 병렬로 입력된 가중치와 직렬로 입력되는 입력 vector를 곱하고 accumulator에 누적시키다가 모든 입력 vector에 대한 연산이 끝나면 활성화 함수를 통하여 출력된다. 이러한 과정은 HANNIBAL구조의 경우와 유사하지만 HANNIBAL의 경우에는 병렬 곱셈기, 병렬 adder 그리고 각 단위연산소자마다 활성화 함수를 내장하고있는 반면 제안된 구조의 경우는 직병렬 곱셈기를 사용함으로써 accumulation을 위하여 단지 수개의 gate로 이루어진 직렬 adder를 사용할 뿐 아니라, 각 단위연산소자의 출력 시점이 다름으로서 하나의 활성화 함수를 다수의 연산소자들이 공유할 수 있어, 곱셈기 자체의 크기 감소 이외에도 단위연산소자의 크기를 감소시킬 수 있게 되었다.

#### 4.2 Learning을 위한 단위연산소자의 구조

오류 역전파알고리즘을 사용한 신경망의 학습과정은 식(3)~식(6)으로 표현할 수 있다. 이들 식에 보듯 학습과정의 연산과정은 recall과정에 비해 훨씬 복잡하다.

$$\delta_k = OUT_k(1 - OUT_k)(Target_k - OUT_k) \tag{3}$$

$$V_{kj}(n+1) = V_{kj}(n) + \delta_k HOUT_j \tag{4}$$

$$\sigma_j = HOUT_j(1 + HOUT_j) \sum_k \delta_k V_{kj} \tag{5}$$

$$W_{ji}(n+1) = W_{ji}(n) + \sigma I_i \quad (6)$$

위의 수식에서 I, HOUT, OUT은 각각 입력, 중간층 출력, 출력층의 출력을 의미하며, Target은 출력층의 기대값, W, V는 중간층과 출력층의 가중치이다. 식(3)과 식(4)는 출력층의 가중치 update를 위한 식이고 식(5)와 식(6)은 중간층의 가중치의 update를 위한 식이다. 제안된 구조의 경우 각 층의 입,출력값의 저장을 위한 memory를 별도로 두어 data들이 서로 충돌하는 것을 방지하였는데 이는 식(2)와 식(5)에서 볼 수 있듯 입력 및 중간층의 출력은 모든 단위연산소자가 필요로 하는 data이고, 활성화 함수의 미분값은 식(4)와 식(5)에서 보듯 항상 자신의 neuron 즉 동일한 연산소자에서만 필요한 정보가 되므로 전자의 경우는 입출력 memory에 후자의 경우는 각 연산소자의 가중치 저장을 위한 memory에 저장해 놓으므로써 정보의 흐름을 효율적으로 운영할 수 있게 하였다. 식(5)와 식(6)의 update과정은 대상 가중치를 accumulator에 미리 loading한 후 가중치 memory에 저장되어 있는  $\delta$  혹은  $\sigma$ 와 입력 또는 중간층의 출력을 recall과정과 동일한 형식의 연산과정을 거쳐 계산한다. 또한 식(5)의 연산 과정은 learning과정에서 하드웨어 구현이 가장 어려운 것으로 알려져 있는데[2] 연산과정을 살펴보면 하나의 출력층 가중치는 모든 중간층 neuron에 관계된 가중치의 연산을 위해 필요하게 된다. 이러한 경우 제안된 구조는 파이프 라인을 사용하지 않고 가중치를 연산소자의 register와 I/O buffer를 통해 외부의 bus에 broadcasting한다. HANNIBAL의 경우 이러한 역방향의 정보전송 및 연산을 위하여 하드웨어의 효율을 급격히 떨어뜨리고 있는데 이는 파이프라인과 systolic구조상 정보 흐름이 반전 혹은 불규칙적인 경우 효율이 급격히 감소하는 원리와 동일한 원인에 의한 것이다[11].

### 4.3 하드웨어 구성

학습능력이 포함된 실제 단위연산소자의 block diagram이 그림 7에 나타나 있다. 가중치와 입력값의 크기는 참고문헌[12]와 software simulation에 의해 결정했으며, 가중치는 16bit, 입출력은 12bit을 사용하였다. 이는 일반적으로 알려진 학습을 위한 최소 정밀도보다 큰 것인데 정밀도에 의한 학습과정의 영향을 최소화 할 목적으로 결정된 것이다.

단위 연산소자 및 주요 하드웨어 구성은 다음과 같다.

- 7bit data pipe용 register : 1bit의 serial data와 6bit의 외부 instruction을 받아 다음 단위연산 소자로 이동시킨다.
- 16bit 곱셈기 : 1-D serial/parallel systolic 곱셈기[5]
- 27bit accumulator : accumulator를 위한 serial adder를 포함하며 shift, pre-loading, parallel Input/output이 가능함
- 16bit register : Memory와 단위연산소자의 정보교환
- compensator : accumulator의 연산도중 overflow 또는 underflow가 발생하면 이를 accumulator의 최대 혹은 최소 값으로 고정시켜 이러한 원인에 의한 시스템의 성능 감소를 최소화 한다.
- Local control block : 6bit 의 instruction을 받아 단위연산소자 내부의 제어신호를 발생시킨다.
- I/O buffer
- 가중치 저장을 위한 memory
- 활성화 함수 : 15segment PWL(Piece Wise Linear) 근사법<sup>[6][13]</sup>으로 구성된 sigmoid와 그의 미분값

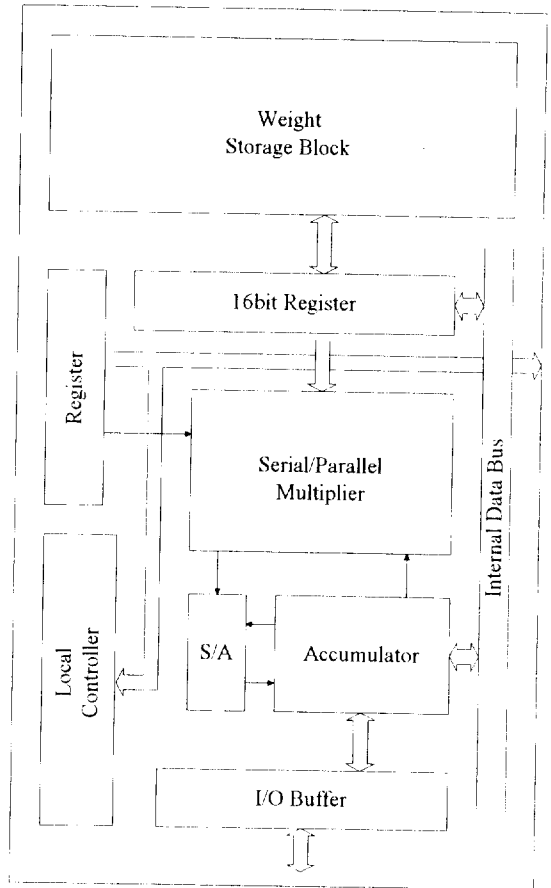


그림 7 단위 연산소자의 Block Diagram  
Fig. 7 Block diagram of node processor

이 외에도 신경회로망 전체를 제어하기 위한 system 제어부를 가지고 있으며, 이는 main memory와 program counter, sequence generator, register등과 flag, control table등을 가지고 있다. 실제 신경망을 주어진 하드웨어에 mapping하는 것은 control table에 coding된 16개의 16bit instruction set을 사용한 초기화 과정을 통하여 이루어진다.

### 5 결과 및 고찰

설계된 회로의 성능을 평가하기 위하여 2개의 입력층 neuron과 각각 4개의 중간층, 출력층 neuron을 가지는 신경회로망을 구성하고 간단한 비선형 매핑문제('XOR/Ex-NOR')에 적용하여 학습 및 recall과정을 Viewlogic을 사용하여 모의실험하였다(그림 8). 구성된 신경망 하드웨어는 4개의 출력층 XOR로 매핑되는 출력층 node의  $Target_k - OUT_k$  값이  $0.125(2^{-3})$  이하가 되면 학습이 종료되도록 하였으며, 약 420 회의 학습과정을 거친 후에 학습이 완료되었다. 그림 9에 학습과정에서의 학습진행에 따른 자승오차 및 최고오차의 나타나 있다.

또한 제안된 구조의 효율을 평가하기 위하여 식(1)과 proctime graph를 사용하였다. 제안된 구조와 직병렬 곱셈기를 사용할 경우 효율은 병렬 곱셈기의 사용시와 거의 동일하며 각 층의 크기에 따른 하드웨어 자원의 이용효율이 그림 10에 나타나 있다. 그림 10에는 HANNIBAL의 경우도 함께 나타나 있는

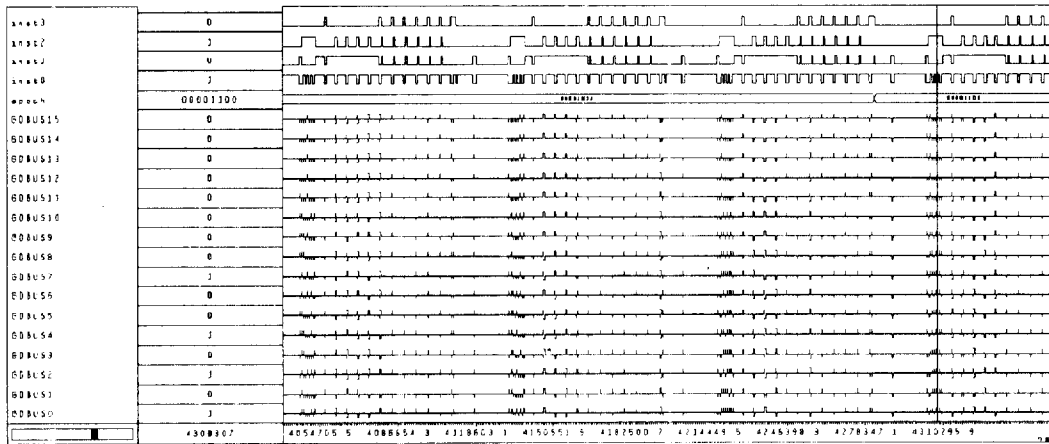


그림 8 설계된 신경망 하드웨어의 회로 모의실험 (학습과정의 일부)  
 Fig. 8 Logic Simulation of proposed digital neural network Hardware during learning

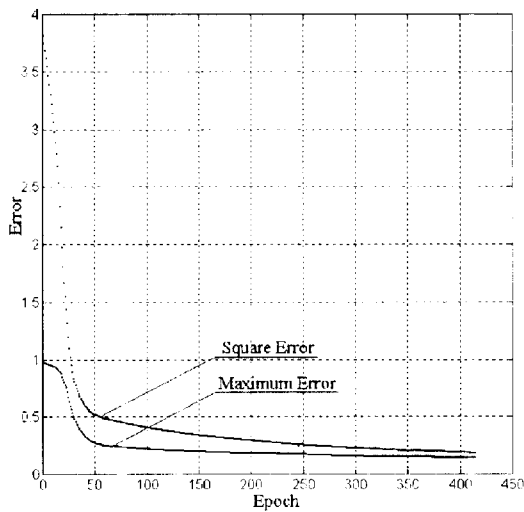


그림 9 설계된 신경망 하드웨어 학습과정의 자승오차 및 최고 오차  
 Fig. 9 Square and maximum error of the neural network hardware during learning

제 제안된 구조의 효율이 HANNIBAL의 경우보다 월등함을 알 수 있다. 또한 표 3의 곱셈기 비교에서 볼 수 있듯이 하드웨어 자원의 이용효율에 관한 영향을 배제한다 하더라도 가격대 성능비에서 직병렬 곱셈기를 사용한 경우가 우수하므로 제안된 구조와 직병렬 곱셈기를 사용하여 신경회로망을 구현할 경우 적은 비용으로 실시간 응용에 적합한 하드웨어를 구현할 수 있을 것이다.

### 6. 결 론

본 논문에서는 HANNIBAL이라 불리는 구조를 수정하여 가격대 성능비가 우수한 신경망 하드웨어 구조를 제안하고, 이를 구조적 관점에서 분석한 후 설계변수 중 가장 중요한 곱셈기의 선택을 위해 이들을 형태별로 분류, 분석 하였다. 이를

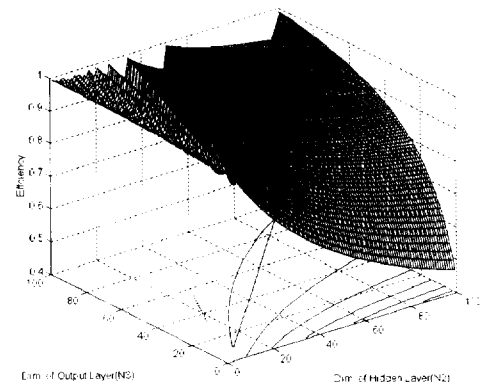
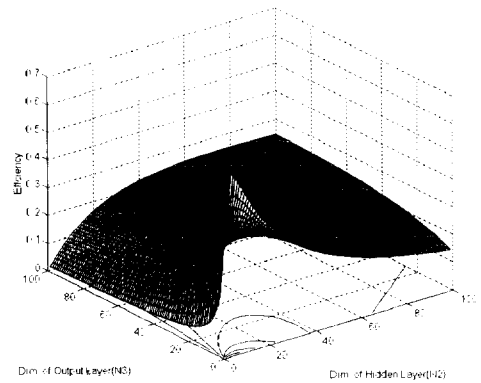


그림 10 HANNIBAL 및 제안된 구조의 하드웨어 사용효율 (학습과정)  
 Fig. 10 Hardware efficiency of HANNIBAL (a) and proposed (b) architecture during learning process

토대로 실제 신경망 하드웨어를 설계하고, 이의 성능을 모의실험을 통하여 입증하였다. 이는 신경망의 실시간 응용을 위한 하드웨어 구현에 있어 성능감소를 최소화하면서 비용을 절감할 수 있도록 하드웨어 자원의 이용효율을 최적화 함으로써 이루어

어졌다. 최근의 많은 연구 결과들로부터 학습속도 및 신뢰성이 우수한 많은 신경망 알고리즘이 개발되고 있는바, 이를 실제 응용에 적용하기 위한 전용하드웨어 개발이 절실히 요구되고 있다. 본 연구를 기반으로 하여 이러한 다양한 알고리즘을 수용할 수 있는 신경망 전용 하드웨어 개발이 이루어질 경우 신경망의 응용 범위 및 가능성은 좀더 확대될 수 있을 것으로 기대 된다.

본연구는 1994년도 인하대학교 연구비 지원에 의하여 수행되었음.

### 참 고 문 헌

- [ 1 ] Dan Hammerstorm, "A VLSI architecture for highperformance, low-cost, on-chip learning," IJCNN, pp.537-543, Feb. 1990
- [ 2 ] D.J. Myers, R.A.Hutchinson, " HANNIBAL : A VLSI building block for neural networks with on-chip backpropagation learning," Elsevier science pub., Neurocomputing, pp.25-37, May. 1993
- [ 3 ] David Naylor, Simon Jones, David Mayers, "Backpropagation in linear arrays - A performance analysis and optimization," IEEE Trans. on Neural Networks, vol.6, no.3, pp.583-595, May. 1995
- [ 4 ] Shlomo Waser. "High-speed monolithic multiplier for real time digital signal processing," IEEE COMPUTER, pp. 19-29, Oct. 1978
- [ 5 ] I-Chen. Wu, "A fast 1-D serial-parallel systolic multiplier," IEEE Transactions on computers, vol.C-36, no.10, pp. 1243-1247, OCT.1987
- [ 6 ] D.J. Myers, R.A.Hutchinson, "Efficient implementation of PieceWise Linear activation function for digital VLSI neural networks," Electronic Letters, vol.25, no.24, pp. 1662-1663, Sep. 1993
- [ 7 ] 모상만, 윤용호, "2의 보수 직병렬 승산기의 설계 및 분석," 대한전자공학회 논문지, 제29권 B편, 6호, pp.181-185, 1992년 3월
- [ 8 ] S.-M.Moh and S.-H.Yoon, "Serial-parallel multiplier for two's complement numbers," Electronic Letters, vol.31. no.9, pp.703-704, April. 1995.
- [ 9 ] S.Sunder, F.El-Glibaly, A.Antoniou, "Two's-complement fast serial-parallel multiplier," IEE Proc.-Circuits Devices System, vol.142, no.1, pp.41-44, Feb. 1995
- [ 10 ] D.Ait-Boudaoud, M.K.Ibrahim, B.R.Hayes\_Gill, "Novel cell architecture for bit level systolic arrays multiplication," IEE Proceedings-E, vol.138, no.1, pp.21-26, Jan. 1991
- [ 11 ] H.T.Kung, "Why systolic architectures ?," IEEE COMP UTER, pp.37-46, Jan. 1982
- [ 12 ] Cesare Alippi, Meyer E.Nigri, "Hardware requirements for Digital VLSI Implementation of Neural Networks," IEEE IJCNN, pp.1873-1878, 1991
- [ 13 ] 김진태, 정덕진, "Digital 신경회로망을 위한 비선형 함수의 구현," 전기공학회 춘계학술대회 논문집, pp.501-503, 1993
- [ 14 ] David Naylor, Simon Jones, "Improving neural hardware performance with structural design guidelines," IEEE Proc. of ICNN, pp.1969-1973, 1994

### 저 자 소 개



#### 이 범 엽 (李範燁)

1968년 8월 11일생. 1991년 인하대 공대 응용물리학과 졸업. 1992년~1994년 한국전력 기술(주) 근무. 1996년 현재 인하대 대학원 전자재료공학과 석사과정



#### 정 덕 진 (鄭德鎭)

1948년 2월 8일생. 1970년 서울대 공대 전기공학과 졸업. 1984년 미국 UTAH 주립대학교 전기공학과 졸업(석사). 1988년 미국 UT AH대학교 전기공학과 졸업(박사). 1996년 현재 인하대 공대 전자재료공학과 부교수