Binary Convolution을 이용한 고속 디지탈 신경회로망의 VLSI 설계

VLSI Design of High Speed Digital Neural Network using the Binary Convolution

崔 承 鎬*, 金 榮 民**
(Seung Ho Choi*, Young Min Kim**)

요 약

현재 신경회로망의 구현에 관한 여러가지 연구가 진행되고 있으며, 이들 중 신경회로망의 VLSI 구현에 대한 연구가 매우 활발하다. 디지탈 신경회로망은 느린 처리속도와 넓은 면적을 차지하는 점이 주요 단점으로 지적되는데 본 논문에서는 neural cell을 곱셈과 덧셈을 Binary Convolution 기법과 Counter를 사용하여 설계함으로써 속도를 높이고 단위 뉴런의 소요 Tr수를 줄여 그 소요 면적을 줄이도록 하였다. 본 cell의 구조를 이용하여 layer당 16개씩의 cell을 가지는 3-layer neural network을 구성하였을 경우 0.8μ standard cell 설계시 50MHz까지 동작하였으며 26MCPS의 동작을 확보하였다.

ABSTRACT

Recently, for implementation of neural networks extensive studies have been done and especially VLSI technology has been regarded as the one of the most attractive means to implement neural networks. The main drawbacks of digital VLSI implementations are their large area and slow processing speed. In this paper to solve the speed and size problems we designed the efficient architecture using the binary convolution method for basic operation of neural cell, multiplication and addition. When it is used for implementing 3-layer neural network with 16 neural cell per layer that used neural cell based on binary convolution, clock speed of 50MHz and 26MCPS on 0.8μ standard cell library has been achieved.

I. 서 론

신경회로망은 기존의 컴퓨터에서 효과적으로 수행하기 힘든 영상인식, 음성인식, 적응제어, 최적화 등에 탁월한 기능을 가지고 있다. 데이터의 처리에 있어서 병렬 분산처리 방식이 사용되어 수많은 뉴런에 분산 저장된 정보가 동시에 병렬 처리된다. 신경회로망의 데이타 저장 방식은 CAM(Content Addressable Memory)이라는 방식으로, 이것은 이용하고자 하는 정보에 대한 주소를 찾아 그 정보를 이용하는 보통의 방식과는 달리 정보의 일부 또는 관련된 안식를 가지고 전체를 찾아내는 방식이다. 그리고 신경회로망은 학습능력을 가지고 있어 스스로 정보를 체계화할 수 있다. 이러한 신경회로망을 개발하므로써 적응 능력이 부족한 기계가 인간을 대신하지 못하였던 많은 응용분야에서 큰 역할을 수행할 수 있을 것으로 기대된다.

신경회로망이 실제 문제에 응용되기 위해서는 실시간 처리가 가능해야 하며, 이의 핵심 기술이 바로 신경회로망 모델의 효율적인 VLSI 구현이라 할 수 있다. 현재 신경회로망은 그 구조가 H/W로 구현될 경우, 병렬 및 동시처리의 장점을 최대한 살릴 수 있으며 그 응용 가능성이 크게 확장될 것이다.

신경회로망의 VLSI 구현에는 아날로그 방식과 디지탈 방식이 있는데, 디지탈 방식으로 구현할 경우 고속, 다중화 접속이 가능하며 잡음에 강하고, 또한 학습 및 구조 변경이 아날로그 방식보다 쉽다. 현재 많이 사용되는 신경회로망의 시냅스들은 곱셈기와 덧셈기를 거쳐 주어진 전달함수를 통과하도록 모델링된다. 이러한 신경회로망의 디지탈 구현은 성숙되고 잘 알려진 기술을 기본으로 하기 때문에 그 유연성, 정확도, scale 가능성은 아날로그 구현에 비해 매우 월등하다.[1]

하지만 이러한 디지탈 구현의 주요 단점은 많은 silicon 면적을 차지하고, 상대적으로 병렬 처리율이 낮아 저속 동작을 하며 processing unit들을 상호 연결하는데 비용이 많이 든다는 단점이 있다. 이러한 문제들은 여러 가지 새로운 neuron 구조 및 network 구조를 개선하므로써 상당부분 해결이 가능하며, 다층 구조의 neuron을 구성함에 있어서 한 개의 chip에 많은 수의 neuron을 집적하기 위해서는 필연적으로 단위 neuron의 크기를 줄여야 한다.[2]

본 논문에서는 Binary Convolution 기법을 사용하여

*동신대학교 정보통신공학과 교수
**전남대학교 전자공학과 교수
접수일자 : 1996년 2월 7일

곱셈과 덧셈 연산을 한 개의 counter로 구현함으로써 그 크기를 최소화하고 불필요한 연산을 제거하여 빠른 속도를 얻도록 한 뉴런 cell을 제안한다. 또한 이를 바탕으로 설계된 3-Layer neural network를 설계하고 그 전체적인 동작속도를 simulation 한다.

Ⅱ. 디지탈 Neural Network의 설계

1. Neural cell의 설계

1.1 Binary Convolution

입력 신호와 weight의 곱셈 연산을 위해 Binary Convolution을 이용하였다. 입력 신호와 weight를 각각 8 bits라고 할 때, 입력 신호 I(x)와 weight W(x)는,

$$I(x) = i_7x^7 + i_6x^6 + \ldots + i_0$$
$$W(x) = w_7x^7 + w_6x^6 + \ldots + w_0 \quad (1)$$

으로 표시되는 두 식으로 나타낼 수 있으며 이 두 식의 곱셈 결과인 Y(x)는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} Y(x) &= I(x)\cdot W(x) \\ &= i_7w_7x^{14} \\ &\quad +(i_6w_7 + i_7w_6)x^{13} \\ &\quad +(i_5w_7 + i_6w_6 + i_7w_5)x \\ &\qquad \vdots \\ &\quad +(i_1w_0 + i_0w_1)x \\ &\quad +i_0w_0 \end{aligned} \quad (2)$$

이 식에서 계수를 구성하는 i와 w사이의 곱은 논리곱(AND)이며, 이 논리곱들의 값이 산술적으로 더해져 각 차수의 계수를 나타낸다. 따라서 각각의 차수에 해당하는 계수는 AND gates와 한 차수에 대해 여러 개의 bits를 더함으로써 발생될 수 있는 carry를 상위 차수로 옮겨 처리할 수 있는 특수한 구조의 counter를 이용하여 구할 수 있다. 앞의 식에서 계산 결과는 carry를 포함하여 모두 16 bits이데 본 논문의 counter는 이중 낮은 차수의 8 bits를 소거시킴으로써 곱셈 결과에 큰 영향을 주지 않고 불필요한 계산을 줄여 동작 속도를 높일 수 있다.

따라서 Y(x)는

$$\begin{aligned} Y(x) &= i_7w_7x^{14} + (i_6w_7 + i_7w_6)x^{13} + \cdots \\ &\quad (i_1w_7 + i_2w_6 + \cdots + i_7w_1)x^8 \end{aligned} \quad (3)$$

으로 나타내어진다.

그림 1.은 binary convolution을 위한 회로의 블럭도 이다. 8 bits의 입력 신호와 weight를 각각의 레지스터로 동시에 불러들여 저장한다. 하위 8 bit에 입력 신호가 저장된 15 bit의 입력 레지스터는 제어 신호에 의해 8 clock마다 상위 bit 쪽으로 입력값을 shift하면서 weight와 AND 연산이 수행되도록 한다. 따라서 입력 신호를 저장한 입력 레지스터의 초기 상태는 i_{14}~i_0의 레지스터의 공간 중에서 i_7~i_0까지만 입력값으로 채워져 있으며 AND 게이트와 연결된 부분은 i_{14}~i_7까지이다.

가장 먼저 '000000i_7'과 '$w_0w_1w_2w_3w_4w_5w_6w_7$'가 bit별로 AND 연산이 수행된다. 그 결과는 '000000(i_7w_7)'이 되며, 이 결과의 각 bit들의 합이 식 (3)에서의 차수 14의 계수를 나타낸다. 두 번째 제어 신호에 의해 입력 레지스터는 1 bit를 shift한다. 다음으로 AND 연산이 수행되는 값은 '00000i_7i_6'과 '$w_0w_1w_2w_3w_4w_5w_6w_7$'이고, 그 결과는 '00000(i_7w_6)(i_6w_7)'이 된다. 결국 한 입력 신호에 대해 같은 과정

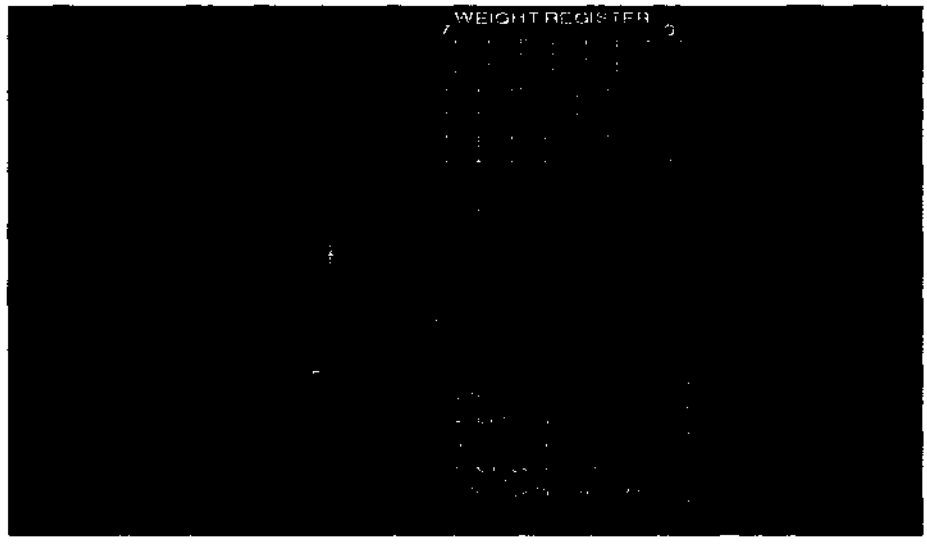


그림 1. Binary convolution의 블럭도
Fig 1. Block diagram of binary convolution

을 모두 7번 반복하여 식 (3)에서의 모든 차수에 대한 계수를 구하게 된다. 그리고 이 계수들은 뒤에 설명되는 counter에 의해 적절히 가산되어 진다.

1.2 단위 Neural Cell의 설계

그림 2.는 단위 neural cell의 구조를 보인 것이다. 입력 레지스터는 load 신호에 의해 입력 신호를 저장하고, 8 clock마다 발생되는 shift 신호에 의해 입력 신호를 shift 한다. 두 신호가 동시에 발생되는 경우 shift 신호는 무시된다. shifter1은 8 bits의 입력을 serial bit stream으로 바꾼다. 입력 레지스터의 shift가 이루어지고 난 바로 다음 clock에서 AND gates를 통과한 값들을 load하고, 8 clock 동안 8 bits의 값을 shift하면서 내보낸다. 이 동작은 입력 레지스터와 동일하다.

그림 3.은 그림 2.의 counter의 schematic diagram이다. 설계된 counter는 'Enable counter'로서 차수가 변할 때마다 enable의 위치가 변하도록 설계되어 있다. 1.1절에서 설명된 바와 같이 가장 높은 차수의 계수부터 계산되므로 counter의 중간에서 count를 시작하여 차수가 낮아질 때마다 하위 bit로 내려가며 count를 한다.

shifter2는 '1'을 shift하여 채워가면서 counter의 enable 위치를 조절해 주는 역할을 한다. shifter1에서 매 clock마다의 출력에 대해 shifter2는 현재의 bit stream이 어느 차수의 것인가에 따라 counter의 enable 위치를 결정한다. 즉, shifter1의 첫 입력 신호는 그림 1.에서와 같이 차수 14와 i_7w_7만이 의미 있는 값이며 차수 14의 bit stream중 가장 먼저 shifter1에서 출력된다. 이 때 shifter2는 counter의 7개의 enable line중 enable[6]에 해당하는 bit에 '1'을 저장하고 차수 14의 bit stream이 끝날 때까지 이 상태를 유지한다.

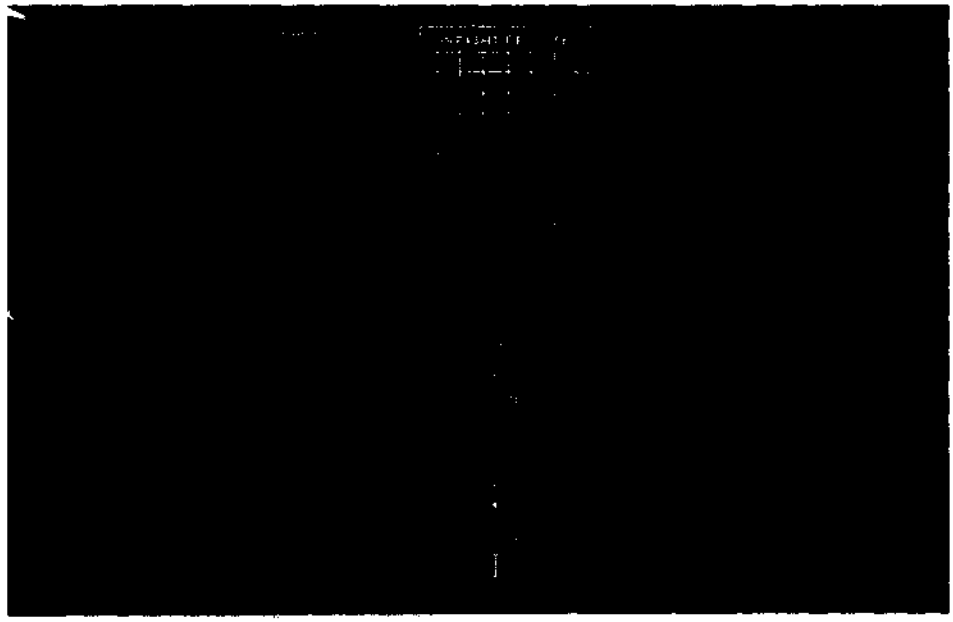


그림 2. Neural cell의 블럭도
Fig 2. Block diagram of neural cell

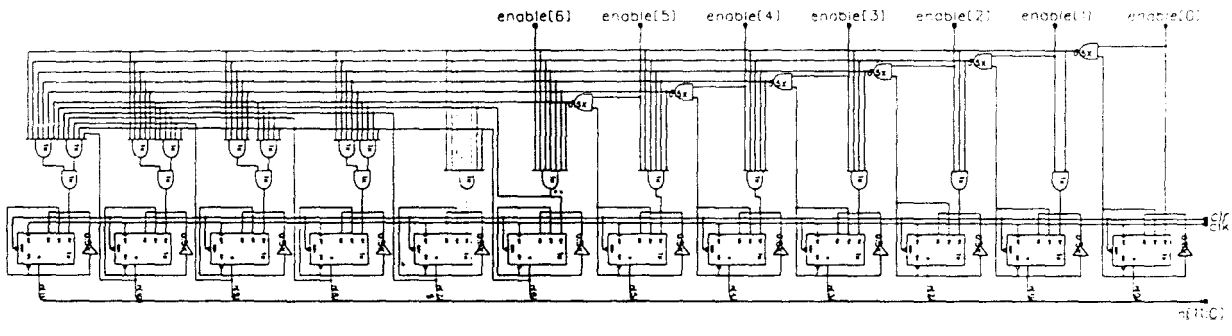


그림 3. Counter의 설계도
Fig 3. Schematic diagram of counter

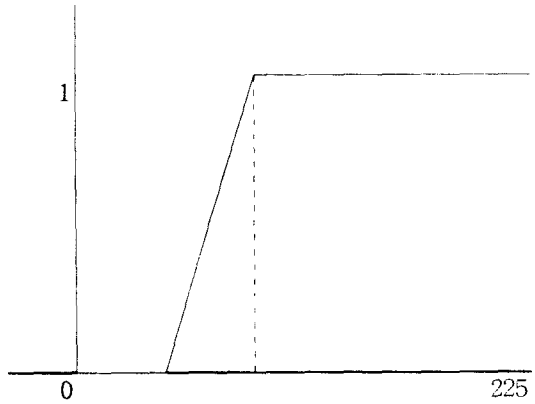
만약 i_7w_7의 값이 '1'이면 shifter2에 의해 counter의 enable[6]은 ON이 되고 나머지 enable[5]~[0]은 OFF된다. 따라서 counter는 7 번째 bit에 '1'을 사상하게 된다. 다음으로 두 번째 bit stream은 차수 13에 해당되며 i_6w_7과 i_7w_6의 합이 계수가 된다. shifter2는 이전 상태에서 1 bit shift하여 enable[5], [6]에 해당되는 bit들에 '1'을 가진다. 그림 2.와 그림 3.에서와 같이 i_6w_7과 i_7w_6의 값에 따라 counter의 enable[5]와 enable[6]이 동시에 ON이 되며 6 번째 bit부터 count된다. counter는 차수 14부터 8까지의 계수들을 이상과 같은 방법으로 계산한다. counter의 bit[6:0]은 enable에 의해서 값이 변하며 나머지 bit[11:7]은 bit[6:0]의 계산 결과의 carry에 따라 값을 갖는다.

한 입력 신호와 weight의 곱에 대해 이상과 같은 동작을 수행한 후 입력 레지스터와 weight 레지스터는 다시 새로운 값을 load하며 shifter2는 지금까지 shift된 '1'을 clear한다. 새로운 입력 신호와 weight에 대해 지금까지의 동작을 반복 수행하며 counter는 이전의 값에 연속해서 count를 수행하여 각각의 입력 신호와 weight의 곱을 더한다. 그리고 counter는 모든 입력 신호에 대한 계산이 끝나면 상위 8 bit의 계산 결과를 sigmoid 함수를 통과시켜 출력 레지스터로 보낸 후 clear되어 초기 상태로 돌아간다.

그림 4.는 그림 1.의 Sigmoid function block의 구성을 보인 것이다. Sigmoid 함수 부분은 counter의 출력이 통과하는 비동기 회로이다. 신경회로망의 전달 함수 구현을 위해 사용되는 함수는 여러 가지가 있으며 본 논문에서는 이들 중 비선형 Sigmoid 함수를 사용하였으며 이의 구현을 위해 그림 4.의 근사적인 비선형 Sigmoid 함수 회로를 이용하였다.

0	00000000	
1	00000001	
2	00000010	'0'
	⋮	
96	01100000	00000000
97	01100001	00000100
	⋮	
158	01111111	01111100
159	10000000	
	⋮	'1'
255	11111111	

(a)



(b)

그림 4. Sigmoid 함수의 구성
Fig 4. Sigmoid transfer function

2. Neural Network의 구성

본 논문에서는 이상에서 설명된 neural cell을 이용하여 3-layer의 신경회로망을 구성하였다. 신경회로망의 3-layer 중 입력층에 대해 본 논문에서는 하나의 입력 레지스터로 입력 신호를 순차적으로 처리하도록 하였다. 은닉층(Hidden layer)에 n개의 neural cell을 둔다. 이 cell들은 한 입력값에 대해 각각 다른 weight를 가지며 또 입력에 따라서 weight가 달라진다. 이 weight는 외부 RAM에 저장하였다가 제어 신호에 의해 레지스터로 불러온다. 이미 앞에서 설명한 바와 같이, m개의 입력값들에 대해 m개의 cell에서의 계산이 끝나면 각각의 결과들을 출력층으로 보낸다. 이 때 각각의 결과들은 MUX를 통해서 출력층으로 순차적으로 load되며 은닉층의 각 cell들의 출력 레지스터들은 현재 자신이 저장하고 있는 값이 출력층으로 load될 때까지 계속 그 값을 유지한다. load된 각 결과에 대해서 출력층에서도 은닉층과 같은 동작을 반복한다.

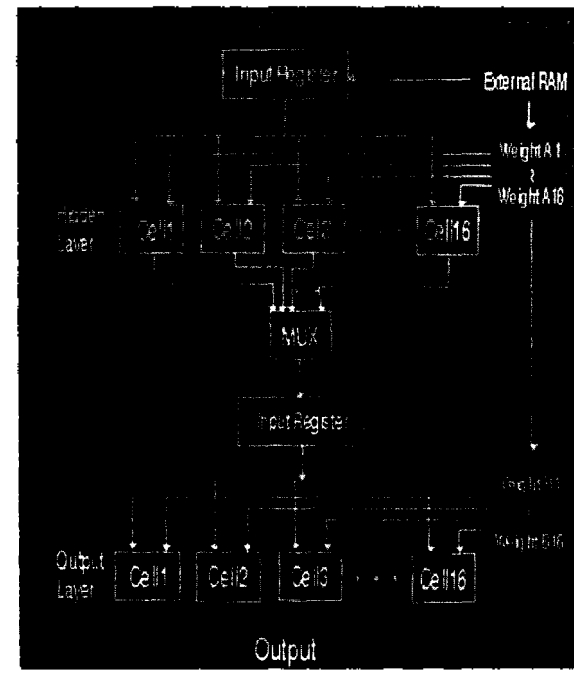


그림 5. 설계된 신경회로망의 블럭도
Fig 5. Block diagram of Neural Network

2.1 Timing Diagram

그림 6.은 neural cell의 timing을 보인 것이다. CLR 신호에 의해 cell을 clear시킨 후, 가장 먼저 i/w load 신호에 의해 입력 신호와 weight를 load한다. 이 때 입력 레지스터에서 입력 신호를 shift시키는 inshift 신호도 동시에 발생되는데 입력 신호는 8 clock 마다 1 bit씩 각각 bit을 shift 시키며, 앞에서도 이미 언급한 바와같이 load 신호가 발생될 때에는 이 신호는 무시된다. 각각의 레지스터로 load되고 나면, 바로 입력 신호와 weight의 AND operation결과가 ldc/sft 신호에 의해 shifter1에 load된다. 이 신호도 역시 8 clock마다 1 번씩 발생되며 shifter2에서 '1'을 shift하도록 하는 신호로도 사용된다. 이 신호사이의 8 clock은 shifter1에서 shift가 수행되는 시간이며 따라서 shifter1의 shift를 일으키는 신호는 CLK가 된다. 이와같은 동작이 계속 반복되어 7번 수행되므로 한 입력 신호와 weight의 곱을 위해 걸리는 시간은 모두 56 clock이 걸리며 이 과정도 역시 일정하게 반복되어 일어난다. 여기에 추가되는 신호로는 56 clock 단위로 발생되는 shifter2_clr 신호가 있는데, 이 신호는 다음의 입력 신호와 weight의 곱을 위해서 shifter2를 clear시킨다. 또 모든 입력 신호와 weight의 곱이 끝나면 출력 레지스터로 counter의 값을 load하는 load_z 신호가 있으며 이 신호는 56×16 clock마다 한 번씩 발생되며, 이 신호에 의어서 counter를 clear시켜 주는 counter_clr 신호가 발생된다.

2.2. Simulation 결과 및 Performance

그림 5.에서 보인 바와 같이 한 층에 16개의 cell을 가지고 있는 신경회로망을 simulation한 결과를 그림 7.에 보였다. 여러 simulation data중에서 입력 신호와 weight가 모두 '11111111'인 경우를 simulation한 것으로 앞의 timing에서 설명하였듯이 8 clock 단위의 ldc/sft 신호와 56 clock 단위의 i/w load 신호 및 shifter clr 신호에 의해 counter 또한 제대로 count가 수행됨을 n[11:0]을 통해 확인할 수 있다. 또한 16개의 입력 신호와 weight를 계산한 후 sigmoid 함수를 통과한 16개의 출력 신호 중의 하나가 ON 되었음을 out[7:0]에서 보이고 있다. simulation은 VLSI Tech.사의 COMPASS Tool을 사용하였으며 0.8μ표준 CMOS 기술로, simulation 환경은 Vdd = 4.65, Vss = 0.1V, 온도는 70℃의 조건하에서 이루어졌다.

그림 8.은 COMPASS에서 설계된 신경회로망의 구조를 보인 것이다. simulation 결과 속도면에서 현재 설계된 신경회로망은 50MHz의 속도로 동작하며, 단위 뉴런당 580개의 게이트를 가지는 32개의 뉴런 cell들에 대해 약 20,000개의 게이트(약 80,000 TRs) 정도를 가지면서 26 MCPS(CPS : interconnection per second)의 동작 속도를 확보하였는데 이 속도는 layer당 cell의 수에 비례하여 빨라질 수 있다. 그러므로 이것이 100만 게이트 수준으로 확장되어 설계될 경우 1,600개의 뉴런을 두고 약 1.3 GCPS 정도의 속도를 예상할 수 있다. 이러한 performance는 지금까지 발표된 여러 논문들 중 일본의 HITACHI사에서 설계한 디지탈 신경회로망 모델이 그 면적을 최소화하기 위하여 단위 뉴런당 하나의 곱셈기를 사용하고도 약 1,000개의 게이트를 가지며 267μs동안 576개의 뉴런이 연결된 것과 비교될 수 있다.[4]

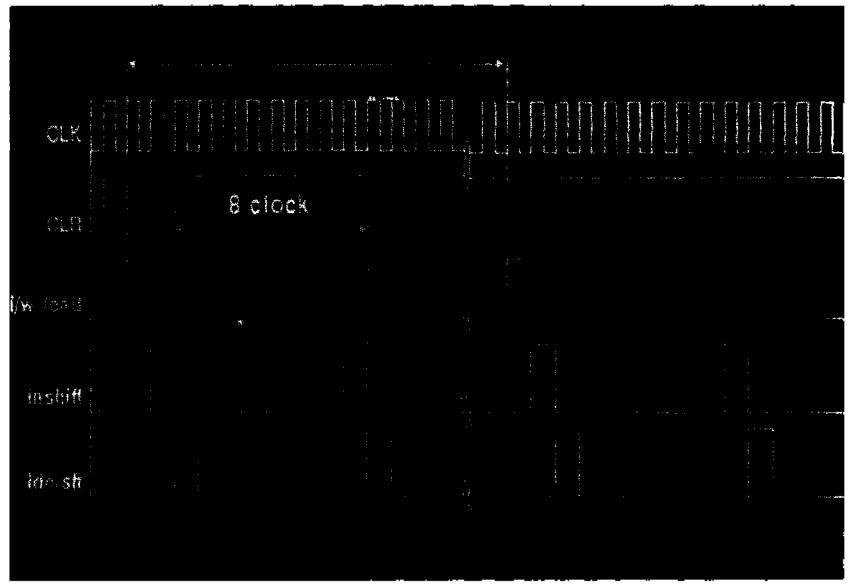


그림 6. Timing diagram
Fig 6. Timing diagram

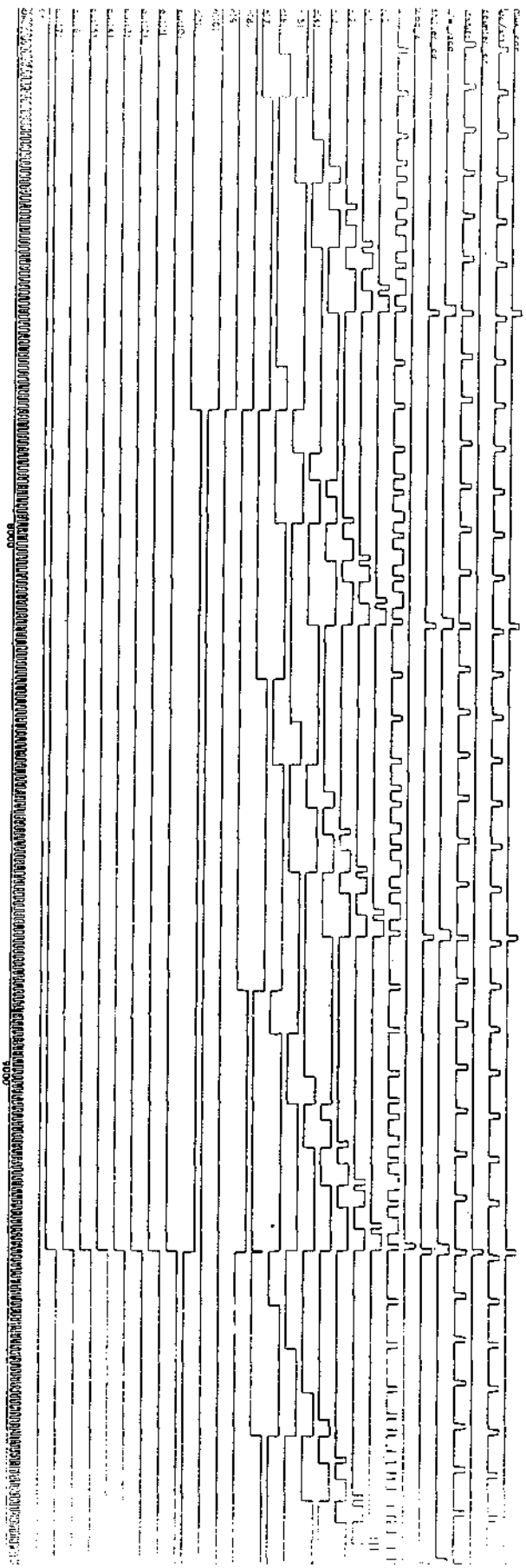


그림 7. Simulation 결과
Fig 7. Simulation result

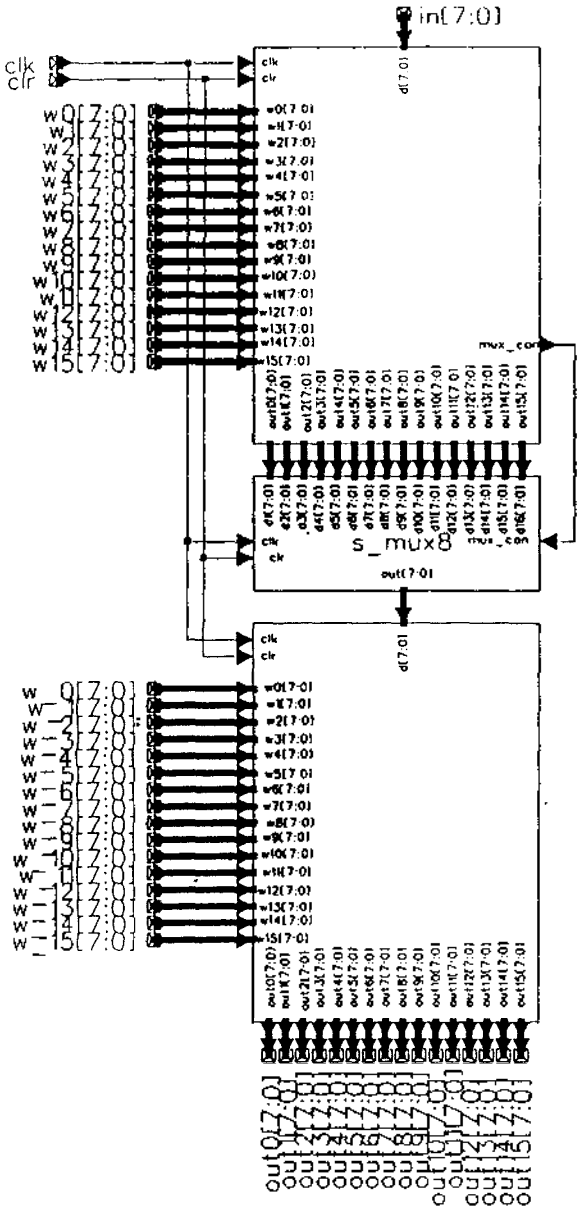


그림 8. 신경회로망의 구조
Fig 8. Architecture of neural network

Ⅲ. 결 론

신경회로망에 대한 디지탈 방식의 접근은 아날로그 방식의 접근에 비해 그 구현이 용이하다는 장점을 가지고 있다. 하지만 현재까지 연구된 디지탈 방식의 뉴런의 구조는 대부분 단위 뉴런당 다수의 빠른 속도가 요구되는 MAC 구조를 사용하는데 신경회로망에 이용 가능한 하나의 8-bit MAC 구조는 적어도 1,000개 이상의 트랜지스터를 필요로 하기 때문에 이는 실제 응용에 있어서 많은 뉴런을 필요로 하는 신경회로망의 칩 면적을 크게 하는 요인이 된다.[5][6]

본 논문에서는 신경회로망의 응용에 있어서 아날로그와 비교한 디지탈 신경회로망의 단점으로 지적되는 속도 문제와 칩 넓이에 관한 문제의 해결을 위하여 Binary convolution을 이용한 곱셈 연산 구조를 고안하여 디지탈 신경회로망의 새로운 구조를 설계하고 시뮬레이션 하였다. 설계된 신경회로망의 뉴런은 MAC 구조를 이용한 것과 비교할 때 전체 데이타에 미치는 영향이 적은 부분의 연산을 제거하여 동작 속도를 높일 수 있으며, 또한 덧셈기 부분을 하나의 카운터로 구성하여 하나의 뉴런 cell에 대해 580개의 게이트만이 필요하므로 이를 이용하여 디지탈 신경회로망을 구성할 경우, MAC을 이용하여 설계된 다른 디지탈 신경회로망에 비해 속도에 영향을 주지 않고 그 면적을 많이 감소시킬 수 있다. 따라서 단위 뉴런비 속도가 보다 향상되게 된다.

본 논문에서 제안한 Binary Convolution 구조를 디지탈 신경회로망의 구현에 이용할 경우, 칩의 performance를 높이면서 그 면적에 영향을 주지 않는 디지탈 신경회로망의 구현이 가능할 것이다. 그러나 보다 완전한 신경회로망의 구현을 위해서 전달 함수의 회로구현에 대한 연구가 필요하고, 이러한 신경회로망 cell들을 모듈화할 수 있도록 하는 등의 연구가 필요하다.

참 고 문 헌

1. Philip D. Wasserman, "Neural Computing-Theory and Practice", VNR.
2. James A. Freeman, David M. Skapura, "Neural Networks-Algorithms, Applications, and Programming Techniques", Addison-Wesley.
3. 선계오, 김영민, "Digital Neural Network based on Binary Convolution", Proceedings of KITE Summer Conference '93, vol.16, pp.562-565, 1993.
4. Moritoshi Yasunaga et al., "Design, Fabrication and Evaluation of a 5-inch Wafer Scale Neural Network LSI Composed of 576 Digital Neurons," Proceedings of the International Joint Conference on Neural Networks, June, 1990.
5. Edgar Sanchez-Sinencio, Clifford Lau, "Artificial Neural Networks-Paradigms, Applications, and Hardware Implementations", IEEE Press.
6. Veljko Milutinovic, Paolo Antognetti, "Neural Networks: Concepts, Applications, and Implementations," Prentice Hall, vol.2.

▲金 榮 民(Yeong Min Kim) 1954년 4월 18일생

1976년 2월 : 서울대 전자공학과 졸업(학사)
1978년 2월 : 한국과학기술원 전기 및 전자공학과 졸업(석사)
1986년 : 오하이오 주립대학교 전자공학 박사학위 취득
1987년 1월~1988년 5월 : 노스캐롤라이나 A&T 주립대학 전자과 조교수
1988년 6월~1991년 8월 : 한국전자통신연구소 실장
1991년 9월~현재 : 전남대학교 전자공학과 교수
※주관심분야 : 영상압축, 신호처리, DSP, 초고속통신망

▲崔 承 鎬(Seung Ho Choi)
현재 : 동신대학교 정보통신공학과 교수(제 13권 1호 참조)