

論文94-31B-5-15

# 신경회로망을 이용한 ATM 교환기의 제어부 설계 (Design of an ATM Switch Controller Using Neural Networks)

金榮禹\*, 林寅七\*

(Young Oo Kim and In Chil Lim)

## 要約

본 논문에서는 신경회로망을 이용하여 입력 버퍼형 ATM(Asynchronous Transfer Mode) 교환기의 출력 조정논리부를 설계한다. 충돌망을 사용하는 ATM 교환기에서 충돌을 피하기 위해서는 교환이 요구되는 ATM셀들을 입력 버퍼에 일시 저장하고 교환 순서를 스케줄링하여야 한다. 스케줄링 단계에서 조직적으로 내부충돌을 제거하기 위하여  $N \times N$  교환 요구행렬을  $N/16$ 개의 부분행렬로 분할하고,  $N/4$ 개의 군으로 나누어 군마다 교번하여 출력경합을 행하며, 입력 버퍼의 윈도우 크기를 증가시키면 한 시간슬롯 동안에 교환되는 셀의 수는  $N$ 에 근접하는 교환율을 보인다. 상호 충돌이 일어나지 않는 셀들의 선택은 신경회로망을 통해 행해지며,  $N/4$ 개의  $4 \times 4$  신경회로망 모듈이 동시에 동작하여, 신경회로망의 대량의 병렬성을 통한 빠른 선택이 가능하게 된다. 본 논문에서 사용된 신경회로망은  $4N$ 개의 뉴런 및  $14N$ 개의 연결선을 가진다. 제안된 조정논리부는 C언어로 구현하였으며, 실험 결과를 통해 그 유효성을 입증한다.

## Abstract

This paper presents an output arbitrator for input buffering ATM(Asynchronous Transfer Mode) switches using neural networks. To avoid blocking in ATM switches with blocking characteristics, it is required to buffer ATM cells in input buffer and to schedule them. The  $N \times N$  request matrix is divided into  $N/16$  submatrices in order to get rid of internal blocking systematically in scheduling phase. The submatrices are grouped into  $N/4$  groups, and the cells in each group are switched alternatively. As the window size of input buffer is increases, the number of input cells switched in a time slot approaches to  $N$ . The selection of nonblocking cells to be switched is done by neural network modules.  $N/4$  neural network modules are operated simultaneously. Fast selection can be achieved by massive parallelism of neural networks. The neural networks have  $4N$  neurons and  $14N$  connections. The proposed method is implemented in C language, and the simulation result confirms the feasibility of this method.

## 1. 서론

\* 正會員, 漢陽大學校 電子工學科  
(Dept. of Elec. Eng., Hanyang Univ.)  
接受日字 : 1993年 7月 16日

광대역종합통신망(Broadband Integrated Services Digital Networks : BISDN)은 음성, 데

이다. 화상 등과 같은 다양한 서비스를 제공하는 차세대 통신망이다. BISDN에서 제공되는 서비스는 음성, 영상, 데이터 등으로 구성되는 다중 매체 서비스의 특징을 가지고 있으며, 교신성 서비스와 분배형 서비스가 공존하고, 대역폭 및 사용 시간이 광범위하게 분포하고 있을 뿐 아니라, 트래픽의 특성이 연속적이거나 군집성인 서비스가 공존하는 다양한 특성을 가지고 있다.<sup>17)</sup> 따라서 BISDN에서는 각 서비스들을 일원화한 통합 수용 방식이 요구되며, 미래의 새로운 형태의 서비스에 대한 융통성있고 효율적인 전송방식이 필요하다. 이러한 여러가지 요구사항을 충족시키기 위해 CCITT에서는 비동기식 전달모드(Asynchronous Transfer Mode : ATM)를 권고하기에 이르렀다.<sup>18)</sup>

ATM은 비동기식 시분할 다중화를 사용하는 특수한 형태의 패킷형 전달 방식으로서 서비스 정보들을 일정한 크기로 절단한 후 53 바이트 크기를 갖는 ATM셀에 매핑하여 다른 ATM셀들과 비동기식 시분할 다중화함으로써 BISDN의 내부 신호를 형성한다. 이때 ATM셀의 첫 5 바이트는 ATM셀의 올바른 전달을 제어하기 위한 각종의 정보를 수록하며, 나머지 48 바이트는 서비스 정보를 수용한다.<sup>19)</sup>

광기술의 발달로 말미암아 망에서의 전송 속도는 초당 수 기가 비트 이상의 전송 능력을 갖추고 있다. 그러나 ATM교환기의 속도는 이에 훨씬 못미치는 수준에 머물고 있어 ATM교환 기술에 대한 연구가 활발히 진행되고 있다.<sup>20)</sup>

ATM교환기에는 셀들간의 충돌을 방지하여 수율(throughput)을 높이기 위해 버퍼를 설치한다. 이때의 버퍼의 배치 방식에 따라 입력큐잉(input queuing), 출력큐잉(output queuing), 공통큐잉(central queuing) 방식으로 나뉘게 된다.<sup>18)</sup> 공통큐잉 방식과 출력큐잉 방식은 높은 수율을 제공할 수 있으나, 입출력의 수가  $N$ 일 경우, 버퍼의 액세스 속도는  $N$ 배 이상의 고속동작을 하여야 한다는 문제점이 있다. 반면, 입력큐잉 방식은 버퍼의 속도는  $N$ 에 무관한 저속동작으로 기능하나, 버퍼가 FIFO(First-In First-Out)로 동작하는 경우 그 선단 셀만이 출력측에 전달될 수 있으므로 이에 따른 HOL(Head-Of-Line) 충돌(blocking, contension)이 문제점으로 대두되어 최대 수율(throughput)이 0.586에 불과하게 된다.<sup>21)</sup> 따라서 입력버퍼를 FIFO형이 아닌 FIRO(First-In Random-Out)형 버퍼를 사용한 교환 방식이 제안되었다.<sup>2,4,5,14,16)</sup>

본 논문에서는 FIRO형 입력버퍼 방식의 ATM교환기에서의 출력경합회로를 신경회로망을 이용하여

설계한다. 교환구조물(switching fabric)로는 반얀망(Banyan network)을 사용한다. 반얀망은 내부충돌(internal blocking)이 존재하는 망이므로 출력경합기는 내부충돌이 일어나지 않는 셀들의 집합중 하나를 선택할 수 있는 기능을 가져야 한다. 이러한 문제는 TSP(Traveling Salesman Problem)와 유사한 문제가 되므로 신경회로망의 장점인 대량의 병렬성을 이용하면 좋은 결과를 얻을 수 있다. 또한 각 셀들의 내부충돌을 분산시키도록  $N \times N$  크기의 요구행렬(request matrix)을  $N/4$ 개의  $4 \times 4$  행렬로 분할하여 구성함으로써 신경회로망의 크기를 줄일 수 있다. 본 논문에서 제안한 출력경합회로는 SUN4 SPARC station 상에서 C언어로 구현하며, Bernoulli process에 의해 발생하는 교환 요구 셀들에 대한 실험을 통해 그 유효성을 입증한다.

## II. 교환망에서의 충돌

교환기는  $N$ 개의 입력을 받아 다른  $N$ 개의 원하는 출력으로의 순열(permutation) 연결이 가능한 장치로 규정지을 수 있다. 이때 임의의 순열로 연결이 가능한 교환망을 비충돌형 교환망(nonblocking switching network)이라 하고, 그렇지 못한 교환망을 충돌형 교환망(blocking switching network)이라 한다. 여기에서 충돌(blocking)이란 하나의 교환소자(switching element)에서 2개 이상의 셀들이 동일한 경로로의 교환을 요구하는 경우이다. 비충돌형 교환망의 대표적인 예는 크로스바 스위치(crossbar switch)를 들 수 있으며, 충돌형 교환망은 반얀망(Banyan network), 오메가망(omega network), 베이스라인망(baseline network) 등을 예로 들 수 있다.<sup>19)</sup>

ATM교환기는 일반적으로 다양한 서비스 처리 능력을 가지며 수율이 높아야 한다. 따라서 크로스바 스위치를 교환구조물(switching fabric)로 사용할 경우에는 교환 설정이 간단하고, 모든 출력 순열이 가능한 관계로 높은 수율을 얻을 수 있으나, 입력수  $N$ 에 따라  $N^2$ 개의 교환소자가 필요하므로 입력 수가 커짐에 따라 교환소자의 수가 기하급수적으로 증가하는 단점을 가지고 있다. 따라서 자기루팅(self-routing)의 특성을 가지고 있어 교환 설정이 어렵지 않으며, 교환소자의 수가 비교적 적고( $N \log_2 N$ ) 규칙적인 구조를 가지고 있어 VLSI화하기에 좋은 다단 상호결합망(multistage interconnection network)을 많이 사용하고 있다.

ATM교환기의 한 입력에 여러개의 교환 요구 셀이

집중되는 것을 입력충돌(input blocking)이라 한다. 입력충돌을 피하려면 하나의 셀이 교환되는 동안 다른 셀들은 기다리고 있어야 한다. 따라서 입력 버퍼가 필요하게 되며, 입력버퍼를 설치하여 교환을 행하는 방식을 입력큐잉이라 한다. 입력큐잉 방식은 버퍼가 교환기의 입출력 수 N에 무관한 저속으로 동작하여도 되지만, 버퍼가 FIFO로 동작하는 경우 그 선단 셀만이 출력측에 전달될 수 있으므로 이에 따른 HOL충돌이 문제점으로 대두되어 최대 수용이 매우 떨어지게 된다.<sup>[4]</sup> 이를 해결하기 위한 방법으로는 각 입력에 요구 출력 주소에 따라 버퍼를 따로 두는 방식<sup>[4]</sup> 과 FIFO형이 아닌 FIRO형 입력버퍼를 사용한 교환 방식이 제안되었다.<sup>[1, 5, 14, 16]</sup> 이들 방식을 사용하면 조정논리부(arbitration logic)가 대기 셀들의 교환 순서를 조정해주어야 한다. 그림 1은 일반적인 입력 버퍼형 ATM교환기의 구조이다.

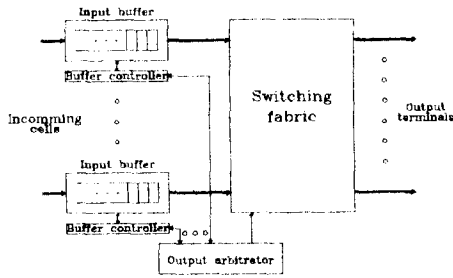


그림 1. 입력 버퍼형 ATM교환기의 일반적인 구조  
Fig. 1. The general architecture of input buffering ATM switch.

각 입력 단자에서 하나의 셀이 각각 교환된다 하더라도 출력지의 주소가 같으면 출력단에서 출력충돌(output blocking)이 발생하게 된다. 출력 충돌을 피하기 위해서는 출력단에 버퍼를 설치하는 출력큐잉 방식<sup>[8]</sup> 을 쓰거나, 입력 버퍼의 셀들 중 조정논리부에서 출력충돌이 일어나지 않는 셀들을 조합하여 선택하는 방식<sup>[1, 4, 5, 14, 16]</sup> 을 사용한다. 출력큐잉 방식을 사용하면 출력 버퍼의 동작속도가 N배 이상의 고속이 되어야하는 단점이 있다.

교환구조물이 충돌형인 경우에는 임의의 입출력간에 오로지 하나의 경로밖에 존재하지 않으므로, 각 교환 셀들이 입력충돌 이나 출력충돌이 일어나지 않는다 하더라도 중간 경로에서 동일한 경로를 요구하는 경우가 발생한다. 이것을 내부충돌(internal blocking)이라 한다. 내부충돌을 피하기 위해서는 우선 교환구조물을 비충돌형인 것으로 사용하는 것을

생각할 수 있다. 크로스바 스위치를 사용하거나<sup>[2, 14, 16]</sup>, 충돌형 다단 상호결합망에 분류망(sorting network)인 배처망(Batcher network)을 부가한 배처반얀망(Batcher Banyan network)<sup>[3, 20]</sup> 을 사용하거나, 공통버퍼를 사용하는 방식<sup>[3, 17, 18]</sup> 등이 연구되고 있다. 크로스바 스위치를 사용하게 되면 입력 수 N에 대해 교환소자의 수가 N<sup>2</sup>개가 되는 단점을 가지며, 배처반얀망을 사용하면 배처망의 규모가 반얀망의 규모보다도 크기때문에(배처망의 교환소자는 크기비교기이다. 이러한 사실을 차치한다 하더라도 배처망의 크기는 반얀망의 크기의 2배 규모가 된다.) 하드웨어 오버헤드가 많아지는 단점을 갖는다. 또한 공통버퍼를 사용하면 버퍼의 동작속도가 N배의 고속 동작이 되어야 한다. 이러한 단점들을 제거하기 위하여 입력버퍼방식에서 조정논리부가 내부충돌이 발생하지 않는 셀들을 조합하여 교환하도록 하는 방식이 제안되었다.<sup>[1, 2]</sup>

본 논문에서는 교환구조물로서 충돌형 다단 상호결합망인 반얀망을 사용한다. 입력충돌을 피하기 위해 입력 버퍼를 채택하며, 출력충돌 및 HOL충돌을 피하기 위해 입력 버퍼에 윈도우를 설정하여 윈도우내에 속해있는 ATM셀들에 대한 교환 순서 조정을 조정논리부에서 행하도록 한다. 또한 내부충돌을 피하기 위해 조정논리부가 교환될 셀을 선택할 때 출력충돌 및 내부충돌이 일어나지 않는 셀들만을 선택하도록 한다.

반얀망은 교환구조물로서 가장 많이 사용되고 있는 다단 상호결합망이다. N×N 반얀망은 2×2 교환소자를 일정한 규칙에 따라 n(=log<sub>2</sub>N)단으로 구성된 구조를 갖는다. 반얀망의 입력과 첫번째 단간의 연결은 완전셔플(perfect shuffle)이며, 이후의 각 단간의 연결은 반얀교환(Banyan exchange)이다. 반얀망에서의 k번째 단의 교환소자는 입력된 셀의 출력주소중에서 상위 k번째 비트의 값에 따라 상(0)하(1)로 교환하여 다음 단으로 연결하여 줌으로써 셀의 출력주소에 의한 자기루팅이 가능하게 된다. 그림 2는 8×8 반얀망이다.

교환을 요구하는 ATM셀의 입력 주소를 I=i<sub>n</sub>i<sub>n-1</sub>...i<sub>1</sub>라 하고 출력 주소를 D=d<sub>n</sub>d<sub>n-1</sub>...d<sub>2</sub>d<sub>1</sub>라 할 때 이 셀을 (I, D)로 표시하면, 각 단의 교환소자의 동작과 연결 상태는 셀의 현재 위치를 나타내는 L의 비트연산으로 다음과 같이 나타낼 수 있다. 먼저 입력단과 첫번째단 사이에서는 완전 셔플이 행해진다.

$$S(a_n a_{n-1} \dots a_2 a_1) = (a_{n-1} a_{n-2} \dots a_2 a_n a_1) \quad (1)$$

따라서 N이 8이고 (001,101)셀이 전송되는 경우에 이 셀은 S(001)=(010)으로 되어 첫번째 단의 (010)번지로 입력된다.

k번째 단의 교환소자에서는 치환(replacement)이 행해진다.

$$R_k(a_n a_{n-1} \dots a_2 a_1) \setminus (d_n d_{n-1} \dots d_2 d_1) = (a_n a_{n-1} \dots a_2 d_{n-k+1}) \quad (2)$$

따라서 (001,101)셀은 첫번째 단에서  $R_k(010)=(011)$ 이 되어 교환소자의 아랫단자로 연결된다.

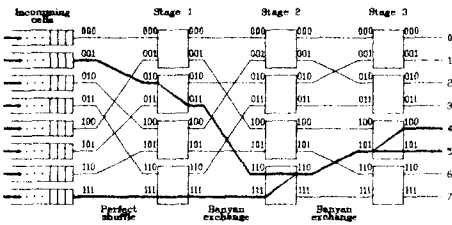


그림 2. 8×8 반양망

Fig. 2. An 8×8 Banyan network.

k번째 단과 k+1번째 단 사이에는 반양교환이 행해진다.

$$B_k(a_n a_{n-1} \dots a_2 a_1) = (a_n a_{n-1} \dots a_{n-k+2} a_1 a_{n-k} \dots a_2 a_{n-k+1}) \quad (3)$$

따라서 (001,101)셀은 첫번째 단과 두번째단 사이에서 B1(011)=(110)이 되어 두번째단의 (110)번지로 입력된다.

이상의 비트연산을 통하여 (001,101)셀은 L이 (001), (010), (011), (110), (110), (101)의 열로 변형되어 원하는 출력으로 루팅된다.

반양망에서는 서로다른 두개의 셀의 입출력 주소가 각각 다음과 같을 때 충돌이 발생한다. 입력주소가 서로 같을 때는 입력충돌이 발생한다. 예를 들어 (001,101)과 (001,010)은 동시에 교환될 수 없다. 출력주소가 서로 같을 때는 출력충돌이 발생한다. 예를 들어 (001,101)과 (110,101)은 출력에서 같은 경로를 요구하게 되므로 동시에 교환될 수 없다. 입력 주소의 k+1번째 비트에서 최하위 비트까지 같고, 출력 주소의 최상위 비트에서 k번째 비트까지 같은 셀은 k번째단에서 내부충돌이 발생한다. 예를 들어 (001,101)과 (111,100)은 두번째 단에서 내부충돌이 발생하며 그림 2의 굵은 선으로 표시하였다.

본 논문에서는 입력충돌, 출력충돌, 내부충돌 등을

모두 배제하는 셀들을 선택한다.

### Ⅲ. 교환 요구행렬의 분할

각 입력 버퍼의 윈도우 내에 있는 셀들은 교환될 집합을 선택하기 위해 교환요구행렬로 구성한다. 교환요구행렬의 각 행은 셀들의 입력 주소를 나타내고, 각 열은 셀들의 출력 주소를 나타내며, 윈도우 내에 (i,j)셀이 있으면 행렬의 i번째 행 j번째열에 1을 표시하고 대응 셀이 없는 위치에는 0을 표시한다. 그림 3은 입출력 수가 각각 8인 교환기에 대한 요구행렬의 한 예이다.

반양망과 같은 충돌망을 교환구조물로 사용하는 경우에는 교환기의 조정논리부에서 충돌이 발생하지 않는 (I,D) 셀의 조합을 선택하여야 한다. 즉 각 단에서 모든 (I,D) 셀은 서로 다른 교환 소자 출력을 통하여 출력되어야 한다.

		출력 주소							
		0	1	2	3	4	5	6	7
입 력 주 소	0	0	1	0	1	0	0	0	1
	1	0	0	0	1	0	0	1	0
	2	0	0	0	0	1	0	0	0
	3	0	0	1	0	0	0	1	0
	4	0	1	0	0	1	0	0	0
	5	1	0	0	1	0	0	0	1
	6	0	0	1	0	0	1	0	0
	7	1	0	0	0	0	0	1	0

그림 3. 8×8 요구행렬

Fig. 3. An 8×8 request matrix.

그림 4는 8×8 요구행렬에 있어서 해당 입출력에 대한 교환이 요구될 때 각 단에서의 스위치의 출력 주소를 나타낸 것이다. 여기에서 같은 번호로 표시된 (I,D) 셀은 동일한 경로를 요구하므로 동시에 교환될 수 없는 경우를 나타내고 있다. 따라서 같은 번호를 갖는 (I,D) 셀이 선택되어 교환되면 해당 단에서 충돌이 발생하게 된다. 열은 2진 순서를 완전 셔플한 순서로 되어 있으며, 행은 최하위 비트를 무시하고 완전 셔플한 순서이다. 8×8 요구행렬의 경우 그림 5(a)와 같이 4개의 4×4 부분 행렬 영역으로 분할하면 부분행렬 ①과 ④는 서로 충돌이 없이 독립적인 교환이 가능함을 알 수 있다. 만약 영역 ①에서 셀들이 선택되면 그에 따라 영역 ②와 ③에서 선택될 수 있는 셀들이 제한되며, 또한 영역 ②와 ③도 영역 ④에

중속적이다. 따라서 ①, ④(또는 ②, ③)의 각 부분 행렬에서 행렬 요소 (i, j), (i+2, j+2), (i+2, j), (i, j+2) 중 하나씩만 선택되면 내부충돌이 발생하지 않는다.

		출력 주소 0 4 1 5 2 6 3 7							
입 력 5 주 2 3 6 7	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1
	4	4	4	4	4	4	4	4	4
	5	5	5	5	5	5	5	5	5
	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3
	6	6	6	6	6	6	6	6	6
	7	7	7	7	7	7	7	7	7

(a) 입력단

		출력 주소 0 4 1 5 2 6 3 7							
입 력 5 주 2 3 6 7	0	0	1	0	1	0	1	0	1
	1	2	3	2	3	2	3	2	3
	4	0	1	0	1	0	1	0	1
	5	2	3	2	3	2	3	2	3
	2	4	5	4	5	4	5	4	5
	3	6	7	6	7	6	7	6	7
	6	4	5	4	5	4	5	4	5
	7	6	7	6	7	6	7	6	7

(b) 제1단

		출력 주소 0 4 1 5 2 6 3 7							
입 력 5 주 2 3 6 7	0	0	4	0	4	1	5	1	5
	1	2	6	2	6	3	7	3	7
	4	0	4	0	4	1	5	1	5
	5	2	6	2	6	3	7	3	7
	2	0	4	0	4	1	5	1	5
	3	2	6	2	6	3	7	3	7
	6	0	4	0	4	1	5	1	5
	7	2	6	2	6	3	7	3	7

(c) 제2단

		출력 주소 0 4 1 5 2 6 3 7							
입 력 5 주 2 3 6 7	0	0	4	1	5	2	6	3	7
	1	0	4	1	5	2	6	3	7
	4	0	4	1	5	2	6	3	7
	5	0	4	1	5	2	6	3	7
	2	0	4	1	5	2	6	3	7
	3	0	4	1	5	2	6	3	7
	6	0	4	1	5	2	6	3	7
	7	0	4	1	5	2	6	3	7

(d) 출력단

그림 4. 각 (I, D)에 대해 각 단에서 요구되는 스위치의 주소

Fig. 4. The addresses of switches for (I, D) cells in each stage.

		출력 주소 0 4 1 5 2 6 3 7							
입 력 5 주 2 3 6 7	0								
	1	①							
	4		②						
	5								
	2								
	3		③						
	6			④					
	7								

(a)

		출력 주소 0 4 1 5 2 6 3 7							
입 력 5 주 2 3 6 7	0								
	1	I							II
	4								
	5								
	2								
	3		II						I
	6								
	7								

(b)

그림 5. (a) 8×8 요구행렬을 4개의 부분행렬 영역으로 분할한 행렬 (b) 4개의 부분행렬을 2개의 군으로 표시한 행렬

Fig. 5. (a) The request matrix divided into 4-part submatrices, and (b) the matrix represented by 2 groups.

조정논리부에서는 하나의 요구행렬에 대하여 첫 시간슬롯(time slot) 동안에는 ①, ④부분행렬에 대한

교환을 행하고, 다음 시간슬롯 동안에는 ②, ③부분행렬에 대한 교환을 행한다. 이때 그림 5(b)에 나타낸 바와 같이 ①, ④부분행렬을 제 I군, ②, ③부분행렬을 제 II군이라 부르기로 한다. 8×8 요구행렬에 있어 각 군에서 입출력 교환이 이루어지는 집합은 다음과 같다.

$$\begin{aligned} \text{제1군} : \{0, 1, 4, 5\} &\rightarrow \{0, 1, 4, 5\} \\ &\quad \{2, 3, 6, 7\} \rightarrow \{2, 3, 6, 7\} \\ \text{제2군} : \{0, 1, 4, 5\} &\rightarrow \{2, 3, 6, 7\} \\ &\quad \{2, 3, 6, 7\} \rightarrow \{0, 1, 4, 5\} \end{aligned} \quad (4)$$

여기에서  $\rightarrow$  기호는 앞의 집합에 있는 입력 주소로부터 뒤의 집합에 있는 출력 주소로의 교환이 일어난을 의미한다. 이때  $ID_i$ 라는 집합을 다음과 같이 정의하면,

$$ID_i = \left\{ 2i, 2i+1, \frac{N}{2} + 2i, \frac{N}{2} + 2i+1 \right\}, i = 0, 1, 2, \dots, \frac{N}{4} - 1 \quad (5)$$

식 (4)는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} \text{제1군} : ID_0 &\rightarrow ID_0 \\ &\quad ID_1 \rightarrow ID_1 \\ \text{제2군} : ID_0 &\rightarrow ID_1 \\ &\quad ID_1 \rightarrow ID_0 \end{aligned} \quad (6)$$

이때 각 군에 대한 교환은 최대 8개의 셀까지 가능하다.

		출력 주소 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1																		
입 력 5 주 2 3 6 7	00	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	01	0	8	0	8	0	8	1	9	1	9	1	9	1	9	1	9	1	9	1
	02	2	A	2	A	2	A	3	B	3	B	3	B	3	B	3	B	3	B	3
	06	0	8	0	8	0	8	0	8	1	9	1	9	1	9	1	9	1	9	1
	09	2	A	2	A	2	A	3	B	3	B	3	B	3	B	3	B	3	B	3
	02	4	C	4	C	4	C	5	D	5	D	5	D	5	D	5	D	5	D	5
	03	6	E	6	E	6	E	7	F	7	F	7	F	7	F	7	F	7	F	7
	10	4	C	4	C	4	C	5	D	5	D	5	D	5	D	5	D	5	D	5
	11	6	E	6	E	6	E	7	F	7	F	7	F	7	F	7	F	7	F	7
	04	0	8	0	8	0	8	0	8	1	9	1	9	1	9	1	9	1	9	1
	05	2	A	2	A	2	A	3	B	3	B	3	B	3	B	3	B	3	B	3
	12	0	8	0	8	0	8	0	8	1	9	1	9	1	9	1	9	1	9	1
	13	2	A	2	A	2	A	3	B	3	B	3	B	3	B	3	B	3	B	3
	06	4	C	4	C	4	C	5	D	5	D	5	D	5	D	5	D	5	D	5
	07	6	E	6	E	6	E	7	F	7	F	7	F	7	F	7	F	7	F	7
	14	4	C	4	C	4	C	5	D	5	D	5	D	5	D	5	D	5	D	5
15	6	E	6	E	6	E	7	F	7	F	7	F	7	F	7	F	7	F	7	

(a)

(b)

그림 6. (a) 16×16 요구행렬에서 각 (I, D)에 대해 제2단에서 요구되는 스위치의 주소 (b) 이에 대한 군으로의 분할

Fig. 6. (a) The addresses of switches for (I, D) cells of 16×16 request matrix in second stage, and (b) its 4-group decomposition.

그림 6(a)는 16×16 요구행렬에 있어서 각 (I,D) 셀이 제2단의 스위치에서 요구하는 출력 주소를 나타낸 것이다. 이 경우도 그림 6(b)와 같이 4개의 군으로 나누어 4개의 시간슬롯 동안에 순회하면서 요구행렬에 대한 교환을 행한다.

일반적으로 입출력 수가 각각 N인 경우에 제k군은 다음과 같이 ID집합간의 교환을 행한다.

$$\text{제 } k\text{군} : ID_i \rightarrow ID_{(i+k-1) \bmod (N/4)}, i=0,1,2,\dots, \frac{N}{4}-1 \quad (7)$$

입출력의 수가 N인 경우 N/4개의 부분행렬이 각각 하나의 군에 속하게 되며, 각 부분행렬은 서로 독립적으로 고려될 수 있으므로 한 시간슬롯 동안에는 최대 N개의 (I,D)셀이 교환될 수 있다.

IV. 신경회로망 모델 및 조정논리부의 동작

그림 7은 본 논문에서 사용한 신경회로망의 뉴런 모델이다. 증폭기는 입력신호  $u_i$ 의 값에 따라  $g(u_i)$ 의 값을 내며, 그 응답 파형은  $u_i$ 가 0보다 작을 때 -1 값을 내고,  $u_i$ 가 0보다 클 때는 +1 값을 낸다.

$$g(u) = \begin{cases} +1 & u > 0 \\ -1 & u \leq 0 \end{cases}, -1 \leq u \leq +1 \quad (8)$$

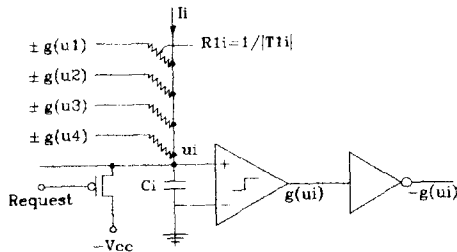


그림 7. 뉴런의 전기적 모델

Fig. 7. The electrical model of a neuron.

뉴런 i와 뉴런 j 사이의 연결 저항 값을  $R_{ij}$ 라 하면, 연결 강도(connection strength)의 절대치  $|T_{ij}|$ 는  $1/R_{ij}$ 가 된다. 이때  $T_{ij}$ 의 값은 흥분성(excitatory) 연결이면 양, 억제성(inhibitory) 연결이면 음이 된다. 본 논문에서의 각 뉴런간의 연결은 모두 억제성 연결이므로  $T_{ij}$ 는  $-1/R_{ij}$ 이다. 이것은  $g(u_j)$ 의 출력을 반전시킨  $-g(u_j)$ 를 공급해 줌으로써 실현할 수 있다. 또한 각 뉴런의 외부로부터의 입력(external input), 내부 문턱치(internal threshold), 내부 귀환 입력(internal feedback input) 등을 통틀어  $I_i$

로 한다.

뉴런 i에서의 전류 방정식을 세우면 식 (9)와 같다.

$$C_i \frac{du_i}{dt} = \sum_{j=1}^N \frac{1}{R_{ij}} (-g(u_j) - u_i) + I_i \quad (9)$$

$T_{ij} = -1/R_{ij}, \lambda_i = \sum_{j=1}^N |T_{ij}|, \lambda_i / C_i = \lambda^*, T_{ij} / C_i = T_{ij}^*, I_i / C_i = I_i^*$ 라 하면, 식 (9)는 다음과 같이 표현할 수 있다.

$$\frac{du_i}{dt} = -\lambda^* u_i + \sum_{j=1}^N T_{ij}^* g(u_j) + I_i^* \quad (10)$$

$u_i$ 의 증분  $\Delta u_i (=u_i(t+\Delta t) - u_i(t))$ 는 식 (11)과 같이 표현된다.

$$\Delta u_i = (-\lambda^* u_i + \sum_{j=1}^N T_{ij}^* g(u_j) + I_i^*) \Delta t \quad (11)$$

3절에서 살펴보았듯이, 요구행렬을 부분행렬로 분할하면, 각 요구행렬의 임의의 행렬요소 (i,j)는 같은 행에 있는 행렬요소와 같은 열에 있는 행렬요소 및  $(i, (j+2) \bmod 4), ((i+2) \bmod 4, j), ((i+2) \bmod 4, (j+2) \bmod 4)$ 의 행렬요소와 서로 충돌이 일어난다. 따라서 신경회로망 모델에서는 이들 뉴런과 (i,j) 뉴런은 상호 억제가 일어나도록 한다. 또한 이들 뉴런을 제외한 다른 뉴런과는 상호 영향을 주지 않도록 한다.

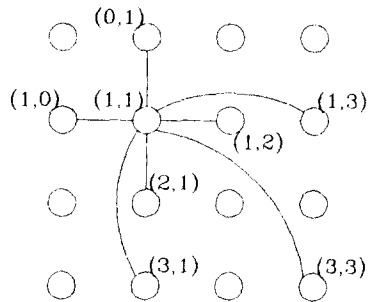


그림 8. (1,1)의 행렬 위치에 있는 뉴런의 억제성 연결

Fig. 8. The inhibitory connections of a neuron at the matrix position (1,1).

즉, 뉴런 i와 j가 상호간에 충돌이 있으면  $T_{ij}^* < 0$ , i와 j가 충돌이 일어나지 않으면  $T_{ij}^* = 0$ 이 되도록 한다. 4×4 분할행렬에 대응하는 신경회로망에 있어서 뉴런 (1,1)과 다른 뉴런간의 억제성 연결의 예를 그림 8에 보였다. 실선은 서로 억제성 연결임을 나타내고, 선의 연결이 없는 뉴런간의 연결 강도는 0임을 나타낸다.

4×4 분할행렬의 각 뉴런은 다른 7개의 뉴런과 역제성 연결을 가진다. 따라서 뉴런  $i$ 와 뉴런  $j$ 가 충돌이 있는 경우의  $T_{ij}^* = K$ 라 하면,  $\lambda_i = \sum_j |T_{ij}^*| = 7K$ 이다. 만약, 상호 충돌인 셀이 하나 이상 발생하면,

$$\sum_{j=1}^N T_{ij}^* g(u_j) \leq 5K \quad (12)$$

가 된다. 이때  $\Delta u_i \leq 0$ 이다. 또한 상호 충돌이 일어나는 셀이 발생하지 않으면

$$\sum_{j=1}^N T_{ij}^* g(u_j) = 7K \quad (13)$$

가 되며, 이때  $\Delta u_i > 0$ 이어야 한다. 따라서 식 (12)와 식 (13)의 조건으로부터  $I_i^*$ 의 범위를 구하면

$$7K u_i - 7K < I_i^* < 7K u_i - 5K \quad (14)$$

가 된다. 식 (14)는 그림 9의 빗금친 부분에  $I_i^*$ 가 있어야 함을 보인다. 본 논문에서는  $I_i^*$ 를 식 (15)와 같이 설정하여 실험하였다.

$$I_i^* = 7K u_i + \gamma, -7K < \gamma < -5K \quad (15)$$

본 논문의 조정논리부는 입출력의 수가 각각  $N$ 일 때,  $N/4$ 개의 신경회로망 모듈로 이루어지며, 각 신경회로망 모듈은 16개의 뉴런이  $N \times N$  행렬 구조로 배열되어 있다. 신경회로망 모듈들은 대응되는 분할요구행렬의 각 행렬요소 값을 입력으로 받는다. 이때 행렬요소의 값이 1(-1)이면 대응 뉴런  $i$ 의  $u_i$  값을 1(-1)로 한다.

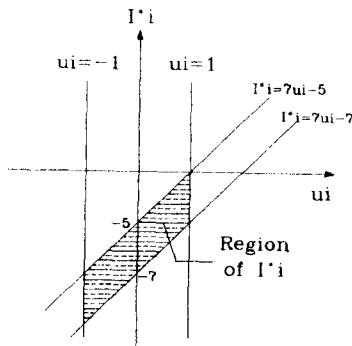


그림 9.  $I_i^*$ 의 범위  
Fig. 9. The region of  $I_i^*$

각 신경회로망 모듈들은 서로 독립적으로 시간발전

을 통해 안정화하며, 안정화된 결과는 조정논리부의 출력으로서, 선택된 (I,D) 셀들을 입력 버퍼에서 출력측으로 교환하는 데 쓰인다. 하나의 시간슬롯 동안에는 요구행렬의 하나의 군에 대한 교환을 행하며, 다음 시간슬롯 동안에 다음 군에 대한 교환을 행한다. 그림 11은 교환 알고리즘을 나타낸다.

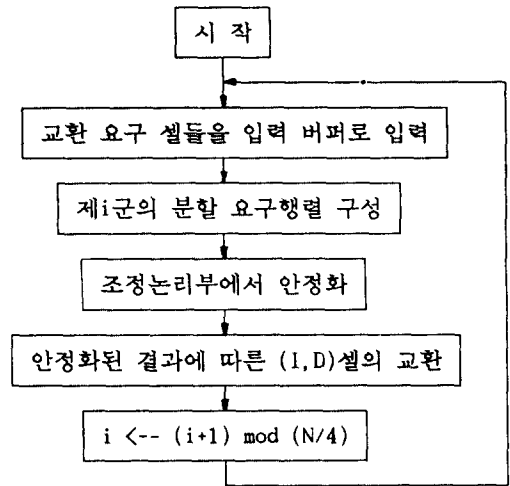


그림 10. 교환 알고리즘  
Fig. 10. The switching algorithm.

본 논문에서의 신경회로망은  $14N (= N/4 \times (16 \times 7/2))$  개의 연결선을 가지며,  $4N (= N/4 \times 4 \times 4)$  개의 뉴런을 갖는다. 이는 [1] 및 [4] 등의 신경회로망의 크기와 비교할 때 매우 적은 수임을 알 수 있다.

### V. 실험 결과 및 검토

본 논문에서 제안한 신경회로망을 이용한 조정논리부는 SUN-4 SPARC station 상에서 C 언어로 구현하였다. 성능 평가를 위해 전송율(transfer rate)을 다음과 같이 정의한다.

$$\text{전송율} = \frac{\text{단위 시간당 교환된 셀의 수}}{\text{단위 시간당 입력단에 들어온 셀의 수}} \times 100$$

시뮬레이션을 위하여 식 (15)에서의  $K$ 와  $\gamma$ 를 각각 1, -6으로 하였다.

윈도우 크기(window size)는 입력 버퍼의 (I,D) 셀을 요구행렬에 포함시키는 범위의 크기로 정의한다. 그림 11은 입출력 수  $N$ 의 크기가 4에서 128까지 일 때, 윈도우 크기의 변화에 따른 신경회로망에 의한 전송율의 변화이다. 시뮬레이션은 각 교환기에 대

해 5000 시간슬롯 동안 행하였다. (I,D)셀은 각 입력 버퍼에 Bernoulli process에 의해 도착하며, 셀 도착 확률  $\alpha$ 는 0.9인 것으로 하였다.

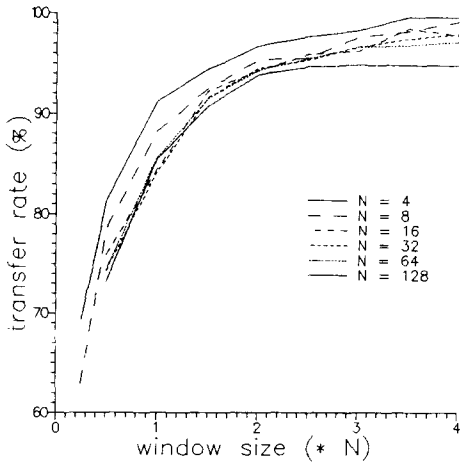


그림 11. 셀 도착 확률이 0.9일 때의 윈도우 크기에 따른 신경회로망에 의한 전송율

Fig. 11. The results of neural network processing of transfer rate versus window size with cell arrival probability of 0.9.

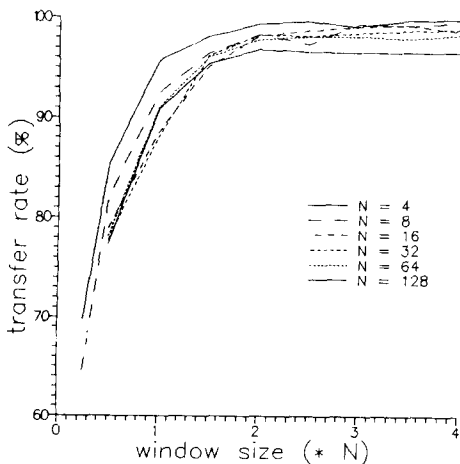


그림 12. 셀 도착 확률이 0.9일 때의 윈도우 크기에 따른 계산에 의한 전송율

Fig. 12. The calculated results of transfer rate versus window size with cell arrival probability of 0.9.

그림 12는 신경회로망에 의하지 않고 계산에 의해 각 시간 슬롯 동안에 전송 가능한 최대 수의 셀을 전송한 경우이다. 그림 13은 N의 크기가 8, 32일 때

신경회로망에 의한 결과와 계산에 의한 결과를 비교한 것이다. 신경회로망에 의한 결과가 약간 낮은 전송율을 나타내는 것은 신경회로망이 안정화될 때 가끔 국소최적해(local minima)를 내기 때문이다. 이는 신경회로망의 모델 및 퍼라미터의 조정을 통해 개선될 수 있을 것으로 생각된다.

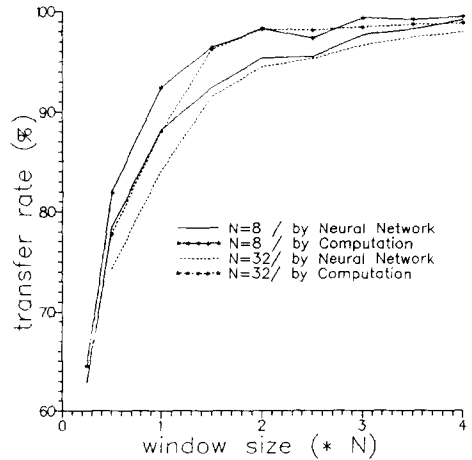


그림 13. 윈도우 크기에 따른 신경회로망에 의한 전송율과 계산에 의한 전송율의 비교

Fig. 13. The comparison between the transfer rates by neural networks and by calculation with respect to the window size.

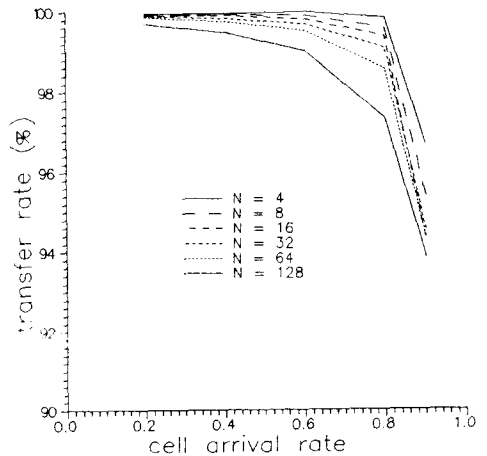


그림 14. 윈도우 크기가 2N일 때의 셀 도착 확률에 따른 신경회로망에 의한 전송율

Fig. 14. The results of neural network processing of transfer rate versus cell arrival probability with window size of 2N.



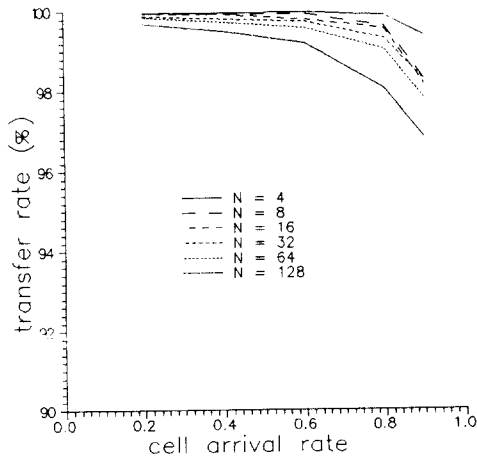


그림 15. 윈도우 크기가 2N일 때의 셀 도착 확률에 따른 계산에 의한 전송률  
 Fig. 15. The calculated results of transfer rate versus cell arrival probability with window size of 2N.

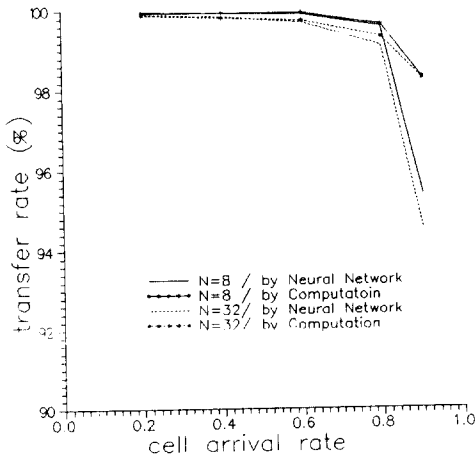


그림 16. 셀 도착 확률에 따른 신경회로망에 의한 전송률과 계산에 의한 전송률의 비교  
 Fig. 16. The comparison between the transfer rates by neural networks and by calculation with respect to cell arrival rate.

그림 14는 셀 도착 확률의 변화에 따른 신경회로망에 의한 확률을 나타낸 것이다. 여기서 윈도우 크기는 2N으로 한다. 그림 15는 계산에 의한 전송율을 나타내며, 그림 16은 신경회로망에 의한 전송율과 계

산에 의한 전송율을 비교한 것이다. 이 결과에서  $\gamma$ 가 0.8 이하인 경우에는 99% 이상의 높은 전송율을 보이고 있음을 알 수 있다.

VI. 결론

본 논문에서는 입력 대기형 ATM 교환기의 FIRO 방식 입력 버퍼를 효율적으로 스케줄링하기 위한 교환 요구행렬 분할 알고리즘을 제안하였다. 교환구조 물로는 총돌망인 반안망을 사용함으로써 비총돌망인 배치-반안망보다 하드웨어의 양을 감소시킬수 있었으며, 반안망의 내부총돌 분석에 근거하여  $N \times N$  교환 요구행렬을  $N/4$ 개의 군으로 분할 처리함으로써 기존의 방식에 필적할만한 수율을 얻을 수 있었다.

조정논리부는 신경회로망으로 설계하였다. 신경회로망은 대량의 병렬성을 얻을 수 있는 장점을 가질수 있으므로 빠르고 효율적인 스케줄링이 가능하다. 또한 전 조정논리부를  $N/4$ 개의  $4 \times 4$  신경회로망 모듈로 나누어 구성하고 동시에 동작할 수 있도록 함으로써 전체 뉴런의 수를 4N개, 전체 연결선의 수를 14N개로 줄이면서도 높은 수율을 얻을 수 있었다.

제안된 조정논리부는 SUN-4 SPARC station 상에서 C언어로 구현하였으며, 128개까지의 입력을 갖는 교환기에 대한 실험 결과를 통해 그 유효성이 입증하였다.

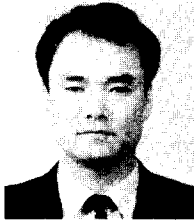
앞으로의 연구 과제는 신경회로망 모델의 개선 및 퍼라미터의 조정을 통하여 국소최적해에 빠지지 않도록 하는 문제와 셀의 교환 우선 순위를 고려할 수 있도록 하는 문제 등이다. 또한 좀더 많은 입출력을 갖는 교환기를 제어할 수 있도록 하는 확장성에 대한 연구가 이루어져야 할 것이다.

參考文獻

[1] T.X.Brown and K.H.Liu, "Neural Network Design of a Banyan Network Controller," *IEEE J. Select. Areas in Commun.*, Vol.8, pp.1428-1438, Oct. 1990.  
 [2] T.X.Brown, "Neural Networks for Switching," *IEEE Commun. Mag.*, pp. 72-81, Nov.1989  
 [3] T.Takeuchi, H.Suzuki, and T. Aramaki, "Switch Architectures and Technologies for Asynchronous Transfer Mode.", *IEICE Trans.*, Vol.E74, No 4,

- pp.752-760, April 1991.
- [4] M.M.Ali and T.Nguyen, "A Neural Network Implementation of An Input Access in A High-Speed Packet Switch," GLOBECOM 89, pp.1192-1196, 1989.
- [5] X.Ji and D.Wang, "Fast Packet Switch Controller Based on Nonlinear Programming Neural Network," IJCNN, Vol.1, pp.270-273, Nov.1992.
- [6] A.Hiramatsu, "ATM Communication Network Control by Neural Network," IJCNN, Vol.1, pp.259-266, June 1989.
- [7] F.A.Tobagi, "Fast Packet Switch Architectures for Broadband Integrated Services Digital Networks," *Proc. of IEEE*, Vol.78, No1, pp.133-167, Jan. 1990.
- [8] K.Murano, K.Murakami, E.Iwabuchi, T.Katsuki, and H.Ogasawara, "Technologies Towards Broadband ISDN," *IEEE Commun. Mag.*, pp.66-70, April 1990.
- [9] J.J.Hopfield and D.W.Tank, "Neural "Computation of Decisions in Optimization Problems," *Biol. Cybern.*, Vol.52, pp.141-152, 1985.
- [10] J.J.Hopfield, "Neurons with Graded Response Have Collective Computational Properties like Those of Two-state neurons," *Proc. Natl. Acad. Sci.*, USA 81, pp.3088-3092, 1984.
- [11] J.J.Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. Natl. Acad. Sci.*, USA 79, pp.2554-2558, 1982
- [12] D.Tank and J.J.Hopfield, "Simple" Neural " Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Trans. CAS*, Vol.33, No.5, pp.533-541, May 1986.
- [13] D.M.Dias and J.R.Jump, "Packet Switching Interconnection Networks for Modular Systems," *IEEE Computer Magazine*, pp.43-53, Dec. 1981.
- [14] T.P.Troudet and S.M.Walter, "Neural Network Architecture for Crossbar Switch Control," *IEEE Trans. CAS* Vol. 38, No.1, Jan. 1991.
- [15] T.Feng, "A Survey of Interconnection Networks," *IEEE Computer Magazine*, pp.12-27, Dec. 1981.
- [16] A.Marrakchi and T.Troudet, "A Neural Net Arbitrator for Large Crossbar Packet-Switches," *IEEE Trans. CAS*, Vol.36, No.7, pp.1039-1041, July 1989.
- [17] 이병기, 강민호, 이종희, '광대역 통신 시스템,' 교학사 1992.
- [18] '비동기식 전달모드,' 한국통신 1992.
- [19] 富田眞治, 末吉敏則, '並列處理マシン' オーム社 1989.
- [20] 이성은, 김근수, 하상진, 신종훈, 박창욱, 김영우, 임인철, "BISDN에서의 신경회로망을 이용한 ATM 스위치 네트워크 제어기의 설계," 전자계산,반도체,재료 및 부품,씨에이디 및 VLSI설계 합동학술발표회 논문집, pp.16-20, June 1992.
- [21] 하상진, 신종훈, 김신, 김미형, 김장호, 김미선, 이주영, 박창욱, 김영우, 임인철, "신경회로망을 이용한 ATM 교환망 출력 경합회로의 설계," 전자계산,반도체 재료 및 부품,씨에이디 및 VLSI설계 합동학술발표회 논문집, pp.239-242, May 1993.

著者紹介



金榮禹(正會員)

1962年 2月 18日生. 1981年 3月  
~ 1985年 2月 한양대학교 전자공  
학과(공학사). 1985年 3月 ~  
1987年 2月 한양대학교 대학원 전  
자공학과(공학석사). 1989年 3月  
~ 현재 한양대학교 대학원 전자  
공학과(박사과정). 주관심 분야는 논리회로 테스트,  
Broadband ISDN 교환제어, CAC(Call  
Admission Control) 및 routing 등임.

林寅七(正會員) 第 30卷 B編 第 2號 參照

현재 한양대학교 전자공학과 교수