論 文

통신 예약 버스 방식을 이용한 IPC 통신망 구성에 관한 연구

準會員 金 祐 鍵* 正會員 朴 永 德** 正會員 金 宣 衡***
正會員 曺 圭 燮**** 正會員 朴 炳 哲*****

The Implementation of the IPC Network using the Reserved Bus Topology

Ho Geon KIM*, Yung Duck PARK**, Sun Hyoung KIM***,
Kyu Seob CHO****, Byung Chul PARK***** *Regular Members*

要 約 근래에 들어 통신기기의 지능화에 대한 필요성과 프로세서 가격의 저렴화가 서로 부합되어 하나의 시스템내에 다수의 프로세서를 실장하는 것이 공통된 경향이다. 본 논문에서는 이와같이 다수의 근접된 프로세서 상호간에 적용 가능한 통신 방식으로 "근접된 프로세서간 통신 방식에 관한 연구"[1]에서 이미 제안한 "통신예약 버스" 방식에 적용할 수 있는 하드웨어및 소프트웨어를 개발, 제시하였으며 이에대한 실험을 통하여 본 방식의 타당성 및 관련 하드웨어와 소프트웨어의 실용성을 검증하였다.

ABSTRACT Nowadays, the needs for intelligence of communication equipments and the cost down of micro processor are showing a tendency to have multi-processor in a single system. In this paper, based on the Reserved Bus Topology which is proposed in "A study on the Communication Method between the adjacent processor", the software and hardware is designed and developed. And the validity of this method and the utility of designed software and hardware functions are also verified through experiments.

I. 서 론

근래에 들어 통신기기의 지능화에 대한필요성과 프로세서 가격의 저렴화가 서로 부합되어 시스템내에 프로세서를 실장하는 것이일반화된 추세이다[2]. 더구나 앞으로의 통신기기는 다기능화 및 유지, 보수 등을 위해 여러개의 프로세서가 하나의 동일시스템 내에 존재하게 될 것으로 예측된다. 따라서 이러한 시스템하에서는 프로세서 상호간에 원할하고 효율적인 메시지 교환이 필수 불가결한 하나의 요소로서 대두되고있으며, 이와같은 환경에 적합한 통신 방식에 대한 연구가 요구되고 있다. 그 한 예로서 앞으로의 교환기는 향후 ISDN 서비스 및 다양한 트래픽 형태

*, **, *****成均館大學校工科大學 電子工學科
Dept. of Electronic Engineering
Sung Kyun Kwan University. Suwon, 170 Korea.
***仁德工業專門大學 電子科
Dept. of Electronic Induk Institute of Design
Seoul, 132 Korea.
****韓國電子通信研究所
Electronics and Telecommunications Research Institute
論文番號 : 88-04 (接受 1987. 11. 24)

에 유연히 대처하기 위해서 독립된 하드웨어와 소프트웨어를 갖는 다수의 프로세서들에 제어기능을 분산시켜야 하며[3-5] 각각의 독립된 제어 프로세서들간에 효율적인 통신을 관장하는 프로세서간 통신(IPC:Inter Processor Communication) 망이 교환기의 성능을 좌우하는 커다란 변수로 작용하게 된다.

본 논문은 이와같은 근접한 프로세서간 통신 방식으로 앞서 제안한 통신 예약 버스[1]의 타당성 및 그 효용성을 검증하기 위해 소규모 실험실용 IPC 통신망을 실현하여 통신 예약 버스의 구성에 관한 제반 검토사항 및 성능을 분석하였다.

서론에 이어 2장에서는 통신 예약 버스의 개요 및 이의 적용형태, 3장에서는 실험실용 IPC 통신망의 구성에 따른 하드웨어 및 소프트웨어구조에 대해 각각 기술하였고 4장에서는 자체 제작한 IPC통신망에 대한 실험 방법 및 결과에 대해 기술한 후 결론을 맺음으로써 논문의 끝맺음을 하였다.

Ⅱ. 통신 예약 버스의 개요

통신 예약 버스의 토폴로지는 그림 1과 같이 기존의 링과 버스구조가 혼합된 이중 구조로 되어 있으며 전송 권리는 링 경로를, 데이터는 버스 경로를 통해 각각 처리된다. 이와 같이 전송권리와 데이터 전송을 이분화시킨 구조는 전송권리 획득에 따른 프레임 overhead를 감소시킬 수 있으므로 관련 소프트웨어 구조가 간편해지며, 또한 전송권리의 처리가 데이터 처리와 별도로 수행되어 관련 하드웨어가 간단해지므로 버스및 링인터페이스 부분도 일반 범용IC의 사용으로 손쉽게 실현할 수 있다.

본 방식에 있어서 데이터의 전송은 그림 2에 도시한 바와 같이 송신을 원하는 프로세서가 전송로의 사용을 위해 송신을 원하는 즉시 전송로 사용을 예약한 후 본 논문에서 제시한 예약 회로에 의해 송신할 권리가 주어지면 HDLC 프레임 구조에 따른 데이터 프레임을 버스 형태인 단일 전송로를 통해 송신하고 수신 프로세서는 데이터의 주소를 확인하여 자신의 주소와 일치하면 데이터를 수신한다. 또한 송신 프로세서는 전송완료와 함께 링 경로를 통해 다음 프로세서로 전송권리를 이양하여 타 프로세서들이 전송을 할 수 있도록 하였다.

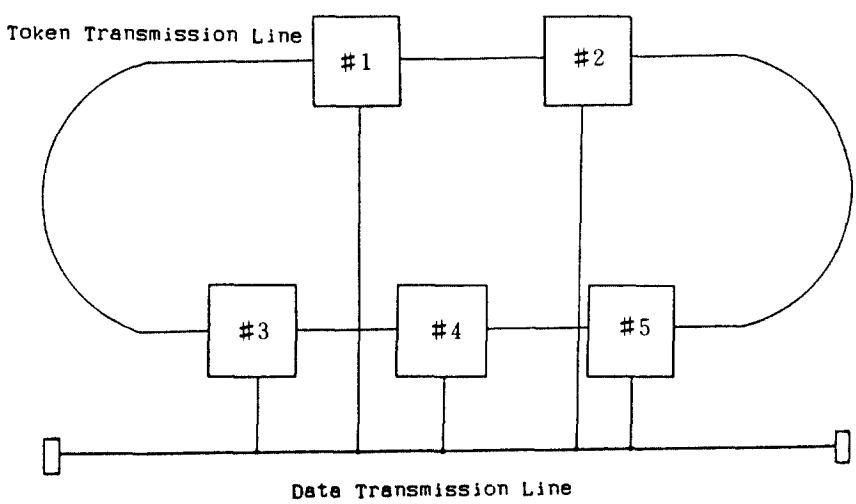


그림 1 통신 예약 버스의 토폴로지
Topology of reserved bus.

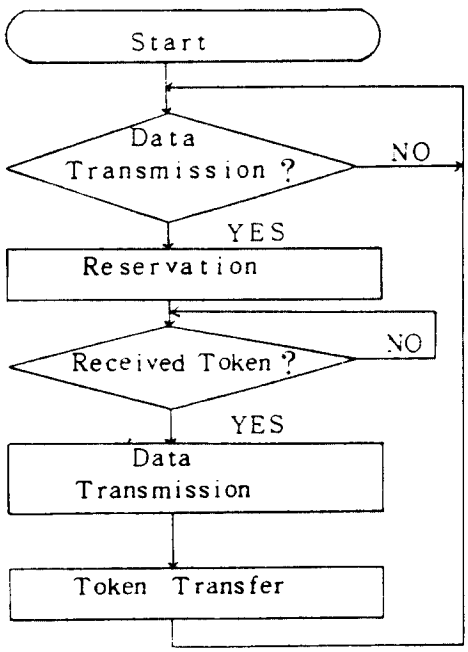


그림 2 통신 예약 버스의 데이터 전송 절차
Date transmission procedure of reserved bus.

전송권리는 링을 통하여 물리적 신호(logic "0"의 펄스, 기존의 토큰 개념과 유사함)로 전송되어지며 예약회로는 이러한 전송권리(이하토큰이라 칭함)가 도착하였을 때 관련된 프로세서가 통신을 예약하였는 지를 확인하게 된다. 프로세서가 통신을 예약하였을 경우 예약회로는 인터럽트를 발생시켜 프로세서에 통신 가능상태를 지시하며 통신이 종료됨과 함께 프로세서에 의해 새로운 토큰이 발생되어 다음 프로세서로 전송된다. 프로세서가 통신을 예약하지 않은 경우 예약회로에 토큰이 도달하게 되면 이 토큰은 예약회로에서 그대로 통과되어 다음 프로세서로 전달된다.

그림 1은 앞의 논문에서 제시하였던 통신 예약 버스의 기본 구조로서 이러한 구조는 트래픽이 모든 프로세서에 고르게 분산되어 있는 경우에 효율적일 것으로 예측된다. 이러한 기본 구조의 또다른 형태로서 그림 3과 같은 구조가 제안될 수 있을 것이다. 즉, 전체적인 IPC 통신망내에서 프로세서는 하나의 상위프로세서와 나머지 프로세서들로서 구성되어 지며 하위 프로세서에서 상위 프로세서로 메시지를 송신할 때에는 지금까지 검토한 바와 같이 링과 버스 1에 의해 데이터가 처리되며 상위 프로세서가 하위 프로세서에 메시지를 전송할 때에는 버스 2로 broadcasting 방식에 의해 데이터가 처리된다. 이와같은 구조는 하위 프로세서간에는 서로의 메시지 교환이 없고 상위 프로세서와 하위 프로세서군간의 메시지 교환만이 있을 경우 즉, 트래픽이 한 곳으로 집중되어 질 때 매우 유용할 것이며 이러한 형태는 바로 교환기 내에서 가입자 모듈군과 이에 대응

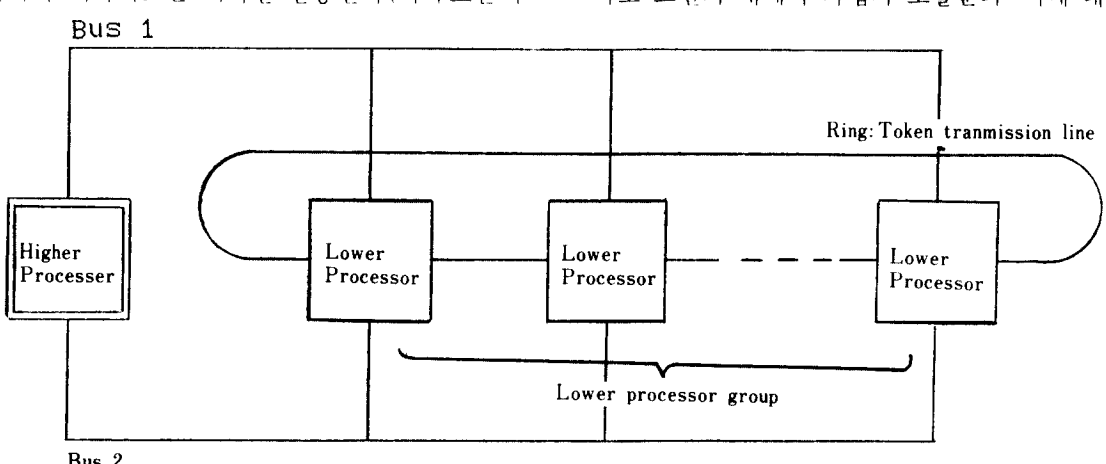


그림 3 트래픽 집중 시 네트워크 토폴로지
Network topology for the traffic concentration.

하는 상위 프로세서간의 통신형태에 대응할 것이다.

본 논문에서는 그림 1과 같은 형태를 기준으로 IPC 통신시스템을 제작한 후 이에 대한 실험을 통하여 본 방식의 타당성을 입증하였다.

Ⅲ. IPC 통신망의 구성

Ⅲ-1. IPC 통신망 구성에 대한 기본 가정

IPC 통신망 구성시 주된 고려 사항은 프로세서당 bit delay의 최소화, 전송 overhead의 최소화, 전송 프로토콜의 간편화에 중점을 두었으며, 통신망 구성에 따른 기본 가정은 다음과 같다.

- 전기한 바와같이 그림 1의 기본구조를 대상으로 하며 외부와의 인터페이스 연구는 일단 제외한다.
- 주 프로세서로 하나의 16비트 마이크로 프로세서를 실장하여 송, 수신 제어를 수행하도록 한다.
- 토큰의 폭이 데이터 전송 bit rate의 1 비트 이상을 초과하지 않도록 한다.
- 데이터 전송 프레임은 HDLC 프레임 구조에 따른다.
- 소프트웨어와 하드웨어를 모듈화하여 시스템 확장에 유연히 대처하게 한다.
- 기본 전송 bit rate는 2 Mbps로 한다.
- IPC controller를 내장한 4개의 프로세서보드를 제작하여 소규모 IPC 통신망을 구성하여 연동 실험을 한다.

Ⅲ-2. IPC controller의 하드웨어구조

IPC controller의 하드웨어 구조는 크게 토큰 및 전송 예약을 관장하는 RRIC (Reserved Ring Interface Controller), 데이터 프레임 전송을 담당하는 BBIC (Broadcasting Bus Interface Controller)와 시스템 전체를 제어하는 주 프로세서 (MP; Main Processor)로 구성되어 있으며, 이에 따른 하드웨어 전체 block diagram은 그림 4와

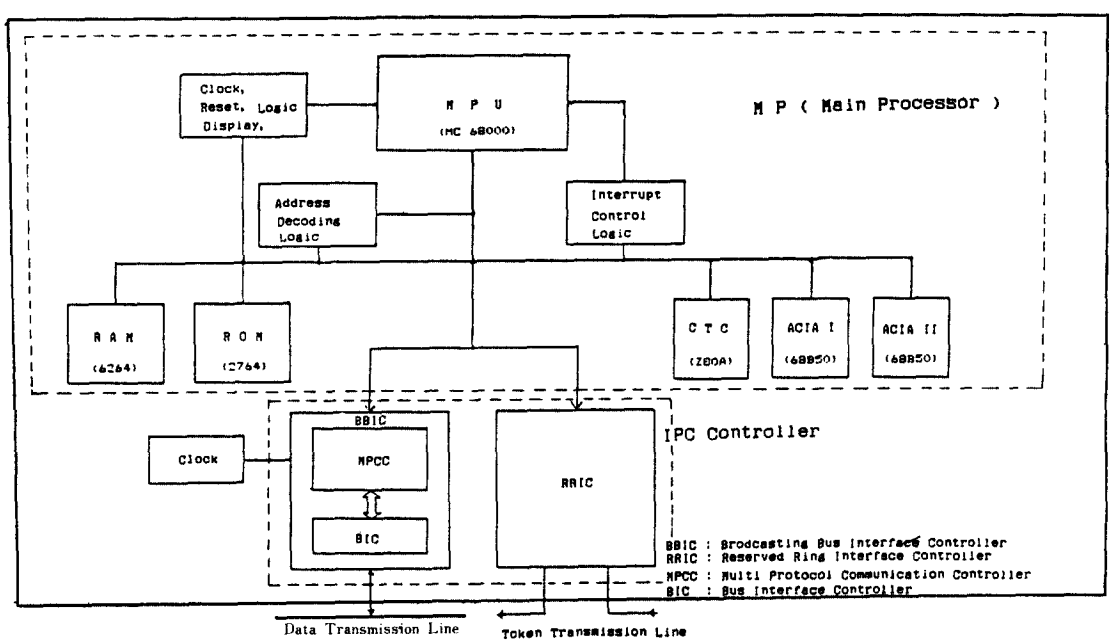


그림 4 하드웨어 전체 블럭도

The hardware block diagram.

같다.

Ⅲ-2-1 RRIC(Reserved Ring Interface Controller)

토큰의 생성, 이양 및 전송 예약 회로를 포함하는 RRIC의 전체 회로도는 그림 5와 같으며 동작 개요는 다음과 같다.

전체 회로는 최초 급전 (power feed) 시에 초기화 되며, 초기화시에 토큰이 전송되는 링을 "Logical HIGH' 상태로 만들어 전 링상에는 토큰이 존재하지 않는다.

회로 초기화 후 링상의 토큰을 감시, 복구하는 기능을 수행하는 감시 프로세서는 토큰의 최초 발생을 명령하게 되며 이에 따라 U8, U9, U10 등에 의해 500ns의 low pulse 폭을 갖는 토큰이 발생되어 링으로 전송한다. 각 프로세서는 자신이 전송할 데이터가 있는 경우 U5를 reset하고, 전송할 데이터가 없는 경우에는 set 상태로 한다. 따라서 U5가 set상태에 있는 때에는 토큰은 U11, U14, U7, U15를 통하여 그대로 다음프로세스로 전송된다. 여기서 U1과 U2는 수신한 토큰을, U7은 송신할 토큰을 500ns의 Low pulse로 재정형시키는 역할을 담당한다. 전송할 데이터가 있는 프로세서는 U5를 reset 시킨 후 토큰이 수신되면 U3, U4에 의해 주 프로세서에 인터럽트가 인가되며 이에의해 프로세서는 인터럽트를 clear시킨 후 전송할 데이터 BBIC를 통해 버스로 broadcasting한다. 데이터 전송을 완료하면 프로세서는 새로운 토큰을 발생하여 링 경로를 통해 다음 프로세서로 전달한다. 이와같은 동작을 수행하는 RRIC 전체 회로는 기존의 TTL 소자를 이용하여 구성하였다.

Ⅲ-2-2 BBIC(Broadcasting Bus Interface Controller)와 MP(Main Processor)

BBIC는 전송을 원하는 프로세서가 버스형태인 다입 전송로를 통해 데이터를 broadcasting하고 수신 프로세서가 이를 받아들일 수 있도록 버스 인터페이스의 송신모드, 수신모드를 결정한다. 즉, 데이터 전송시 버스 인터페이스는 송

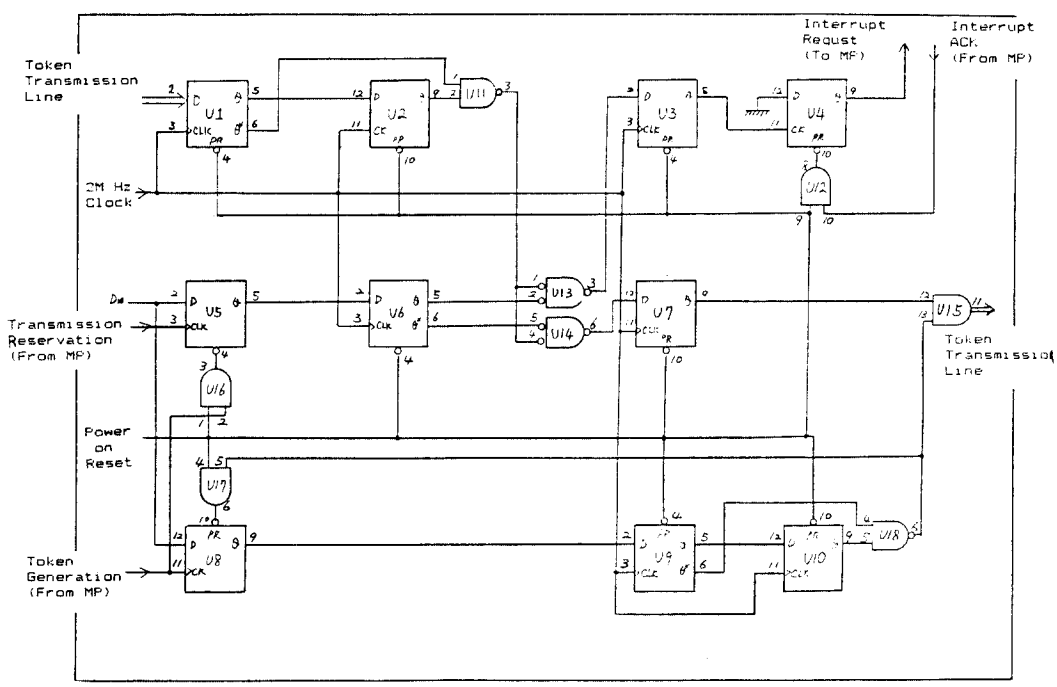


그림 5 RRIC의 전체 회로도
Overall logic of RRIC.

신모드로, 그외에는 항상 수신모드로 동작하며 프로세서의 제어를 받는 양방향 tri-state buffer로 간단히 구성하였다.

MP 모듈은 범용 16 비트 마이크로 프로세서인 Motorola의 68000과 기본 logic 구성에 따른 회로외에 프로그램 timer 관리를 CTC(Counter Timer Controller), 프로그램 down load 및 monitoring을 위한 시스템 개발용 하드웨어를 부수적으로 포함하고 있다. 또한, HDLC 프레임에 따른 데이타 송수신을 위하여 Rockwell 사의 68561[6] 을 사용하였다.

Ⅲ-3 IPC controller의 소프트웨어 구조.

통신 예약 버스의 소프트웨어는 크게 각 제어 기능을 분담할 일반 프로세서부와 일반 프로세서들로 구성된 IPC 통신망 전체를 관리하는 감시 프로세서부로 나누어 구성하였으며, 각 프로세서부의 소프트웨어 구성표는 표 1과 같다. 표 1

표 1 프로세서 별 소프트웨어 구성표
Software table of each processor

	Base level	Interrupt level
IPC network management processor	• Token generation routine • Token reservation routine • Interrupt condition handling routine • Token regeneration according to token loss	• Tx interrupt routine • Rx interrupt routine • TLT interrupt routine • THT interrupt routine
General Processor	• Token reservation routine • Interrupt condition handling routine	• Tx interrupt routine • Rx interrupt routine • THT interrupt routine

TLT: Token Loss recovery Timer
THT: Token Holding Timer

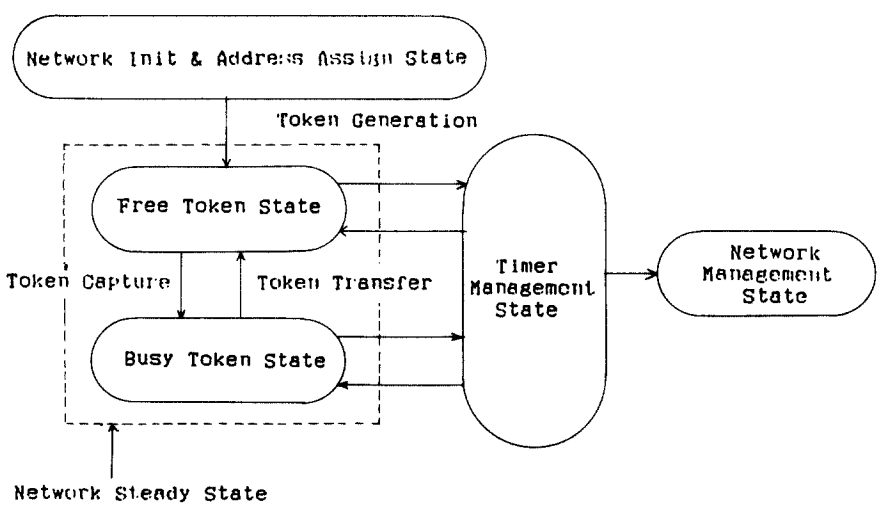


그림 6 IPC controller의 전체 상태 천이도
Overall state transition of IPC controller.

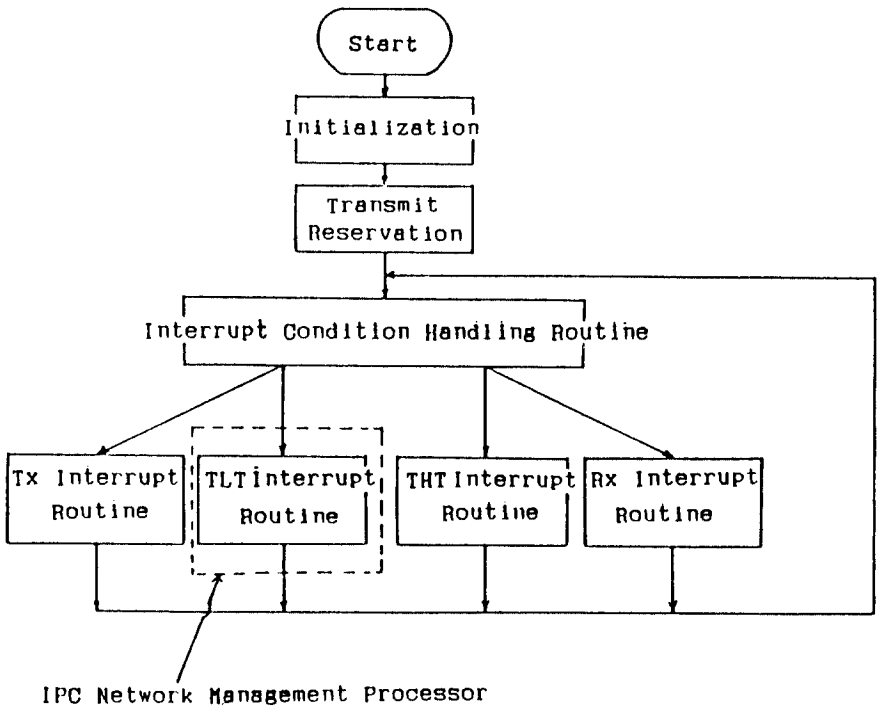


그림 7 IPC controller의 전체 순서도
Overall flowchart of IPC controller.

에 도시되어 있는 일반적인 기능외에 실제 시스템의 성능과 수학적 모델링에 의해 구해진 결과를 비교 분석하기 위한 부가적인 소프트웨어를 각 프로세서에 첨가하여 통신예약버스의 타당성 및 효용성을 검증할 수 있도록 하였다.

각 프로세서부의 소프트웨어는 베이스 레벨에서 초기화 및 job 분배 routine을, 인터럽트 레벨에서는 타 프로세서와의 통신을 위한 데이터 프레임 송수신 routine과 network 관리 routine을 각각 두었다.

그림 6은 각 프로세서부의 상태 천이도로써 네트워크 초기화 및 각 프로세서의 어드레스 할당상태, 어떤 프로세서도 토큰을 예약하지 않아 토큰이 링을 계속 일주하는 free token state, 프로세서가 전송예약을 한 후 토큰을 획득하여 패킷을 전송하는 busy token state, 각 상태에 따른 timer관리상태, 전체망을 관장하는 network management state 로 구성되어 있으며, 그림 7과 같은 전체 flowchart에 따라 각 상태를 천이하면서 주어진 동작을 수행한다.

Ⅲ-3-1 초기화 및 job 분배 routine

초기화 routine에서는 기본 파라미터의 초기화, 메모리 clear, I/O (input/output) 칩의 초기화, 토큰 초기 발생등의 작업을 수행하며, job 분배 routine에서는 interrupt조건에 따른 job 분배 및 관리, 토큰 예약 및 해체, 토큰 패킷의 메모리 할당 작업등의 일을 수행 하며, 이에 따른 flowchart는 그림 8과 같다.

Ⅲ-3-2 데이터 송수신 routine

데이터 송수신 routine에서는 interrupt 요구에 의한 데이터 송수신, 프레임 error검출, 버스 인터페이스 관리등의 일을 수행한다. 데이터 송신은 68000의 auto interrupt level에서, 데이터 수신은 여러 가지 수신 데이터 프레임의 조건 (end of frame detection, secondary mode에따

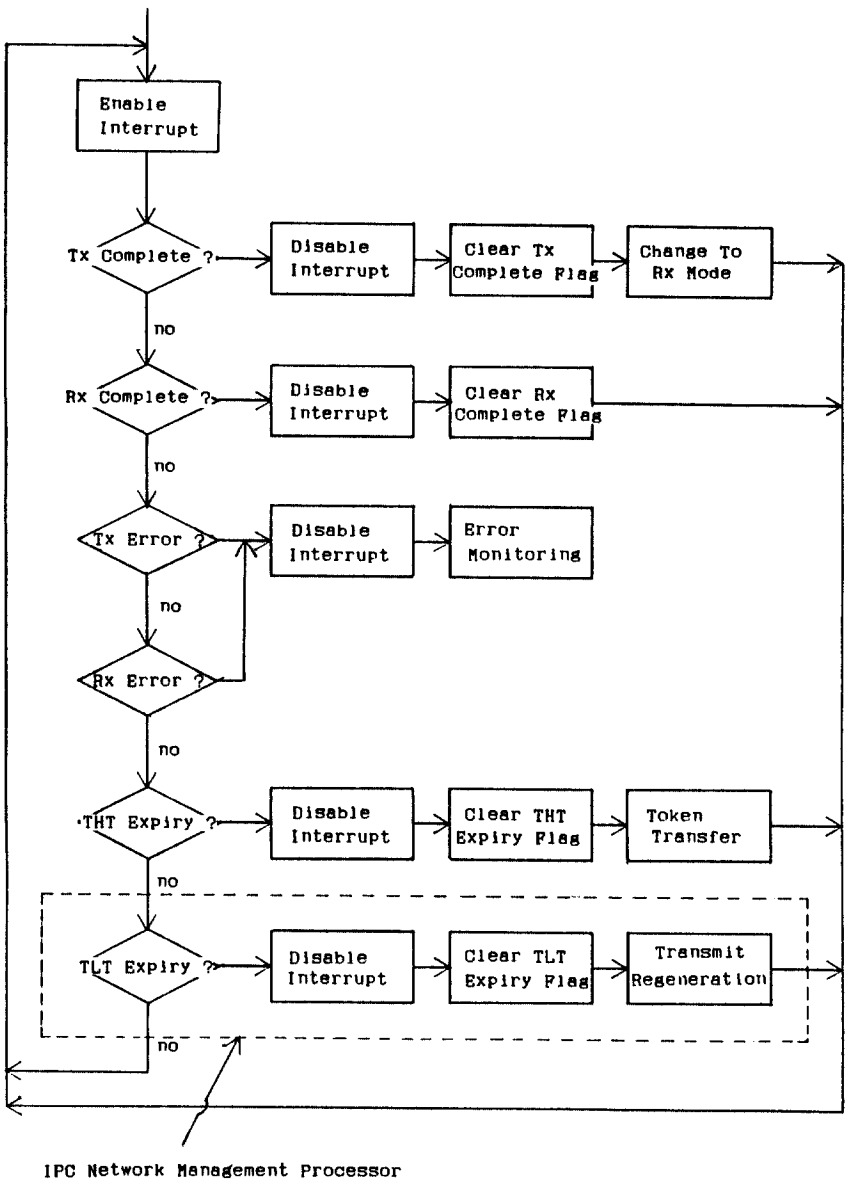


그림 8 job 분배 순서도
Flowchart of interrupt job handling.

른 address detection, frame error detection 등)에 따라 vectored interrupt level에서 각각 처리 하였으며 이에 따른 flowchart는 그림 9~10과 같다.

Ⅲ - 3 - 3 네트워크 관리 routine

네트워크 관리 routine에서는 각종 timer관리 및 데이타 송수신 routine에서 처리 하지 못하는 error 복구 및 재초기화의 작업을 수행한다. 본 논문에서는 어느 한 프로세서의 토큰이 독점함을 방지하기 위해 최대 패킷 전송 길이를 제한한 token holding timer (THT)를 각각의 프로세

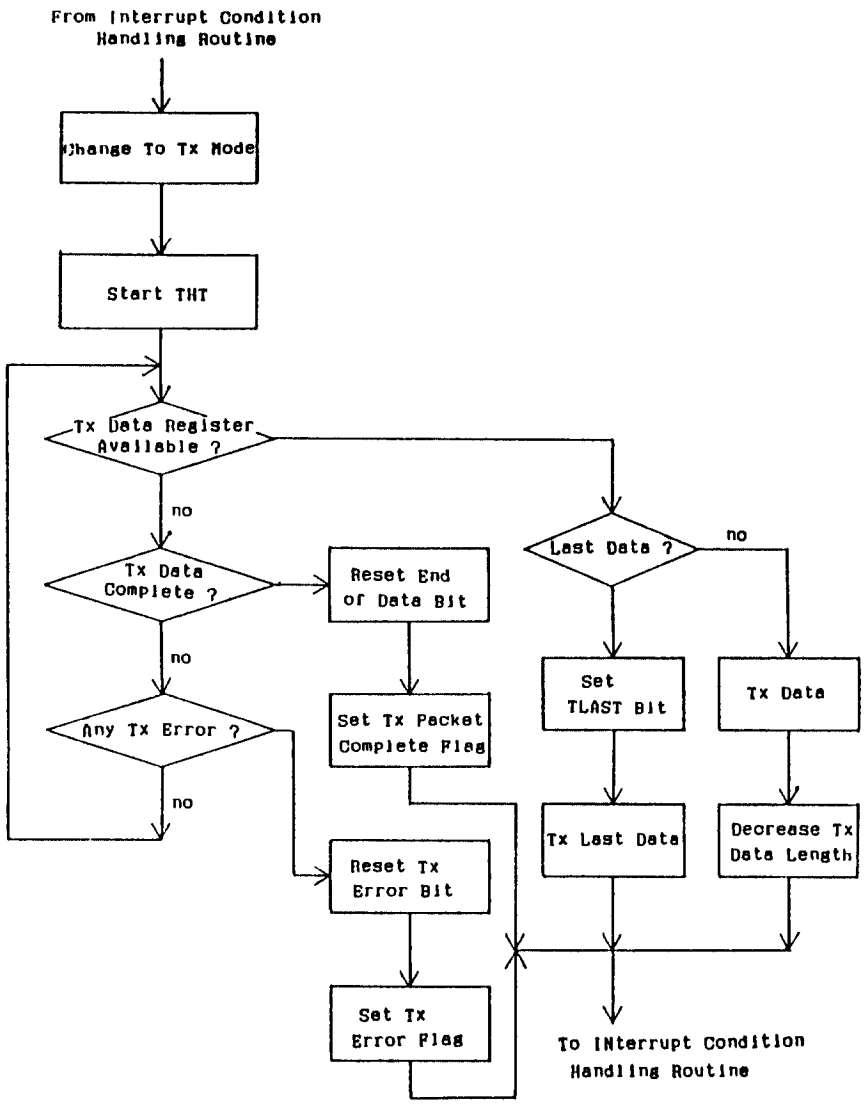


그림 9 데이터 송신 순서도
Flowchart of data transmission.

서부에 두어 네트워크의 균등성을 보장하였으며, 토큰 유실에 대처하기 위한 token loss timer(TLT)를 감시 프로세서에 두어 네트워크의 안전성을 도모하였다.

이와같은 timer의 값은 네트워크의 효율성에 큰 변수로 적용하게 되는데 IEEE 802.5 표준안[7]을 바탕으로 TLT는 n×THT+bit latency +propagation delay(여기서 n은 최대 네트워크 구성 프로세서 수)의 값으로 정하였고, THT는 최대 패킷 길이/전송속도의 값으로 하였다.

Ⅳ. 실험 및 고찰

본 논문에서 제안한 통신 예약 버스 방식의 효

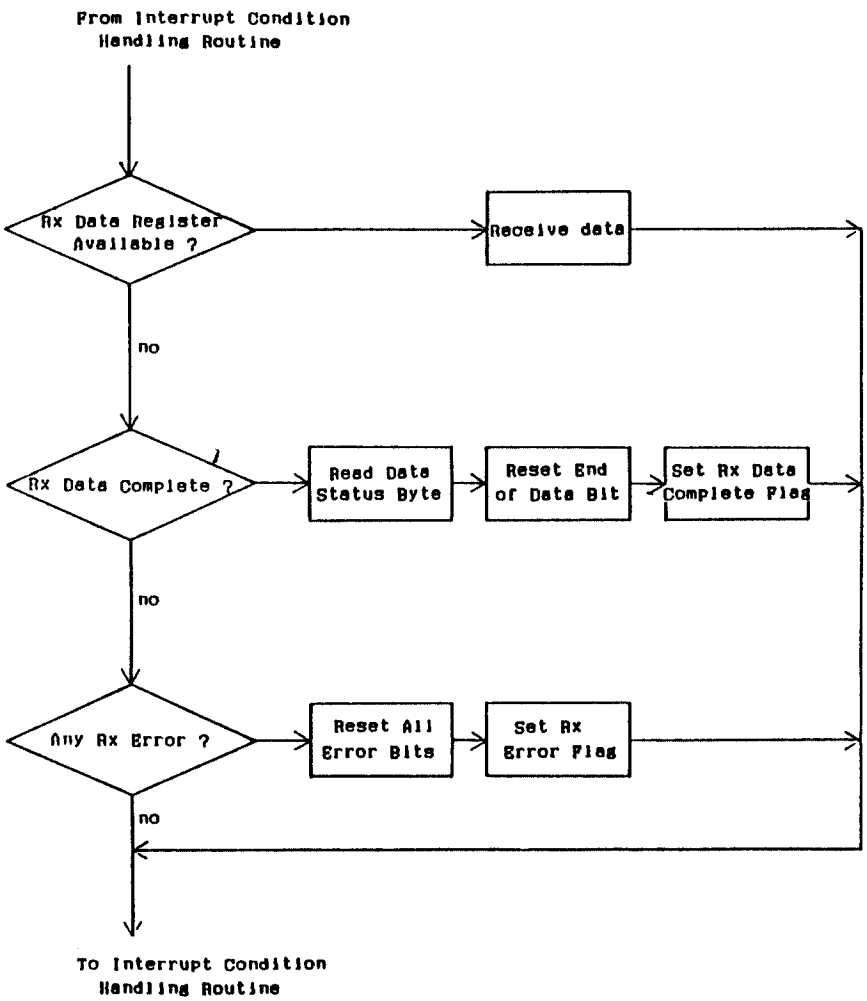


그림10 데이터 수신 순서도
Flowchart of data reception.

용성 및 타당성을 검증하기 위해 전장에서 기술한 내용을 토대로 하여 제작한 소규모 IPC 통신망을 연동하여 구성한 후 다음과 같은 실험을 수행하였다.

실험은 제작한 시스템의 관련 하드웨어 및 소프트웨어의 타당성을 확인하기 위한 일반적인 실험과 실제 환경하에서 시스템의 성능을 분석 하기 위한 성능 분석 실험의 순으로 진행하였으며 실험에 대한 각 프로세서 보드의 상태는 자체 개발한 모니터 프로그램을 이용하여 외부의 터미날에 표시하게 하였다.

1단계 실험으로 행한 일반적인 실험에서는 본 논문에서 제한한 통신 예약 버스의 가장 주요 부분이라 할 수 있는 토큰 발생, 획득 관련 회로와 전송 예약 및 해제 관련 회로를 아래와 같이 중점적으로 실험하였다.

- 토큰 감시.
- 토큰의 링 순환 과정 감시.

-전송 예약 후 토큰 획득 과정 감시.
-프로세서에서 전송한 데이터 프레임이 단일 전송로로 전송되는 과정 감시.
-수신 프로세서의 수신 데이터 프레임 확인.

프로세서를 정상적으로 구동시켜 프로세서의 데이터 송수신, 토큰의 순차적 전송 및 전송 예약이 효율적으로 동작되는지를 실험한 1단계 실험 결과 프로세서의 동작 상태가 통신 예약 버스의 물리적인 전송 특성을 갖는 프로토콜에 따라 효율적으로 동작된다는 것을 확인하였으며 본 회로가 실제의 상황 하에서 그대로 적용될 수 있으리라 본다. 본 논문에서 구성한 링 인터페이스 관련 회로는 기존의 TTL소자로 제작 되었으나 제시한 바와 같이 그 구조가 매우 간단하므로 손쉽게 custom IC 화 할 수 있을 것이다. 토큰의 유실은 감시 프로세서의 timer(TLT)에 의해 관리할 수 있으나 잡음(noise)등에 기인하여 이중의 토큰(double token)이 발생될 경우에 본 회로 및 소프트웨어는 아직 대처할 방안이 없는 것이 사실이다. 그러나 이러한 문제점은 low pulse의 토큰이 아니라 특수한 패턴을 갖는 즉, "010"등과 같은 패턴인 토큰으로 대처함으로써 그 발생 확율을 거의 없도록 할 수 있을 것이다.

제2단계 실험인 성능 분석 실험에서는 실험시스템을 실제의 환경에 적응하여 트래픽 변화에 따른 각 프로세서 보드의 동작 상태를 관찰하였다. 본 실험에서는 그림 11과 같이 random number generator 로 프로세서의 트래픽을 가변시키면서 각각의 트래픽에 따른 프로세서의 전송 지연 시간을 측정하였다. Random number 발생은 IBM 시스템에서 사용하는 random number generation algorithm을 사용하여 각 프로세서 보드에 할당된 트래픽 조절 조건과 비교하여 전체 통신망의 트래픽을 변화시켰다. 전송 지연 시간의 측정은 random number generator로 부터 발생된 random number와 프로세서에 미리 주어진 전송 조건 number가 일치되면 통신 예약을 한 후 CTC를 사용하여 매 20μsec마다 시간을 counting하여 토큰이 자신의 보드에 도착할 때까지 counting한 값을 트래픽 조건 별로 할당된 메모리에 기억시키는 방법을 사용하였으며 이를

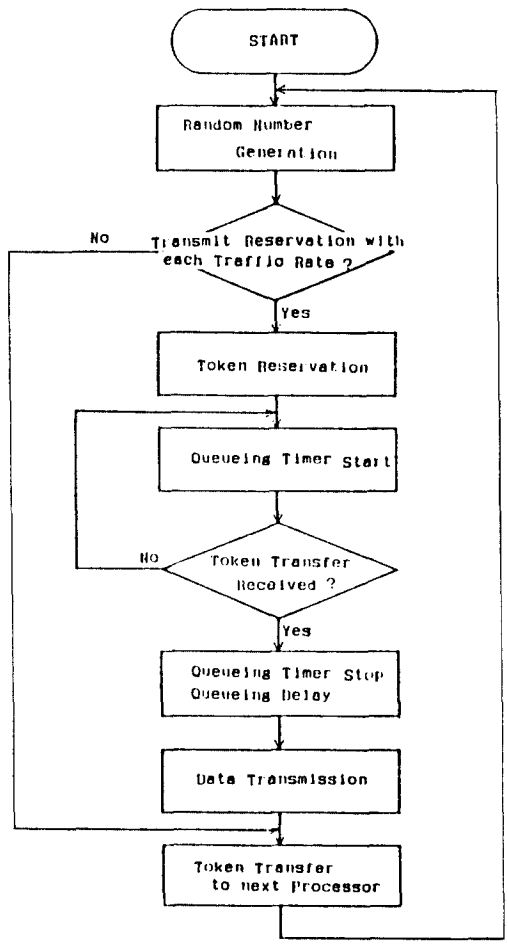


그림11 성능분석 실험절차
The procedure of performance analysis experiment

장기간 반복하여 그 평균치를 구하였다. 전송조건 number를 변경함으로써 전체 통신망의 트래픽을 변화시켰으며 이때 각각의 프로세서에 상이한 전송조건을 주었다.

표2에 이에 대한 값을 도시하였으며, 이에 대한 측정 결과 위와 같은 환경하에서 구한 실험치는 실험 대상이 소규모 실험실 모델이며 random number generation에 의한 트래픽 변화이기 때문에 실제 상황을 가정한 계산치와는 어느 정도의 차이가 있으나 전체적인 magnitude는 같은 수준을 유지하며 트래픽 변화에 따른 delay throughput의 graph도 그 전체적인 형태가 거의 일

표 2 트래픽 변화에 따른 전송지연시간
Transmission delay time with traffic change.

Traffic Rate(ρ)	Calculated value	Experimental value
0.079	100.6	114.6
0.137	184.2	248.3
0.218	293.2	352.1
0.288	386.9	471.8
0.370	497.1	554.4
0.452	606.8	669.6
0.521	701.2	773.5
0.583	784.3	865.7
0.668	897.7	976.8

*단위 : micro second

치함을 확인하였다.

V. 결 론

통신기기의 지능화는 시스템내에 다수의 프로세서를 요구하게 될 것이므로 근접한 프로세서 상호 간에 효율적인 메세지 교환을 담당하는 프로세서간 통신망에 대한 중요성이 강조되고 있다. 그 한 예로써 교환기를 고려하여 볼 때 교환기 제어 구조는 독립된 하드웨어와 소프트웨어를 갖는 다수의 프로세서에 제어 기능을 분산시키는 분산 제어 방식이 시스템의 확장성이나 효용성 측면에서 가장 이상적인 방식이라 할 수 있다. 따라서 본 논문에서는 이러한 환경 즉, 근접된 프로세서간의 효율적인 통신을 관장하는 프로세서간 통신 방식으로 앞서 제안한 통신 예약 버스에 기초하여 이에 적용할 수 있는 하드웨어 및 소프트웨어를 개발하여 소규모 실험실 IPC통신망을 구성하였으며 이에 대한 실험을 통하여 본 방식의 타당성 및 실현 가능성을 제시하였다.

IPC 통신망 실현 시 가장 중점적으로 고려된 사항은 전송 효율의 극대화, 전송 프로토콜의 간편화등이며 앞서 제한한 통신 예약 버스의 기본 개념을 충분히 수용하도록 설계 제작하였다. 이와같이 자체 제작한 4개의 프로세서 보드를 연동 실험한 결과 프로세서의 동작 상태가 통신 예약 버스의 물리적인 전송 프로토콜에 따라 효율적으로 동작된다는 것을 확인할 수 있었으며 실제적인 시스템의 성능을 실험한 결과 본 논문에서 제안한 예약 버스가 타 방식에 비해 우수한 성능을 갖고 있다는 사실을 인식할 수 있었다. 또한 관련된 하드웨어 및 소프트웨어가 대단히 단순화될 수 있음을 확인할 수 있었으며 custom IC화, DMA(Direct Memory Access) 등을 도입할 경우 IPC를 위한 전체적인 하드웨어 및 소프트웨어가 시스템 전체에서 차지하는 비중이 더욱 감소될 것으로 예측된다. 앞으로 이중 토큰의 발생에 대비한 연구만 추가된다면 본 방식 및 관련 회로는 실제의 시스템상에서 적절히 활용될 수 있을 것으로 기대된다.

끝으로 본 논문은 한국 전자 통신 연구소의 위탁 연구 과제로 진행 되었음을 밝히며 본 연구수행을 위해 많은 도움을 주신 관계자 여러분께 감사의 뜻을 전한다.

參 考 文 獻

(1) 황 대환 외 4인, "근접한 프로세서간 통신방식에 관한 연구," 한국통신학회논문지, vol. 12. no. 6, Dec. 1987

(2) H. Ebel H. Helmrich, "ISDN Terminals-Basic facts," Telcom report, Aug. 1985, pp64-pp68

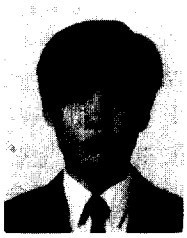
(3) G. B. Bhusri, "Considerations for ISDN planing and Implementation," IEEE Com. magazine vol. 22, no. 1. 1984.

(4) G. Comi, G. Gragia, "MIDOS : A distributed operation system for reliable supervision of telecommunication system," ICC '84 AMSTERDAM, MAY 1984.

(5) Jan. Vytopil and P. Muller, "Communication architecture of TPC 16, Highly reliable, distributed system," ICC '84 AMSTERDAM, MAY 1984.

(6) Rockwell, R68561 "Multi-Protocol Communications Controller(MPCC)," Manual.

(7) IEEE Standard 802. 5 Token Passing Media Access Control.



金 祜 鑽(Ho Geon KIM) 準會員
1964年 1月17日生
1986年 2月：成均館大學校工科大學電子工學士 卒業(工學士)
1988年 2月：成均館大學校 大學院 電子工學科 通信工學 專攻(工學碩士)

朴 永 德(Yung Duck PARK) 正會員
1957年11月24日生
1984年 2月：成均館大學校 工科大學 電子工學科(工學士)
1983年 9月～1985年 2月：三星電子株式會社 研究員
1987年 2月：成均館大學校 大學院 電子工學科(工學碩士)
1987年 3月：成均館大學校 大學院 電子工學科 博士課程 在學中

金 宣 衡(Sun Hyoung KIM) 正會員
1955年 3月 9日生
1979年 2月：成均館大學校 理工大學 電子工學科(工學士)
1981年 2月：成均館大學校大學院 電子工學科(工學碩士)
1988年 2月：成均館大學校 大學院 電子工學科(工學博士)
1983年 3月～現在：인덕工業專門大學助教授

曺 圭 燮 (Kyu Seob CHO) 正會員
1951年5月3日生
1974. ：成均館大學校 電子工學科(工學士)
1976. 2：成均館大學校 大學院 電氣工學科(工學碩士)
1977. 3：韓國電子通信研究所
～現在 責任研究員
1984. 3：成均館大學校 大學院 電子工學科 博士課程 在學中

朴 炳 哲(Byung Chul PARK) 正會員
1930年4月30日生
1957. 9：서울大學校 工科大學 通信工學科 卒業(工學士)
1975. 2：仁荷大 大學院 電氣工學科(工學博士)
1972. 3：成均館大學校 教授 電子工學科
1980. 9：日本 東京大學 外國人研究員(1年間)
1986. 12～現在：成均館大學校 工科大學長