

다중처리를 지원하는 CAT 인터페이스에 관한 연구

正會員 田 東 根* 正會員 盧 承 煥* 正會員 車 均 鉉*

Realization of CAT Interface Supporting Multitask

Jeon Dong Keun*, Soong Hwan Ro*, Kyun Hyon Tchah* *Regular Members*

요 약

이 논문에서는 멀티태스킹을 지원하는 CAT 인터페이스를 구현하였다. 컴퓨터와 계측기를 연결하기 위해 인터페이스 카드를 제작하고 구현하였다. OOP와 GUI를 사용한 컨트롤과 디스플레이는 C++로 프로그램되었다.

스펙트럼 분석기와 전력계가 제어할 계측기로 선택되었다. 9개의 모듈이 여러 가지 자원을 공유하도록 구성되었다. 또한, 여러대의 계측기를 동시에 측정하기 위해 멀티태스킹이 구현되었다. 멀티태스킹은 시간 분할 DOS환경에서 수행되었다. 멀티태스킹은 프로세싱 기법으로 트레드 방식을 사용하였고 스케줄링으로 라운드 로빈방식을 사용하였다. 다른 목적계측기를 위해서는 목적 계측을 구동할 소프트웨어만 합하면, 시스템은 더 많은 계측기를 컴퓨터에 의해 동시 측정할 수 있다.

사용자는 전문가적인 지식없이 시간과 오차를 줄일 수 있다.

ABSTRACT

In the paper, a CAT interface supporting multitask is realized. To interface a computer with measuring instruments, a GPIB card is designed and implemented. Controlling and displaying software using OOP and GUI are programmed with C++. A spectrum analyzer and a power meter are chosen as object instrument to be controlled. Total 9 modules are configured to manage the various resources and each module in integrated system.

Also in case that several instruments are used, the system is realized to be capable of multitasking to exchange the data mutually. The multitasking is implemented under the time-sharing DOS environment. Thread-based method is used for processing, and Round Robin method for scheduling.

Provided proper software modules for other object instruments are integrated, the system can control more measuring instruments simultaneously by the computer. Users can save the time and errors even without expert knowledge.

*高麗大學校 電子工學科
Dept. of Electronic Eng., Korea University
論文番號 : 92-141 (接受1992. 8. 6)

I. 서 론

VISI 기술의 발전과 더불어 특별한 사양을 갖춘 제품이 날로 증가하고 있는 추세이다. 그러나, 개개의 제품에 대한 검사를 하는 데에는 많은 시간과 비용이 소모된다. 이에 따라 최근 자동검사 시스템이 도입되고 있으며 그 대표적인 시스템이 CAT이다. CAT를 이용할 경우 컴퓨터의 도움으로 검사과정을 자동화할 수 있으므로 검사비용과 시간을 대폭 절약할 수 있고 다양한 계측장비를 VXI 버스(VME Extended for Instrument)와 GP-IB(General Purpose-Interface Bus)에 연결함으로써 집중적이고 효과적으로 제품의 특성을 조사하고 검사할 수 있다.

이와 같이 IEEE에 의해 표준화된 VXI버스와 GP-IB의 기능은 CAT로 이어져 VXI버스와 GP-IB 등을 이용하고 여기에 컴퓨터를 부가한 계측장비를 구성하여 각종 계측장비를 연결, 통합계측을 하고 있으며 이러한 CAT기술은 컴퓨터가 갖는 고속처리 능력을 이용하여 검사시간을 감소시킨 만큼 생산성의 향상에 크게 기여하고 있다.

본 논문에서는 이러한 CAT를 적용하여 컴퓨터와 계측모듈을 대상으로 상호 인터페이스를 구현하였고 다양한 계측장비와의 인터페이스를 구성하기 위하여 OOP(Object Oriented Programming)를 이용하여 인터페이스용 소프트웨어를 개발하였다[21-25].

또한 동시에 여러대의 계측기를 측정하기 위하여 다중처리 프로그램을 구현하였다. 다중처리 프로그램 구현에서, 중앙처리장치 스케줄링은 라운드 로빈(round robin)방식을 사용하였고 프로세스 기법은 트레드 베이스(thread based)방식을 사용하였다.

II. CAT 및 GUI

1. CAT

최근 컴퓨터 사용의 확산과 기능의 향상은 여러 분야에 큰 파급효과를 주었고 새로운 개념의 도입을 촉진하였으며, 특히 시스템 설계와 계측기 시스템은 컴퓨터의 사용영역 확대에 힘입어 눈부신 발전을 거듭하였다. CAD(Computer Aided Design), CAM(Computer Aided Manufacturing), CAE(Computer Aided Engineering), CASE(Computer Aided Software Engineering)와 CAT 등이 그 대표

적인 예로서 이들은 모두 컴퓨터의 프로그램과 통계 및 수치해석, 제어 기능을 이용한 자동화들이다[5][6][9]. 이 중에서도 계측기 산업과 관계있는 CAT 분야에 대해서 많은 연구가 이루어지고 있다.

CAT는 제품의 검사나 혹은 어떤 실험의 측정을 컴퓨터로 자동화시키는 것이며 검사나 실험은 원하는 결과를 얻기 위해 실험 대상에 대한 다양한 정보를 추출하는 과정이다. 따라서 CAT를 도입함으로써 다음과 같은 잇점을 얻을 수 있다.

- 자동화에 의하여 실험에 소비되는 시간을 절감
- 실험의 절차를 일치시켜 실험의 일관성을 부여
- 품질의 통계적 정보를 즉각적으로 확보
- 실험 환경이 직접적인 수동작을 허용하지 않을 경우
- 실험 대상의 분석을 자동화하므로 빠른 품질 개선이 가능

그 밖에도 컴퓨터의 통계자료 분석능력이나 프로그래밍 능력은 계측기기 시스템의 자동화를 가속화하는 주요요인이다. 현재 사용되고 있는 CAT는 크게 두 부류로 분류할 수 있다.

1.1 디자인 시스템

디자인 시스템의 역할은 테크니컬 컴퓨터를 근간으로 기계적인 부문, 즉 전기적인 하드웨어, 펌웨어 및 소프트웨어를 개발하는 것이라 할 수 있다. 설계 자동화(design automation)를 추구하는 이 부문은 신제품의 질을 높이고 디자인 사이클(design cycle) 및 생산 라인에서 설계자료를 신속, 정확하게 하는데 그 목적이 있다. 여기서 사용하는 컴퓨터 시스템 플랫폼(computer system platform)의 OS로서는 UNIX나 MS-DOS가 주종을 이루고 있다. 현재 개발되어 있는 전문 시스템은 EDS(Electronic Design System), MDS(Microwave Design System) 등이 있다.

1.2 테스트 시스템

테스트 시스템은 계측기와 컴퓨터를 결합시켜 테스트 절차를 자동할 수 있다. 예를 들어 온도와 습도의 변화에 따른 프로토타입 제품의 동작을 검사하는 환경 검사는 계측기에 컴퓨터를 연결시킴으로써 장시간에 많은 데이터 추출이 가능하며, 컴퓨터의 계산 기능을 이용함으로써 여러가지 측면의 분석도 가능

하다.

계측기와 컴퓨터를 연결시키는 가장 전형적인 방법은 GP-IB와 VXI 버스를 연결하여 인터페이스를 구성하는 시스템이다[1-9]. 컴퓨터와 명령어 모듈 사이에서는 GP-IB 버스가 쓰이고 커맨드 모듈과 각 메시지 모듈 사이에는 VXI 버스가 쓰이는 일반적인 CAT 시스템의 기본구조가 그림 1에 나타내었다.

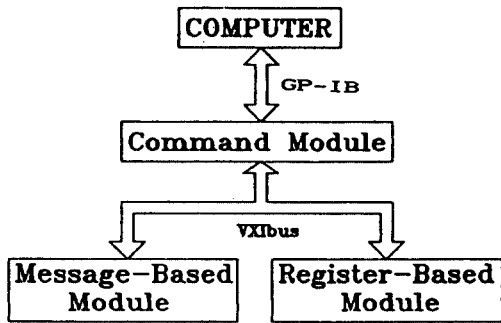


그림 1. CAT 시스템 구조
Figure 1. Structure of CAT system

표 1. 다중처리를 위한 프로세스의 사용법

Table 1. Comparison of processing methods

프로세스 베이스 다중처리 시스템	트레드 베이스 다중처리 시스템
- 동시에 둘 이상의 프로세스를 처리하는데 유용 장점 : 다수의 프로그램이 CPU공유 단점 : 수행시간을 빠르게 향상시키지 못함 예 : UNIX	- 동시에 단일프로그램의 여러 부분을 실행하는데 유리 장점 : 수행시간을 단축시킴 단점 : 하나의 프로그램만을 수행시킴 예 : DOS

는 다음과 같은 방법들이 있다[25].

2.1 단계별 분류

중앙처리장치 스케줄링에는 그림 2와 같이 네 가지의 중요한 단계가 있다.

II. 다중처리를 위한 소프트웨어 개발 환경

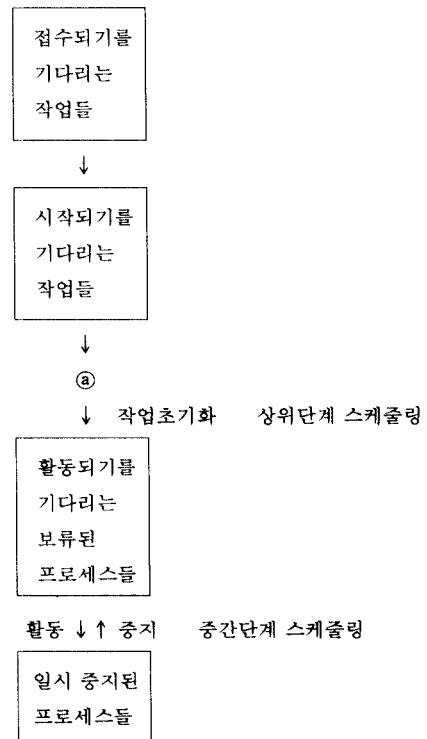
1. 프로세스 기법

다중처리는 보통 OS /2, UNIX, 또는 대형의 메인 프레임 OS로 실현된다[19-23]. 여러 작업이 하나의 시스템에서 실행되면 OS는 CPU시간을 나누어서 처리한다. 다중처리를 구성하는 단일 프로그램의 많은 부분들은 OS로부터의 도움없이 처리가 된다. 논문에서 구현한 다중처리 커널은 일반 사용자가 쉽게 사용할 수 있도록 하기 위하여 Turbo C를 이용하여 DOS 하에서 실행되도록 설계하였다. 다중처리에서 프로세스를 사용하는 방법에는 두 가지 방법이 있는데 두 가지 방법의 차이와 내용을 표 1에 나타내었다.

논문에서 구현된 다중처리 시스템은 단일 프로그램으로 여러 부분을 실행하도록 구성되어 있기 때문에 프로세스 기법으로 트레드 방법을 사용하였다.

2. 스케줄링

다중처리 시스템은 임의의 작업에서 다른 작업으로 수행을 바꾸는 루틴을 가지고 있다는 장점이 있다. 이 루틴을 스케줄러라고 부르며 스케줄링기법에



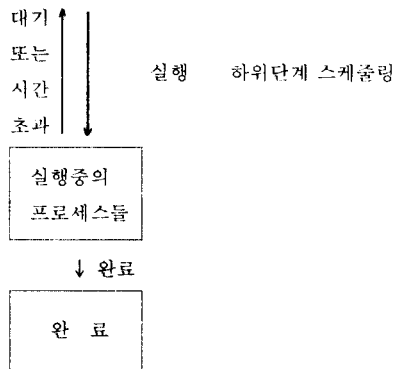


그림 2. 중앙처리장치 스케줄링

Figure 2. Scheduling of central processing unit

3. 프로세스 스케줄링 알고리즘

3.1 FCFS 스케줄링

FCFS(First Come First Served) 기법은 가장 간단한 스케줄링 방식으로서 비선점 스케줄링 방법 중의 하나이다. 프로세스들은 대기 큐에 도착한 순서에 따라 중앙처리장치를 할당받으며, 일단 프로세스가 중앙처리장치를 차지하면 완료될 때까지 수행된다. FCFS기법에서 모든 프로세스가 동일하게 취급되므로 응답 시간의 차이가 적게 나며, 대부분의 다른 방식에 비하여 작업 완료 시간을 예측하기가 용이한 스케줄링 방식이다.

많은 스케줄링 방식들이 프로세스들을 우선 순위에 따라 수행하지만, 같은 순위를 가진 프로세스들 간에는 FCFS방식에 의해서 중앙처리장치를 사용하도록 한다.

3.2 SJF 스케줄링

SJF(Shortest Job First)는 기다리고 있는 작업중에서 수행시간이 가장 짧다고 판정된 것을 먼저 수행하는 비선점 스케줄링 방식이다. 이 방식은 FCFS보다 평균 대기 시간을 감소시키는 반면 특히 큰 작업들에 대해서는 대기시간들의 분산이 FCFS방식에 비하여 크고 예측하기가 어렵다. SJF는 오버 헤드면에서 볼 때는 짧은 작업일수록 더 좋은 서비스를 받을 수 있다.

이 방법에서는 특히 작업의 우선 순위를 위한 일반적인 규약이 없으므로 SJF의 서비스를 위한 다음 작업의 선택은 작업시간이 가장 짧은 것으로 선택한다.

이것은 대기하고 있는 작업들의 수를 줄이고 큰 작업 뒤에서 기다릴 작업의 수를 줄임으로써 그 결과 SJF는 작업들이 시스템을 완전히 통과하는 평균 대기 시간을 최소화 할 수 있다.

SJF에서 나타나는 문제점은 수행하게 될 작업이나 프로세스가 얼마나 긴 것인가를 정확히 알아야 하는데 이 정보는 얻기가 어렵다는 점이다. 가장 좋은 방법은 시간 예측을 사용자에게 의존하는 것으로서 규칙적으로 같은 작업들이 수행되는 상황에서는 적당한 예측이 가능하다. 하지만 개발 중인 프로그램의 경우에는 사용자가 자신의 프로그램이 처리되는데 시간이 얼마나 소요될 것인가를 모르는 경우가 대부분이다. 따라서 SJF는 FCFS처럼 비선점 방식하므로 적절한 응답시간이 보장되어야 하는 시분할 방식의 시스템 상황에서는 적당하지 못하다.

3.3 라운드 로빈 스케줄링

라운드 로빈 스케줄링은 시분할 시스템을 위하여 고안된 선점 스케줄링 방식의 하나이다.

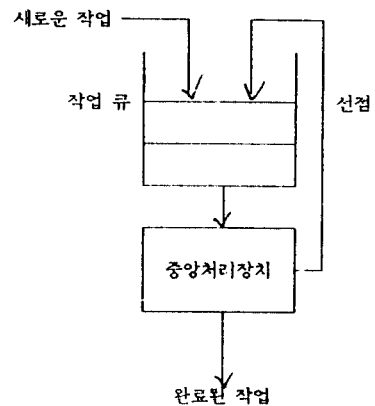


그림 3. 라운드 로빈 스케줄링

Figure 3. Round-robin Scheduling

이 방법은 그림 3과 같이 FCFS에 의해서 프로세스들이 내보내어지며 각 프로세스는 같은 크기의 중앙처리장치를 할당받는다. 만약 프로세스가 중앙처리장치 시간이 만료될 때까지 처리를 완료하지 못하면 그 중앙처리장치는 대기중인 다음 프로세스로 넘어가며, 수행 못한 프로세스는 준비 완료 리스트의 가장 뒤로 보내어진다.

IV. 인터페이스 및 프로그램의 구현

1. 인터페이스 처리부 소프트웨어 구조

본 절에서는 계측기를 제어하기 위해 구현한 소프트웨어에 대해 기술한다. 소프트웨어 개발은 DOS상에서 객체지향 언어인 C++를 이용하였고 CRT디스플레이를 위하여 GUI방식을 이용하였다. 논문에서는 HP8590A 스펙트럼 분석기와 HP437B 전력계를 운용 계측기로 이용하였다. 그림 4는 인터페이스 처리의 계층적 구조를 나타내고 있다.

화면에 나타나는 모든 정보를 제어하기 위해서 화면에 윈도우를 설정하고 윈도우 내부를 관리하는 기능이 윈도우 처리부다. 윈도우 처리부는 윈도우에 모든 정보를 표현하며 GUI를 제공하기 위해서 마우스와 소프트웨어 버튼, 한글의 명령어를 사용한다.

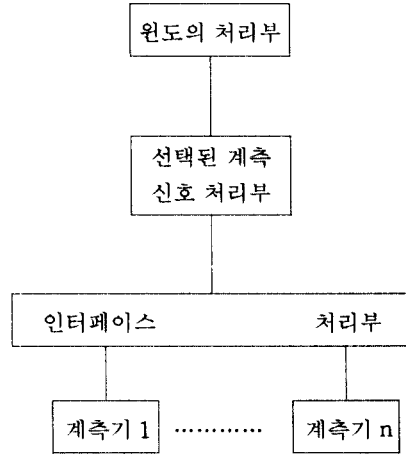


그림 4. 인터페이스 처리부
Figure 4. Interface processing unit

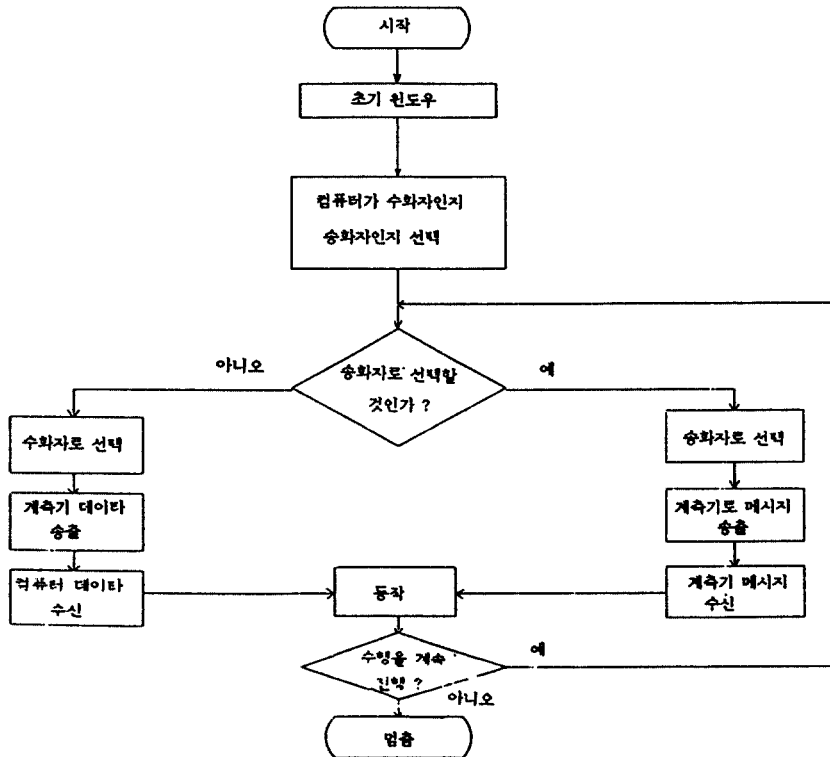


그림 5. 인터페이스 처리부의 송수신 흐름도
Figure 5. Flow chart of processing unit for transmitting and receiving messages

1.1. 인터페이스 처리부

데이터와 명령어를 처리하는데 있어서 인터페이스 처리부에서는 컴퓨터와 계측기간에 그림 5와 같이 메시지를 송수신 한다.

그림 5에 보는 바와같이, 컴퓨터가 송신자가 될 것인지 수신자가 될 것인지를 선택한 후 송신 상태에서는 송신 메시지를 송출하여 계측기가 송신명령에 따라 동작하도록 하며, 수신 상태에서는 컴퓨터가 데이터를 수신하여 데이터 값을 표시하도록 구현했다.

2. 다중 처리 소프트웨어 구조

Turbo C는 인터럽트 서비스루틴을 위해 제공되는 인터럽트 함수형 수정자가 있어서 인터럽트형으로 함수를 선언하면 컴파일러에서는 SP(stack pointer)와 SS(stack segment)를 제외한 모든 레지스터들을 자동적으로 보존시키고 인터럽트 복귀명령으로 함수를 종료한다. 인터럽트가 실행되었 때 8086은 자동으로 플래그, CS, 그리고 인터럽트 포인트 레지스터들을 푸쉬(push)한다. 그림 6은 작업교환과 다중 처리를 위한 전체작업 구조를 보이고 있다.

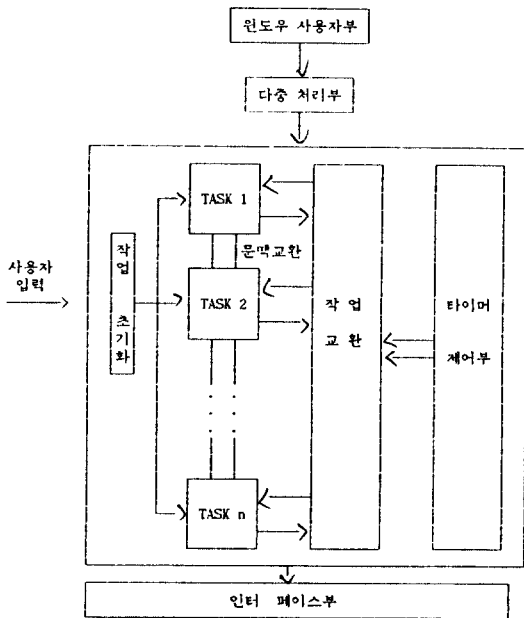
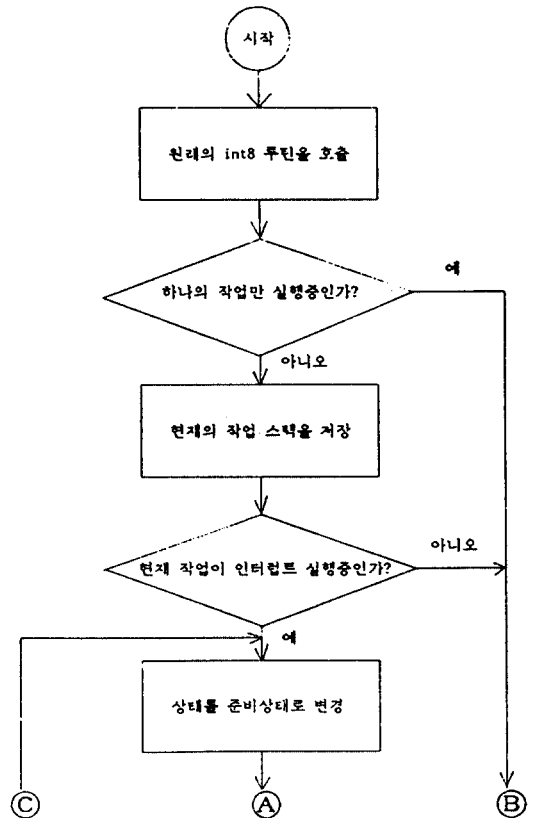


그림 6. 다중처리 전체작업구조
Figure 6. Entire task structure for multitask

작업이 바뀌는 과정은 인터럽트에 의해 수행된다. 예를 들어 시스템에서 A와 B라고 하는 두 개의 작업이 있고 현재 작업 A가 실행중이라면, 스케줄러는 타이머가 인터럽트 8을 일으킬 때마다 2개의 작업사이에서 앞뒤로 작업을 바꾸어가며 실행하며 이와 같은 스케줄러 흐름도를 그림 7에 나타내었다.

3. 프로토콜 분석 알고리즘

프로토콜 분석 알고리즘은 버스상의 정보를 분석하여 제대로 정보를 전달하는가를 알아보는 알고리즘으로 먼저 버스의 상태를 모니터링 할 수 있는 버스 모니터 알고리즘이 필요하다. 모니터 알고리즘이 그림 8에 보인다.



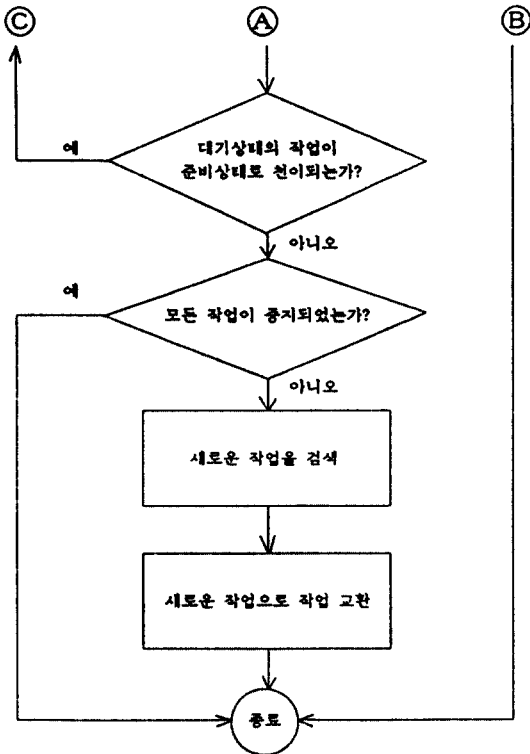


그림 7. 스케줄러 흐름도
Figure 7. Flow chart of scheduler

그림 8. 모니터 알고리즘 흐름도
Figure 8. Monitor algorithm flow chart of bus state

여기에서는 버스의 상태를 정확히 읽어서 기록하고 화면에 표시하는데, 표시하는 단위는 3선 핸드셰이크를 행할 때 각 상태를 표시하는 것이다. 만약 버스의 상태가 데드록(deadlock)이 되면 계속해서 3선 핸드셰이크가 이루어지지 않으므로 더 이상 버스의 상태가 변화하지 않는다. 이 점을 이용하여 일정시간 이상 버스의 상태변이가 없으면 버스에 데드록이 발생한 것으로 간주하고 데드록 확인을 하게 되는데 한번 더 일정 시간동안 지켜보다가 더 이상 아무변화가 없으면 버스가 데드록이 되었음을 표시한다. 그리고 버스에 데드록이 일어나지 않았을 경우에는 계속해서 정해진 조건만큼 버스의 상태를 기록하고 수행을 완료한다.

정보 송수신 모드의 프로토콜 분석 알고리즘은 버스의 모니터 알고리즘에 의해서 버스의 데드록이 발생하였다는 보고를 받았을 때 사용하는 것인데 버스 상에 접속된 각 기기들을 버스에서 분리하고 각각의 기기들이 갖고 있는 정보의 송수신 프로토콜을 알아내는 것이다. 이에 대한 알고리즘은 그림 9와 10에 나타내었다.

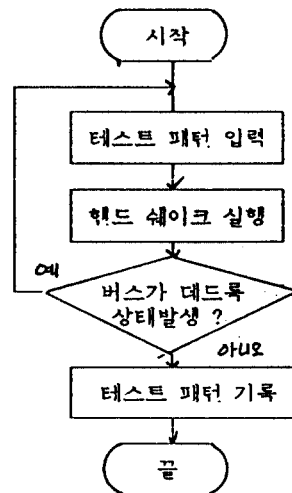
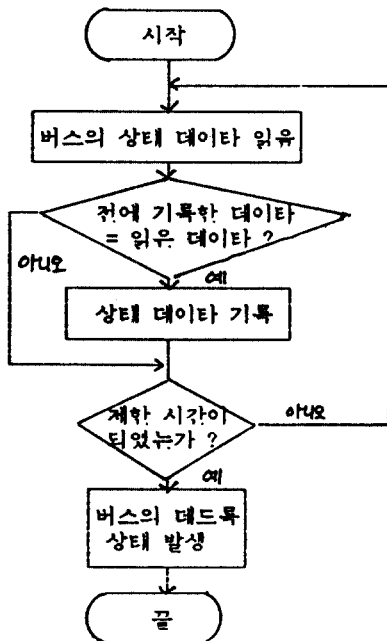


그림 9. 명령어 송수신 모드 알고리즘
Figure 9. Algorithm flow chart of command transceiver mode

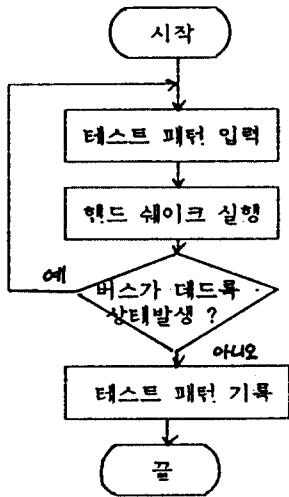


그림 10. 데이터 송수신 모드 알고리즘
Figure 10. Algorithm flow chart of data transceiver mode

4. 프로토콜 분석기 하드웨어

인터페이스의 프로토콜 분석을 위한 하드웨어의 블록 다이어그램은 그림 11과 같다.

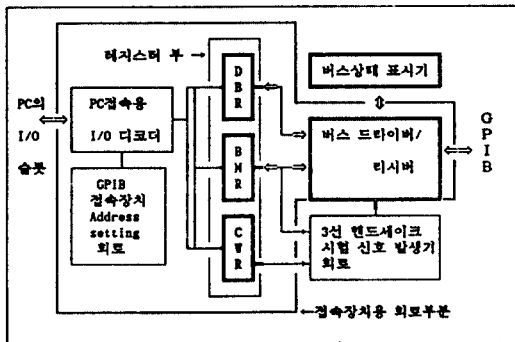


그림 11. 프로토콜 분석기 블록 다이어그램
Figure 11. Block diagram of protocol analyzer

제작된 하드웨어는 3부분으로 나누어지는데 그림 12는 버스상태 표시부와 3선 핸드셰이크 시험신호 발생기회로와 커넥터가 모여있는 윗부분 기판이며 그림 13은 나머지 기판이 모여있는 아랫부분 기판의 사진이다. 제작된 프로토콜 분석기의 전체구성은 14와 같다.



그림 12. 제작된 프로토콜 분석기(상)
Figure 12. Designed protocol analyzer (upper)

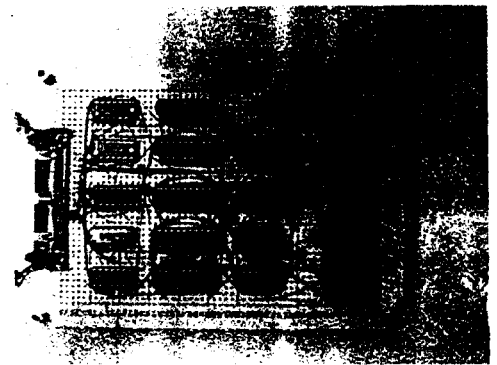


그림 13. 제작된 프로토콜 분석기(하)
Figure 13. Designed protocol analyzer (lower)

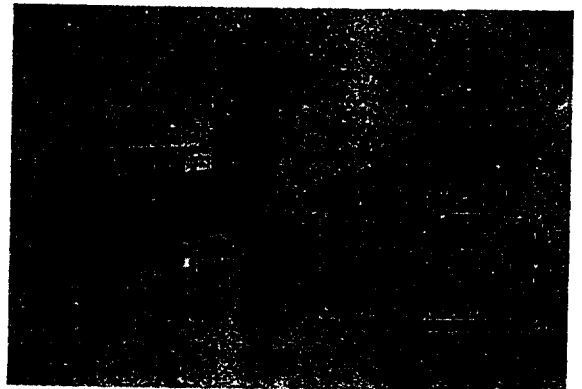


그림 14. 제작된 프로토콜 분석기(전체구성)
Figure 14. Designed protocol analyzer(entire)

V. 실험 및 검토

1. 멀티 태스크를 이용한 계측기 제어 소프트웨어 구현

논문에서 개발된 멀티태스크를 이용한 계측기 제어프로그램의 전체적인 구조는 그림 15와 같다.

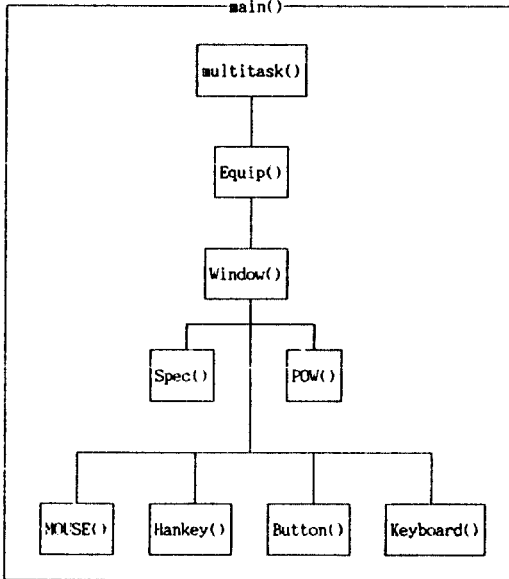


그림 15. 멀티태스크를 이용한 계측기 제어프로그램의 구조
Figure 15. Structure of control program using multitask for instruments

그림 15에 나타난 소프트웨어구조에서 다중 프로그램은 모든 다른 프로그램을 제어한다. 즉 제어할 계측기를 선택하거나 화면에서 제거시키는 역할을 한다. 또한 DOS상에서 멀티태스크를 실행하거나 단일 프로그램을 제어시키는 역할을 한다.

Spec() 루틴은 스펙트럼 분석기를 초기화시키고 구동시키는 역할을 하며 제어 대상이 되는 주파수, 대역폭, 이득, 감쇠에 대한 제어를 담당한다. Pow() 루틴은 파워 미터를 구동시키는 역할을 하며 제어대상이 되는 주파수, 전력, 조정 요소(cal factor)를 담당한다. 사용자는 단지 마우스만으로 모든 제어를 할 수 있다.

2. 전체 시스템 구성 및 실험

본 논문에서 구현된 GP-IB 카드는 기능상 인터페이스 기능 신호처리부와 버스라인 드라이버/리시버 IC에 의해 구성되는데(그림 16 참조), 인터페이스 기능 신호처리부에서는 4장에서 설명한 프로토콜과 프로토콜을 이루기 위한 기본적인 10가지 서브세트를 지원하는 전자회로로 구성되어 있다[11-14]. 버스라인 드라이버/리시버 IC는 GPIB버스라인 케이블에 공급되는 전압/전류 특성에 맞도록 신호를 변환시키는 역할을 담당한다.

인터페이스 기능 신호처리부의 회로를 어떻게 설계했느냐로 접속장치 구성형태를 분류할 수 있는데 범용부품만으로 구성하는 것과 전용 칩(LSI)으로 구성하는 것이 있다[9]. 범용부품만으로 구성할 때는 대부분 쉽게 구입할 수 있는 TTL IC를 사용해서 해당회로를 구성하게 되는데, 단점으로는 GP-IB를 세밀히 이해하지 못하면 만들 수 없는 점과 운용소프트웨어에서 비트처리 수준의 복잡한 프로그램을 해야만 하는 것인데 프로그램상의 실수가 발생하면 GPIB규격에 대한 호환성을 상실하게 된다[9-10]. 전용 칩을 사용해서 구성할 때는 각 서브세트의 기능이 칩 자체내에 회로로 구현되어 있으므로 운용 소프트웨어 작성과 GP-IB규격에 대해서는 약간의 유리한 점이 있으나 국내에서는 구하기 어렵고 가격이 비싸다는 단점이 있다. 여기서 사용되고 있는 버스라인 드라이버/리시버 IC는 데이터 선용으로 텍사스 인스트루먼트사의 75160A와 멀티콘트롤러가 가능한 75162A를 사용하였고 CPU로서 TMS9914A를 사용하였다[6-11].

실험에 의한 전송방식의 타이밍은 그림 17과 같다.

3. 실험 결과

3.1. 결과

현재의 CAT 시스템은 여러대의 계측기를 단일 프로그램으로 구동시키도록 구성되었다. 이러한 문제로 인하여 하나의 실험보드를 테스트하기 위해서는 측정할 실험 보드에 알맞은 계측기를 구동하기 위해서 컴퓨터는 한번에 하나의 프로그램만을 실행시키도록 되어 있기 때문에 동시에 두 대의 계측기를 구동하지 못한다. 예를 들어, 하나의 실험보드에 주파수와 전력을 측정할 경우 현존의 시스템은 주파수만

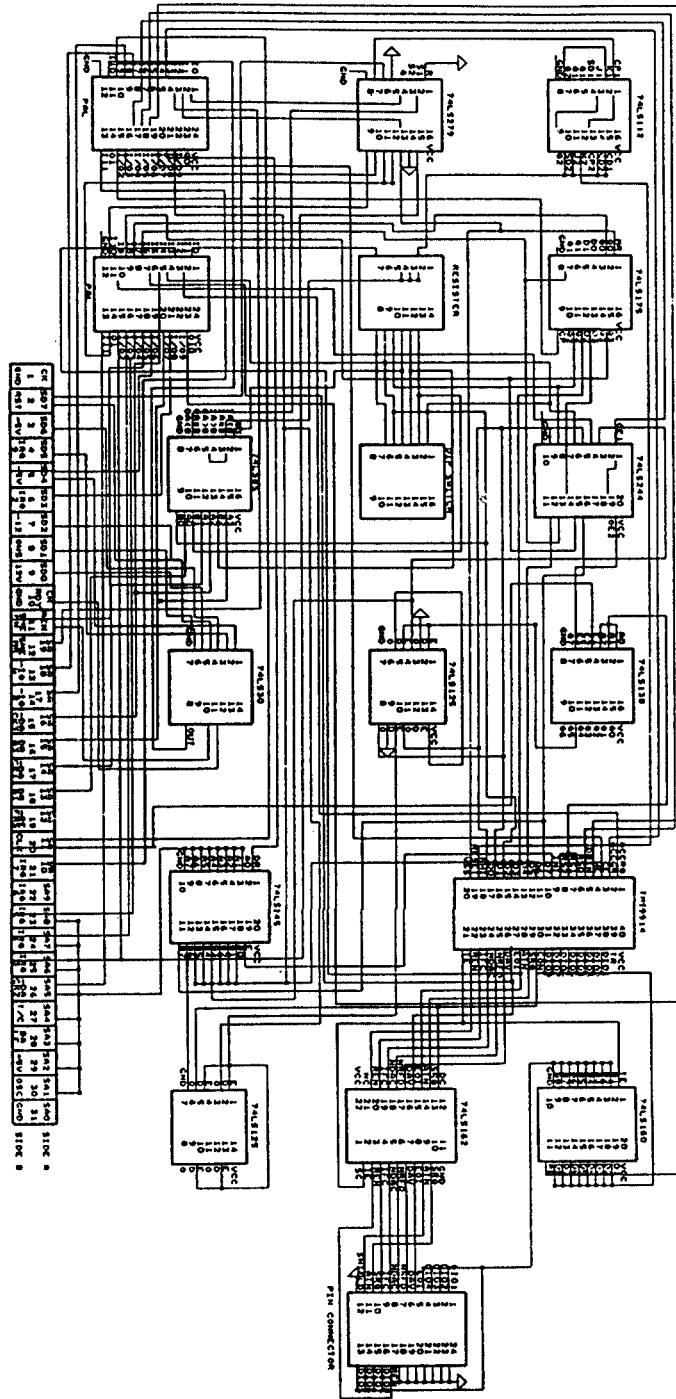


그림 16. 접속 장치 회로
Figure 16. circuit diagram of connection device

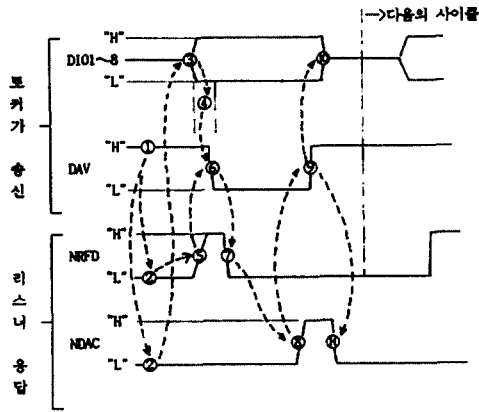


그림 17. 정보전송 타이밍도
Figure 17. Timing diagram of message transmission

을 측정하고 전력을 측정할 경우 주파수 정보를 없애고 전력을 측정할 수 있도록 구성되어 있다. 본 논문에서는 이러한 점을 보완하고 계측기 사이의 상호 데이터를 교환하기 위하여 다중프로그램을 수행할 수 있도록 하였다. 다중프로그램을 수행시킴으로써 사용자는 메시지 손실없이 다른 계측상태를 관측할 수 있는 장점을 가질 수 있다.

그림 18은 다중 프로그램을 실행시킬 때 나타날 수 있는 화면상태를 보이고 있다.

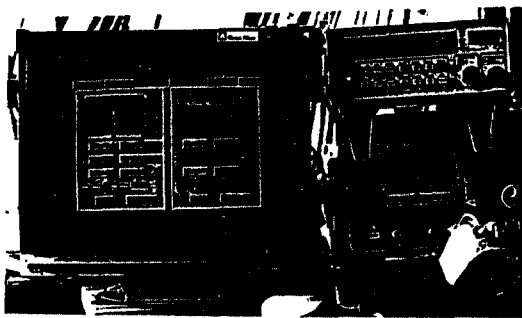


그림 18. 다중 프로그램을 수행시킨 화면상태
Figure 18. CRT display implemented multitask

그림에서 알 수 있듯이 각각 디바이스 명령을 전송한 결과 아무 이상없이 데이터를 전송함을 알 수 있다. 물론, 각 디바이스에 따라 전송하는 속도가 차이

가 날 수 있으나 계기를 측정하는 데는 별다른 지장을 주지 않는다. 또한, 본 논문에서 측정된 계측기는 두 대로 제한했으나 더 많은 계측기를 연결하여 사용해도 다중처리를 할 때의 처리속도가 늦어질 뿐 다른 것에는 관계가 없다.

3.2. 이론성과 실험 결과

라운드 로빈방식에 있어서는 각 작업간에 작업을 교환하기 위해서 각 작업에 시간을 할당하여야 한다. 본 논문에서는 각 작업에 할당되는 시간을 똑같이 획득하도록 분배하였다. 본 논문에서 구현한 다중처리는 시간 분할 형태로 구현하였기 때문에 작업 A는 작업 B가 작업을 멈출 때를 기다릴 필요없이 작업을 실행할 수 있다. 이렇게 함으로써 긴 작업을 수행할 때 작업을 완전하게 수행 못하고 다른 작업으로 교환되는 경우가 발생하나, 각 작업에 할당된 시간이 매우 작기 때문에 사용자는 계속해서 작업을 수행할 수 있다. 그러나, 동시에 처리할 작업이 어느 정도(8대 이상) 증가하면 작업을 처리하지 못하는 단점이 있다. 다음 표 2는 동시에 처리할 작업의 수가 증가할 때의 처리속도를 나타내고 있다.

표 2. 작업수에 따른 작업 처리 속도

Table 2. Processing speed of task

동시에 처리할 작업의 수	처리속도(단위 Kbps)
2대	500
4대	250
8대	175
10대	50
15대	28

표2에서 알 수 있듯이 이론적으로 15대까지 가능하나 실험치로는 8대 정도 까지가 적당하다. 그 이상은 작업하나가 한번 실행 후 다시 작업을 수행하기 위해서는 기다리는 시간이 많기 때문에 데이터가 손실될 우려가 있다.

라운드 로빈에서 할당시간의 크기는 시스템에 절대적인 영향을 끼친다. 할당시간의 크기를 크게 하면 프로세스의 처리시간이 많아짐으로 데이터를 손실할 우려는 작아지나 오버헤드가 증가하는 단점이 있다. 그림 19는 할당시간에 따른 문맥 교환의 횟수를 이론적인 예로 들었다.

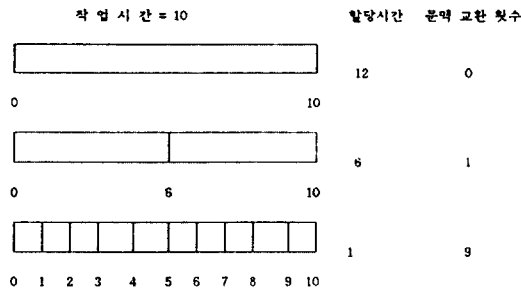


그림 19. 문맥 교환 횟수
Figure 19. Context switching number

본 논문에서는 할당시간의 크기를 절대적인 분할로 배분하였다. 즉, 2대일 경우는 할당시간이 1/2로 감소된다. 이렇게 함으로써 오버헤드는 감소하나 데이터 양이 많아지면 데이터를 손실할 우려가 많아진다. 그러나, 표 2에서도 보였지만 실제 계측기의 처리량이 작으므로 8대 까지는 데이터를 손실없이 처리할 수 있다.

VI. 결 론

최근 컴퓨터의 사용의 확산과 기능의 향상은 여러 분야에 커다란 파급효과를 가져왔고 이에 따라 새로운 개념들을 도입하기에 이르렀다.

결과적으로 90년대 세계 계측기산업은 단일 계측장비의 기능향상을 위한 노력보다는 컴퓨터를 이용한 계측장비의 시스템화, 통합화, 표준화에 주력하고 있다. 즉, 컴퓨터가 모든 계측장비를 제어하도록 시스템화되면서 공통규격을 갖는 표준화된 계측장비가 90년대 계측장비의 핵심분야로 대두되고 있다.

그 중에도 IEEE에 의해 표준화된 CAT기능은 컴퓨터를 부가한 계측시스템을 구성하고 각종 계측장비를 연결, 통합계측을 가능케 하고 있으며, 컴퓨터가 갖는 고속처리 능력을 이용하여 검사시간을 감소시킨 만큼 생산성의 향상 등에 크게 기여하고 있다.

이와같은 목적으로, 본 논문에서는 계측기와 컴퓨터를 연결하기 위해 GP-IB 카드를 제작하여 컴퓨터내에 장착하였으며, 시스템을 운용하기 위한 소프트웨어를 개발하였다.

또한, 다중처리를 위한 프로세스 기법은 트레드 배이스 방식을 사용하였고 스케줄러로는 라운드 로빈

방식으로 각 작업에 유효시간을 할당할 수 있도록 하였고 동시에 여러대의 계측기를 사용하기 위하여 다중처리 프로그램을 구현하였다. 시스템 운용 환경은 일반적으로 많이 쓰이고 있는 DOS를 사용하였다.

본 논문에서는 다중처리 시스템을 구현하기 위하여 HP8590A 스펙트럼 분석기와 HP437B 전력계를 시스템 운용 계측기로 이용하였다. 이 외에도 15대까지 계측기를 첨가하여 시스템을 운용할 수 있도록 할 수 있다. 개발된 소프트웨어는 총 9개의 모듈로 구성되어 있고 각 모듈들은 상호 공유되도록 구현되었다. 더 많은 계측기를 부가하려면 목적 계측기에 알맞은 소프트웨어 모듈을 합하여 구성할 수 있도록 하였다.

또한, 계측기를 제어하여 정보 전송 프로토콜의 패턴을 찾을 수 있었으며 각 기기의 디바이스 명령을 전송한 결과 이상없이 정보를 전송함을 알 수 있었다.

본 논문을 통해 개발된 시스템은 시간 절약과 사용자의 잘못으로 인한 오류를 방지할 수 있으며 또한 테스트 검증에 걸리는 시간의 절감효과를 가져올 수 있고 원하는 개발시스템을 보다 더 빠르게 구현할 수 있다. 또한 사용자가 이 운용 소프트웨어를 사용함으로써 전문가적인 지식없이도 쉽게 계측기를 제어할 수 있을 것이다.

이와같은 연구를 토대로 앞으로는 계측기를 모듈화하여 하나의 큰 랙(RACK)에 슬롯의 형태로 끼울 수 있도록 하여 모듈화된 디바이스 사이를 연결하는 VXI버스와 연계하는 시스템을 고려해 보면 더욱 향상된 시스템을 구현할 수 있으리라 사료된다.

참 고 문 헌

1. IEEE Standard Digital Interface for Programmable Instrumentation, IEEE Std New York, 1978.
2. Narbert Laengrich, "Assuming Measurement Accuracy in IEEE488 Based ATE System," Test & Measurement World, pp.102~135, May 1985.
3. 김종식, CAT시스템 개념과 도입, 전자과학, 1990. 3
4. VIXbus Consortium, VIXbus System Specification Revision 1.3, Jul. 1989.

5. 김영민, 사용자 인터페이스의 세 물결 GUI, PC 어드벤스, 교학사, October 1990.
6. David A Howorth, An Architecture for Modular Instruments, Textronix, 1988.
7. P. Anderson, GPIB Controlled electrical engineering Lab, Frontiers in Education conference, 1989.
8. 김용득, IBM-PC용 GPIB 접속장치에 관한 연구, 한국전자통신연구소 연구보고서, 1989. 2
9. 편집부편, IEEE 488 표준 디지털 버스-계측 제어용 표준버스의 기초에서 설계까지-, 가남사, 1989. 4
10. 트라ソツスタ 技術, GPIB Interface, pp.349~359, Jan.1984.
11. ASIC Lab., Designed for HP-IB systems, Hewlett Packard Corp., 1990.
12. The General-Purpose Interface Bus, IEEE MICRO Electronics, Feb. 1982.
13. 전자과학, GP-IB 콘트롤러, 전파과학사, 1990. 5
14. Donald C.Loughry and Mark S.Allen, "IEEE Standard 488 and Microprocesor Synergism Software." Proceedings of the IEEE, Vol.66, pp.162~172, Feb. 1978.
15. Peter Coad and Edward Yourdon, Object-Oriented Analysis, Prentice Hall, pp.82~91, 1990.
16. Danforth, Scott and Tomlinson, Chris, "Type Theories and Object Programming." ACM Computing Surveys, pp.15~21, March 1988.
17. Peter Coad and Edward Yourdon, Object-Oriented Design, Prentice Hall, 1991.
18. Richard S. Wiener and Lewis J. Pinson, An Introduction to Object-Oriented programming and C++, Addison Wesley, 1988.
19. Smalltalk /V Tutorial and Handbook, Digital, Inc., 1986.
20. Edmund W.Faison, Jr., Graphical User Interface with Turbo C++, SAMS, 1991.
21. Ben Ezzel, Turbo C++ Programming An Object-Oriented Approach, Addison Wesley, 1990.
22. Al Stevens, Teach Yourself C++, Mis Press, 1990.
23. Joseph E.Muller, "Efficient instrument design using IEEE-488.2," IEEE Trans. Instrument. Meas., vol.39, No.1, pp.125~130, Feb.,1990.
24. John Novellina, New Bus Arbitration Extended VXIbus, Electronic Design, pp.87~100, Apr., 1989.
25. Michael Tisher, PC System Programming for Developers, Abacus, 1989.

《본 논문은 한국과학재단 연구과제임 90-01-00-14》

田 東 根(Dong Keun Jeon)

※제 17 권 제 1 호 참조

정회원

盧 承 煥(Soong Hwan Ro)

정회원

1962년 8월 19일생

1987년 8월 : 고려대학교 전자공학과 졸업(공학사)

1989년 8월 : 고려대학교 전자공학과 대학원 졸업(공학석사)

1989년 9월 ~ 현재 : 동대학원 박사과정 재학중

※주관심분야: 정보통신, CAT(Computer Aided Test)
등임



車 均 鉉(Kyun Hyon Tchah) 正會員

1939年 3月 26日生

1965年 : 서울大學校 工學士

1967年 : 美國 일리노이大學校 工學
碩士

1976年 : 서울大學校 工學博士

1977年 ~ 現在 : 高麗大學校 電子工
學科 教授