

## 공유메모리 프로토콜을 이용한 VXIbus 시스템 구현에 관한 연구

正會員 盧 承 煥\* 正會員 姜 敏 鎬\* 正會員 金 惠 鎮\*

### A Study on Implementation of a VXIbus System Using Shared Memory Protocol

Soong Hwan Ro\*, Min Ho Kang\*, Duck Jin Kim\* *Regular Members*

#### 要 約

기존의 계측기기는 기능에 따라 독립적으로 이루어져 사용자는 사용목적에 따라 각각의 계측기를 이용하여 원하는 계측 시스템을 구성하였다. 그러나 1980년대 후반 VXIbus는 다양한 계측장비를 각각 한장의 카드로 만들어 선택적으로 계측시스템을 구성하는것을 가능하게 하였다.

VXIbus의 기본적인 통신 프로토콜에는 word serial 프로토콜이 있다. 그러나 측정된 데이터의 양이 증가하게 되면 word serial 프로토콜로 인한 전체 시스템의 성능 저하를 가져오게 된다.

본 논문에서는 이러한 성능제한요소를 해결하기 위하여 공유(shared) 메모리 프로토콜을 제안하고, 기존의 word serial 프로토콜과 공유메모리 프로토콜을 GSPN(Generalized Stochastic Petri Net)를 이용하여 분석하였다. 분석한 결과 공유메모리 프로토콜이 word serial 프로토콜보다 성능이 우수함을 알 수 있었다.

또한 제안된 공유 메모리 구조를 갖는 VXIbus 시스템을 구현하였으며 사용된 계측기기로는 VXIbus 인터페이스 모듈과 VMEbus 전용 신호처리모듈로 구성된 FFT 분석 디바이스 그리고 신호발생 디바이스가 이용되었다.

FFT 분석실험 결과 최대 80KHz 입력신호에 대하여 정확하게 분석되었으며 이 결과는 기존의 FFT 분석기의 결과와 잘 일치하였고, 신호발생 실험에서는 200KHz에서 1.1GHz까지의 정현파 신호가 발생되었다.

#### ABSTRACT

Existing Instruments are composed independently according to their function and user constructed instrumentation system with those instruments.

But in the late 1980s VXIbus enables to construct instrumentation system with various modular type instruments.

\*高麗大學校 電子工學科  
Dept. of Electronic Eng., Korea University  
論文番號 : 93-135

For an VXIbus system with the word serial protocol, an increase of data size can degrade the system performance.

In this paper shared memory protocol is proposed to overcome performance degradation. The shared memory protocol is analyzed using the GSPN and compared with that of the word serial protocol. It is shown that the shared memory protocol has a better performance than the word serial protocol.

The VXIbus message based-system with the proposed shared memory protocol is constructed and experimented with signal generating device and FFT analyzing device.

Up to 80 KHz input signal the result of FFT analysis is accurate and that result is agree with that of conventional FFT analyzer. In signal generating experiment from 100 KHz to 1.1 GHz sine wave is generated.

## I. 서 론

90년대 세계 계측산업은 단일계측장비의 성능향상에 질주하기보다는 컴퓨터를 기본으로 한 시스템화, 통합화, 표준화로 급진전하고 있는 추세에 있다. 즉 컴퓨터가 모든 계측장비를 제어할 수 있도록 시스템화 되면서 공통된 규격을 갖는 표준화 계측장비가 미래 계측산업의 핵심분야로 떠오르고 있다.

VXIbus는 여러가지 계측기기들을 각각 한장씩의 카드로 만들어 선택적으로 계측시스템을 갖추게 한 새로운 아키텍처로서 계측기기를 한 곳에 통합화 하는데 최적의 환경을 제공할 수 있으며, 서로 다른 업체에서 개발한 계측카드라도 호환이 가능하게 되는 장점을 지니고 있다. 또한 VXIbus는 GPIB에 비하여 매우 빠른 전송속도를 가질 뿐만 아니라, GPIB가 인터페이스를 드라이브하는 전력과 케이블 길이에 의한 지연 때문에 15대 까지만 확장할 수 있으나 VXIbus는 최대 256대까지 계측기기를 확장할 수 있는 점등 많은 장점을 가지고 있다.<sup>[1][2]</sup>

VXIbus의 디바이스에는 register-based 디바이스와 message-based 디바이스가 있으며 message-based 디바이스의 기본적인 통신 프로토콜로는 word serial 프로토콜을 사용한다. 또한 커맨드를 주는 디바이스를 commander라고 하고 받는 디바이스를 servant라고 하며, word serial 프로토콜은 servant의 읽기 또는 쓰기 데이터 레지스터를 이용하며 전 이중 UART (Universal Asynchronous Receiver/Transmitter)의 일반적인 모델에 기본을 두고 있다.<sup>[3]</sup> 그러나 message-based 디바이스에서 명령(command)은 ASCII 형태로 전송이 되므로 명령의 길이가 증가하는 경우, 또는 계측 디바이스로 부터 측정된 데이터가 많아지는 경우 servant의 응답(response) 레지스터를 pol-

ling하기 위하여 버스요구, 버스중재등으로 인한 많은 시간이 소요되며, 이러한 통신 프로토콜로 인한 전체 시스템에 성능 저하를 가져온다.

그러므로 이러한 word serial 프로토콜의 성능제한 요소를 해결하기 위한 연구가 필요하며, 본 논문에서는 해결방안의 하나로써 공유 메모리 구조를 갖는 VXIbus 시스템을 구현하고자 한다.

공유 메모리 프로토콜은 현재 VXIbus 사양에서 message-based 디바이스간의 통신 프로토콜로 규정된 word serial 프로토콜보다 향상된 성능을 갖는 프로토콜로써 VXIbus 사양에서 제안은 되고 있으나 공유메모리의 구조 및 운용방법등 구체적인 사항은 아직 규정되어 있지는 않은 상태이다.<sup>[3]</sup>

## II. VXIbus 개요 및 프로토콜

### 1. VXIbus 모듈<sup>[4]</sup>

VXIbus에서는 VMEbus의 A, B 모듈크기(module size)에 더하여<sup>[5]</sup> C, D 모듈크기를 추가적으로 정의하였으며, D 모듈크기에서는 P3 커넥터를 포함한다.

### 2. VXIbus 서브시스템(subsystem)

VXIbus 시스템은 한개 또는 그 이상의 VXIbus 서브시스템을 포함하며 256개의 디바이스를 포함할 수 있다. VXIbus 서브시스템은 slot0라고 하는 중앙 제어모듈과 최대 12개의 모듈을 포함할 수 있다. P2와 P3 커넥터는 VXIbus 서브시스템내에서 정의되며, P1을 포함하여 다음과 같은 신호군으로 분류된다.

- VMEbus
- 클럭버스(clock bus)
- 스타버스(star bus)

- 트리거 버스(trigger bus)
- ECL 트리거 버스
- 로컬버스(local bus)
- Analog SUMBUS
- Module Identification Bus
- 전원분배버스
- 예비단자(Reserved Pins)

**3. VXIbus 시스템 아키텍처(Architecture)**

일반적으로 VXIbus 시스템에서 한개의 디바이스는 한개의 모듈을 형성하나, 한개의 디바이스가 큰 경우 2개이상의 모듈로도 구성가능하며, 한개의 모듈 내에 2개이상의 디바이스가 존재할 수도 있다. 한 시스템내에는 최대 256개의 디바이스가 포함될 수 있으며, 0에서 255 까지의 논리적 디바이스 번지(logical device address)로 할당된다. VXIbus 시스템에서 configuration 영역은 64K 바이트의 A16 번지내의 상위 16K 바이트에 위치하며, 각 디바이스는 총 64바이트의 영역을 할당 받는다.

**4. Word Serial 프로토콜**

VXIbus 디바이스는 크게 register based 디바이스와 message based 디바이스로 구분이 되며 register based 디바이스의 통신에는 규정된 프로토콜은 없다. 그러나 message based 디바이스에서는 기본적으로 word serial 프로토콜을 사용하게 되어 있다.

Word serial 프로토콜은 전 이중(full duplex) UART (Universal Asynchronous Receiver/Transmitter)의 일반화된 모델에 기초를 두고 있다. 데이터의 송·수신은 양방향 데이터 레지스터와 응답 레지스터를 이용한다. 데이터 레지스터에 write 하는것은 앞의 커맨드에서 데이터라고 정의하지 않은 경우에는 커맨드로 해석된다. 커맨드는 종속된 데이터를 포함하거나 연속되는 write에서 필요한 데이터를 전송할 수 있다. 그러나 커맨드/데이터의 송·수신 절차는 일반적으로 연속적으로 수행이 된다.

데이터의 전송은 응답 레지스터의 write 레지스터가 비어있는지, read 데이터 레지스터에 데이터가 있는지를 나타내는 bit에 의해서 조절된다. 데이터는 응답 레지스터에 Write Ready bit가 1로 세트 되었을 때에만 써질 수 있다. 데이터가 write 데이터 레지스터에 써지면, servant에 의하여 데이터가 받아들여질 때까지, Write Ready bit는 0 값을 유지한다. 반대로 응답 레지스터의 Read Ready bit가 1로 세트되

어 있는 경우에만 데이터 레지스터에 유효한 데이터가 있게된다. 데이터 레지스터의 데이터가 읽히지면 servant가 또 다른 데이터를 쓰기전까지 0 값을 유지한다.

대부분의 경우 servant가 데이터 레지스터에 데이터를 내는 경우는 커맨드에 대한 응답인 경우이다. servant로부터 데이터를 요구하는 커맨드를 내기전에 데이터 레지스터의 데이터를 읽어내는 것은 commander에 의존한다. 그러므로 servant가 출력버퍼 구조를 가질 필요가 없다. 그림 1에서는 VXIbus 통신 레지스터와 응답 레지스터의 비트 할당을 보여주고 있으며, 통신레지스터에서 프로토콜 레지스터는 디바이스가 제공하는 또 다른 통신수단을 알리기 위하여 사용되어진다.

Device Dependent Registers	3F <sub>16</sub>
VXIbus Reserved Registers	1F <sub>16</sub>
A32 Pointer	18 <sub>16</sub>
A24 Pointer	14 <sub>16</sub>
Data Low	0E <sub>16</sub>
Data High	0C <sub>16</sub>
Response/Data Extended	0A <sub>16</sub>
Protocol/Signal Register	08 <sub>16</sub>
Configuration Registers	00 <sub>16</sub>

응답 레지스터

Bit#	15	14	13	12	11	10	9	8	7	6 - 0
Contents	0	reserved	DIR	DIR	Errs	Read Ready	Write Ready	FHS Active	Locked	Device Dependent

DIR : Data In Ready  
FHS : Fast Hand Shake

그림 1. VXIbus 통신 레지스터 및 응답 레지스터의 비트 할당

Fig. 1. VXIbus Communication Register and Bit Allocation of Response Register

**5. Word serial 프로토콜의 문제점 및 대책**

Word serial 프로토콜은 기본적으로 제공하여야 하지만 데이터 량이 많아짐에 따라 전체 시스템의 성능을 저하 시킬 수 있는 다음과 같은 문제점이 있다.

**1) 버스 사용량의 증가**

commander에서 servant로 데이터를 전송하는 경우 응답 레지스터의 Write Ready 비트를 반드시 polling해야 하므로 잦은 버스요구를 하게되고 이로 인

하여 버스 사용량이 증가하게 되므로 다른 버스 마스터와의 버스충돌이 자주 발생한다. 또한 commander에서 servant로부터 데이터를 읽어오는 경우에도 응답 레지스터의 Read Ready 비트를 polling해야 하므로 같은 효과로 인하여 버스충돌이 자주 발생한다.

### 2) 데이터 버퍼의 제한

Serial 방식의 전송은 데이터 레지스터를 통하게 되므로 1개의 데이터를 보낸 후 다음 데이터를 보내기 위해서는 받는 프로세서에서 데이터를 읽어간 후 이어야 한다. 그러므로 두 프로세서간 버퍼는 1개의 레지스터만이 존재하므로 많은 송·수신 대기시간이 필요하다.

위와 같은 문제를 해결하기 위해서는 공유메모리 프로토콜과 같이 향상된 성능의 프로토콜이 필요하며, 응답 레지스터를 polling하는 동작을 하지 않으므로 버스의 사용량이 감소하고 공유메모리가 데이터 버퍼 역할을 하므로 두 프로세서의 송·수신 대기시간을 감소시킬 수 있다.

## III. Word Serial 프로토콜과 공유메모리 프로토콜의 성능비교

본 장에서는 VXIbus 사양에서 규정하고 있는 word serial 프로토콜의 데이터 전송방식과 공유메모리 프로토콜의 데이터 전송방식에 대하여 성능을 분석하여 비교하였다. 본 연구에서는 성능분석의 방법으로 해석적(analytic) 방법의 하나인 GSPN(Generalized Stochastic Petri Net)를 이용하였다.

### 1. GSPN(Generalized Stochastic Petri Net)<sup>[9][10]</sup>

PN(Petri Net)는 동시성(concurrency)과 동기(synchronization)등을 표현하기에 적절하므로 컴퓨터 시스템을 모델링 하는데 널리 사용된다. PN 모델은 시스템에 있어서 deadlock과 같은 현상을 찾아내는데 이용되며, 시간이 고려되지 않은 성능 측정에도 이용될 수 있다.<sup>[6]</sup>

SPN(Stochastic Petri Net)은 각 천이에 지수분포함수를 갖는 점화시간을 부여한 것으로, M. K. Molloy에 의하여 제안되었다.

SPN은 다음과 같이 표현된다.

$$SPN = (P, T, A, M', R)$$

$$P = \{p_1, p_2, \dots, p_n\}$$

$$T = \{t_1, t_2, \dots, t_n\}$$

$$A \subseteq \{P \times T\} \cup \{T \times P\}$$

$$M' = \{m'_1, m'_2, \dots, m'_n\}$$

$$R = \{r_1, r_2, \dots, r_n\}$$

Molloy는 SPN이 지수분포의 점화시간이 memoryless 특성때문에 연속시간(continuous time) MC(Markov Chain)과 동일(isomorphic)하다는 것을 증명하였으며, 특히 k-bounded PN은 유한 MC과 동일하고, SPN의 marking이 MC의 상태와 같다는 것을 증명하였다.<sup>[7][8]</sup>

본 논문에서 사용한 GSPN은 M. A. Marsan 등에 의하여 제안되었으며, 지연(timed)천이와 순간(immediated)천이로 구분 되어지고, 지수분포함수의 천이시간은 지연 천이에 의해서만 정의된다. GSPN 또한 MC과 같으며 SPN 보다 쉽게 풀이를 얻을 수 있다는 장점이 있어 멀티 프로세서 시스템의 모델링과 성능평가에 많이 이용된다.<sup>[10]</sup>

### 2. GSPN에 의한 word serial 프로토콜의 모델링

본 논문에서 모델화한 serial 프로토콜은 크게 commander에서 servant로 데이터를 전송하는 경우와 servant에서 commander로 전송하는 경우로 들 수 있으며, 모든 경우에 servant의 데이터 레지스터를 사용한다.

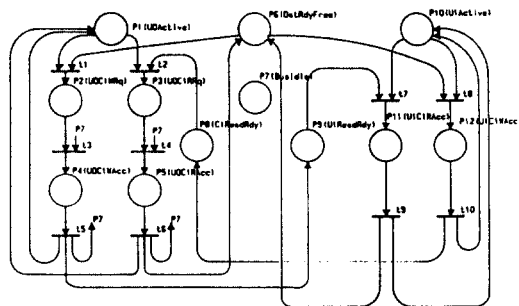


그림 2. Word Serial 프로토콜의 모델링  
Fig. 2. Modeling of Word Serial Protocol

그림 2는 일반적인 word serial 프로토콜의 데이터 전송을 모델링 한 것으로 commander로 나타내어지는 U0와 servant로 나타내어지는 U1의 2개 프로세서로 구분되어진다.

모델의 초기상태는 U0와 U1이 각각 로컬 메모리를 액세스하고 있는 액티브(active) 상태이고, 데이터 레지스터가 비어있으며 버스가 사용 가능하다.

먼저 U0에서 U1으로 데이터를 전송하는 경우 데이터 레지스터가 비어있는 상태를 감지하면 버스 요구(request)를 하고 데이터 레지스터 액세스 시간후에 다시 액티브 상태로 되돌아 온다. 이때 U1은 데이터 레지스터를 액세스하여 데이터를 읽어온 후 다시 액티브 상태로 돌아온다. 반대로 U1에서 U0로 데이터를 전송하는 경우도 같은 절차를 따르나 U1에서 데이터 레지스터를 액세스할 때는 버스요구를 하지 않고 자신의 on-board의 레지스터를 액세스하게 된다.

위의 모델에는 모두 12개의 표지소 및 10개의 천이 가 있으며, 순간(immediate) 천이는 t3, t4가 존재하며, 이 시간은 버스중재 및 해제에 소요되는 시간으로 다른 시간에 비하여 매우 짧다고 가정하여 순간 천이로 처리하였다.

표 1. 표지소 및 내용

Table 1. Places and Contents

표지소	내	용
p1	u0 Active	
p2	u0가 데이터 레지스터 쓰기	
p3	u0가 데이터 레지스터 읽기	
p4	u0가 데이터 레지스터 쓰기	
p5	u0가 데이터 레지스터 읽기	
p6	데이터 레지스터 쓰기 준비	
p7	Bus Idle	
p8	u0 데이터 레지스터 읽기 준비	
p9	u1 데이터 레지스터 읽기 준비	
p10	u1 Active	
p11	u1이 데이터 레지스터를 읽기를 위하여 액세스중	
p12	u1이 데이터 레지스터를 쓰기를 위하여 액세스중	

표 2. 천이 및 내용

Table 2. Transitions and Contents

천이	내	용
t1	u0 커맨드 처리	
t2	u0 데이터 처리	
t3	u0가 버스를 열고, 데이터 레지스터 쓰기 시작	
t4	u0가 버스를 열고, 데이터 레지스터 읽기 시작	
t5	u0가 데이터 레지스터 쓰기 마침	
t6	u0가 데이터 레지스터 읽기 마침	
t7	u1 커맨드 처리	
t8	u1 데이터 처리	
t9	u1 데이터 레지스터 읽기 마침	
t10	u1 데이터 레지스터 쓰기 마침	
지연천이	t1 t2 t5 t6 t7 t8 t9 t10	
순간천이	t3 t4	

표 3은 가능한 천이를 점화시켰을 때 토르의 움직임에 따라 발생하는 상태의 집합을 나타낸 것이다.

상태는 지연천이만을 가능(enable) 시키는 tangible 상태와 순간천이를 가능시키는 vanishing 상태로 나뉘어진다.

표 3. 도달가능 상태의 집합

Table 3. Reachability Set

Tangible State

place state	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
St <sub>1</sub>	1					1	1			1		
St <sub>2</sub>	1						1					1
St <sub>3</sub>	1						1	1		1		
St <sub>4</sub>				1						1		
St <sub>5</sub>					1					1		
St <sub>6</sub>	1						1		1	1		
St <sub>7</sub>	1						1					1

Vanishing State

place state	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
Sv <sub>1</sub>		1					1			1		
Sv <sub>2</sub>			1				1			1		

### 3. GSPN에 의한 공유 메모리 프로토콜의 모델링

공유 메모리 프로토콜에서도 commander를 나타내는 U0와 servant를 나타내는 U1으로 구성되며, U0에서 U1으로 데이터를 전송하는 경우는 serial 프로토콜과 마찬가지로 servant의 데이터 레지스터를 이용하며, U1에서 U0로 데이터를 전송하는 경우는 commander의 공유 메모리를 이용한다.

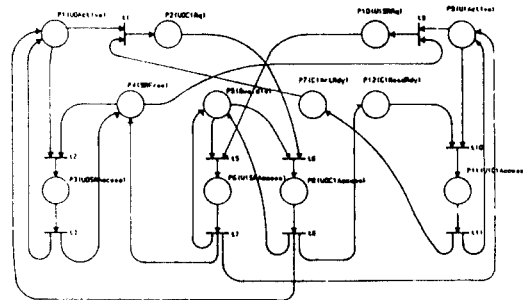


그림 3. 공유메모리 프로토콜의 모델링

Fig. 3. Modeling of Shared Memory Protocol

모델의 초기상태는 U0와 U1이 각각 로컬메모리를 액세스하고 있으며, servant의 데이터 레지스터가 비어있으며, 버스가 사용가능하고, commander의 공유 메모리가 액세스 가능하다.

U0에서 U1으로 데이터를 전송하는 경우는 word serial 프로토콜과 같은 방법으로 데이터를 전송하며, U1에서 U0로 데이터를 전송하는 경우는 word serial 프로토콜에서 데이터 ready flag를 이용하지 않고 공유 메모리를 U0가 액세스하지 않고 있으면 버스의 중재를 받아 액세스 할 수 있게 되므로 데이터 버퍼 역할을 하게된다.

표 3과 4는 각각 표지소 및 천이의 내용을 나타내고 있다. 위의 천이의 내용을 나타내는 테이블에서 순간천이는  $t_5, t_6$ 가 있으며 이는 버스의 중재와 해제에 걸리는 시간이 매우 짧다고 가정하여 순간천이로 처리하였다.

위의 테이블은 가능한 천이를 점화시켰을 때 토큰의 움직임에 따라 발생하는 상태의 집합을 나타낸 것으로서 모두 9개의 tangible 상태와 2개의 vanishing 상태가 존재한다.

표 4. 표지소 및 내용

Table 4. Places and Contents

표지소	내	용
p1	u0 Active	
p2	u0가 데이터 레지스터 쓰기를 위하여 버스를 기다림	
p3	u0가 Shared 메모리를 읽기 위하여 액세스 중	
p4	Shared 메모리 Available	
p5	Bus Idle	
p6	u1이 Shared 메모리 쓰기	
p7	u0 데이터 레지스터 쓰기 준비	
p8	u0가 데이터 레지스터 쓰기	
p9	u1 Active	
p10	u1이 Shared 메모리 쓰기를 위하여 버스를 기다림	
p11	u1이 데이터 레지스터 읽기	
p12	u1이 데이터 레지스터 읽기 준비	

표 5. 천이 및 내용

Table 5. Transition and Contents

천 이	내	용
t1	u0 커맨드 처리	
t2	u0 데이터 처리	
t3	u0가 Shared 메모리 읽기 마침	
t5	u1이 버스를 얻어, Shared 메모리 쓰기 시작	
t6	u0가 버스를 얻어, 데이터 레지스터 쓰기 시작	
t7	u1이 Shared 메모리 쓰기를 마침	
t8	u0가 데이터 레지스터 쓰기	
t9	u1이 데이터 처리	
t10	u1이 커맨드 처리	
t11	u1이 데이터 레지스터 읽기 마침	

표 6. 도달가능 상태의 집합

Table 6. Reachability Set

Tangible State

place state	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>	P <sub>9</sub>	P <sub>10</sub>	P <sub>11</sub>	P <sub>12</sub>
St <sub>1</sub>	1			1	1		1		1			
St <sub>2</sub>			1		1		1		1			
St <sub>3</sub>	1						1	1				
St <sub>4</sub>				1					1	1		
St <sub>5</sub>		1					1					
St <sub>6</sub>	1			1	1				1			1
St <sub>7</sub>			1		1				1			1
St <sub>8</sub>	1			1	1							1
St <sub>9</sub>			1		1							1

Vanishing State

place state	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>	P <sub>9</sub>	P <sub>10</sub>	P <sub>11</sub>	P <sub>12</sub>
Sv <sub>1</sub>	1				1		1			1		
Sv <sub>2</sub>		1		1	1				1			

#### 4. 성능평가 및 비교

##### 1) 상태천이행렬

위의 모델에서 U0에서 U1으로 보내는 데이터를 커맨드, U1에서 U0로 보내는 데이터를 계측기기에서 처리된 데이터라고 하면 각 천이와 관련된 천이의 점화율은 다음과 같다.

- U0, U1의 커맨드 처리율은 각각 동일( $\alpha$ )
- U0, U1의 데이터 처리율은 각각 동일( $\beta$ )
- U0의 데이터 레지스터 액세스시간과 U1의 공유 메모리 액세스시간은 동일( $1/\mu$ )
- U0의 공유 메모리 액세스시간과 U1의 데이터 레지스터 액세스시간은 동일( $\mu/(11/5)$ )

위의 천이시간은 모두 지수분포의 함수를 갖는다고 가정하였으며, U0의 공유 메모리 액세스 시간과 U1의 데이터 레지스터 액세스 시간은 4장에서 VXIbus에서 버스를 통과하여 메모리를 액세스하는 시간과 로컬메모리를 액세스하는 시간을 측정하여 정한 값이다.

표 7은 앞에서 정의한 천이율 및 시간에 의하여 구해진 상태천이행렬을 나타내고 있다. 이때 vanishing 상태는 지연시간 없이 tangible 상태로 천이가 되므로 상태에서 상태로의 이전은 tangible 상태에 한해

표 7. 상태전이 행렬

Table 7. State Transition Matrix

Shared Memory	Word Serial
상태 → 상태 : 점화율	상태 → 상태 : 점화율
St <sub>1</sub> → St <sub>1</sub> : α	St <sub>1</sub> → St <sub>1</sub> : α
St <sub>1</sub> → St <sub>2</sub> : β	St <sub>1</sub> → St <sub>2</sub> : β
St <sub>1</sub> → St <sub>3</sub> : β	St <sub>2</sub> → St <sub>3</sub> : (11/5)μ
St <sub>2</sub> → St <sub>1</sub> : (11/5)μ	St <sub>1</sub> → St <sub>5</sub> : β
St <sub>3</sub> → St <sub>5</sub> : α	St <sub>4</sub> → St <sub>6</sub> : μ
St <sub>3</sub> → St <sub>1</sub> : μ	St <sub>5</sub> → St <sub>1</sub> : μ
St <sub>4</sub> → St <sub>6</sub> : μ	St <sub>6</sub> → St <sub>7</sub> : α
St <sub>5</sub> → St <sub>4</sub> : μ	St <sub>7</sub> → St <sub>1</sub> : (11/5)μ
St <sub>6</sub> → St <sub>7</sub> : β	
St <sub>6</sub> → St <sub>8</sub> : α	
St <sub>7</sub> → St <sub>6</sub> : (11/5)μ	
St <sub>7</sub> → St <sub>9</sub> : α	
St <sub>8</sub> → St <sub>9</sub> : β	
St <sub>8</sub> → St <sub>1</sub> : (11/5)μ	
St <sub>9</sub> → St <sub>8</sub> : (11/5)μ	
St <sub>9</sub> → St <sub>2</sub> : (11/5)μ	

서만 해당된다.

앞에서 주어진 천이시간을 이용하여 입력 파라미터는 다음과 같이 정의하였다.

$$\begin{aligned} \text{평균 로컬메모리 액세스시간} &= \left(\frac{1}{\lambda}\right) \\ &= \frac{1}{\text{커멘드처리율} + \text{데이터처리율}} = \frac{1}{\alpha + \beta} \quad (1) \end{aligned}$$

$$\text{시스템 부하율}(\rho) = \frac{\text{비스엑세스 시간}}{\text{로컬엑세스 시간}} = \frac{1/\mu}{1/(\alpha + \beta)} \quad (2)$$

$$\gamma = \frac{\text{데이터 처리시간}}{\text{커멘드 처리시간}} = \frac{1/\beta}{1/\alpha} \quad (3)$$

위의 식 (1), (2), (3)에 의하여 α와 μ값을 아래와 같이 나타낼 수 있다. 모든 천이율은 γ, ρ와 β의 식으로 표현 가능하며, β는 안정상태확률을 구하는 과정에서 소개된다.

$$\therefore \alpha = \gamma\beta, \quad \mu = \frac{(1 + \gamma)\beta}{\rho} \quad (4)$$

2) 안정상태확률의 풀이

Molloy에 의하여 k-bounded SPN이 유한 MC와 동일하다는 것을 증명하였으므로, SPN이 주어지면 MC와 관련하는 다음과 같은 규칙이 발생한다.

가. MC의 상태공간 S는 SPN의 도달가능상태집합

R(M')와 동일하다.

나. 상태 i (marking M<sub>i</sub>)로부터 상태 j (marking M<sub>j</sub>)로의 천이율은 다음과 같다.

$$q_{ij} = \sum_{k \in H_{ij}} l_k \quad (5)$$

위의 식 (5)에서 H<sub>ij</sub>는 marking M<sub>i</sub>에서 marking M<sub>j</sub>로 천이하는 모든 가능한 천이의 집합을 나타낸다.

앞의 규칙에 의하여 다음 식에 의하여 안정상태확률을 구할 수 있다.

$$\pi Q = 0 \quad (6)$$

$$\sum_i \pi_i = 1 \quad (7)$$

위의 식에서 π는 안정상태확률의 벡터이며, Q는 infinitesimal generator로서 요소는 위의 식 (5)에 의하여 주어진다.

앞의 풀이과정을 통하여 구해진 안정상태확률을 이용하여 다음과 같은 값을 구할 수 있다.

가. SPN의 특정조건 확률: R(M')의 부분집합 A가 다음 조건을 만족하면 다음과 같은 값에 의하여 구하여진다.

$$P\{A\} = \sum_{i \in A} \pi_i \quad (8)$$

나. 표지소의 평균 토큰 수: 표지소 p<sub>i</sub>의 토큰의 수가 x라하고 k-bounded를 만족하며, A(i, x)가 R(M')의 부분집합이면 표지소 p<sub>i</sub>에 있는 평균 토큰의 수는 다음에 의하여 구하여진다.

$$E\{m_i\} = \sum_{n=1}^k [nP\{A(i, n)\}] \quad (9)$$

다. 단위시간당 평균 점화수: A<sub>j</sub>가 R(M')의 부분 집합이고 t<sub>j</sub>가 점화가능하면 단위시간 t<sub>j</sub>의 평균 점화수는 다음 식에 의하여 구하여진다.

$$f_j = \sum_{M_i \in A_n} [\pi_i (l_j / \sum_{\forall l_k \text{ enabled in } M_i} l_k)] \quad (10)$$

3) 분석비교결과

본 논문에서 사용한 성능지표(performance index)는 각 프로세서의 단위시간당 데이터 처리율로서 다음식에 의하여 구하여진다.

$$\text{Throughput} = P\lambda \quad (11)$$

이때 P는 Processing Power로서 액티브상태에 있는 평균 프로세서의 수를 나타낸다. 그러므로 각 모델링에 대한 processing power는 각 프로세서가 액티브 상태에 있을 확률의 합과 같으며 다음과 같이 구할 수 있다.

word serial 프로토콜	$(P(1) \cdot P(3) + P(4)) \times 2 + P(2) + P(5) + P(6) + P(7)$
공유 메모리 프로토콜	$(P(1) \cdot P(6)) \times 2 + P(2) + P(3) + P(4) + P(7) + P(8)$

표 8과 그림 4에 시스템 부하율에 따른 데이터 처리율을 수치와 그래프로 각각 나타내었다.

그림 4의 b) 그래프에서 시스템 부하율이 0.7 이하에서는 공유 메모리 프로토콜이 성능이 우수하며, a) 그래프와 표 8에서 보듯이 시스템 부하율의 값이 약 0.01에서 word serial 프로토콜보다 20% 정도 성능 개선이 되는 것을 알 수 있다. 그러나 시스템 부하율이 0.7 이상으로 증가하는 경우 word serial 프로토콜의 성능이 우수한 것으로 나타난다. 그 이유는 word serial 프로토콜에서 한개의 데이터를 전송하기 위하여 상대 프로세서에서 데이터를 읽기 전에는 다른 데이터를 다시 전송할 수 없으나 공유 메모리 프로토콜에서는 메모리가 데이터 버퍼 역할을 하므로 처리속도가 개선되는 것을 알 수 있다.

그러나 시스템 부하율이 증가함에 따라 버스 사용률이 증가하고, serial 프로토콜의 경우 시스템내에서 commander만이 버스마스터가 되며, 공유 메모리 프로토콜에서는 commander, servant가 모두 버스마스터가 되므로 버스요구, 대기등으로 인한 성능 저하요소가 발생한다. 그러므로 시스템 부하율이 증가하게 되면 전체 시스템 처리율면에서 word serial 프로토콜의 성능이 우수한 것으로 나타난다.

그러나 실제의 시스템에서 시스템 부하율이 0.7 이상이 될 수 없으며, 본 모델링은 시스템의 운용방법 등을 고려하지 않은 성능분석이므로 실제시스템에서 운용방법까지 감안한다면 공유메모리 프로토콜이 serial 프로토콜보다 성능이 우수하다고 볼 수 있다.

표 8. 데이터 처리율

Table 8. Throughput

부하율	프로토콜	공유메모리	Word Serial
0		0	0
0.01		18,115	15,149
0.02		36,102	30,305
0.03		53,961	45,460
0.04		71,695	60,615
0.05		89,308	75,772
0.06		106,801	90,930
0.07		124,177	106,089
0.08		141,439	121,250
0.09		158,589	136,411
0.1		175,628	151,574
0.2		340,429	303,255
0.3		496,328	455,031
0.4		644,791	606,889
0.5		787,023	758,819
0.6		923,959	910,815
0.7		1,056,359	1,062,868
0.8		1,184,847	1,214,972
0.9		1,309,936	1,367,123
1		1,432,053	1,519,315

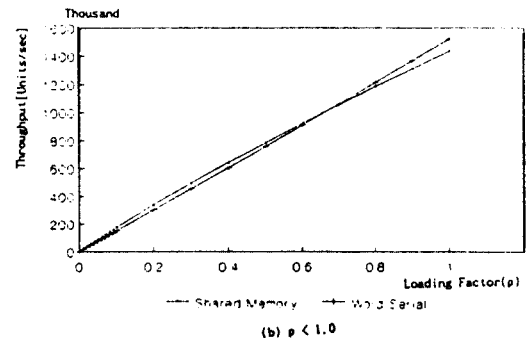
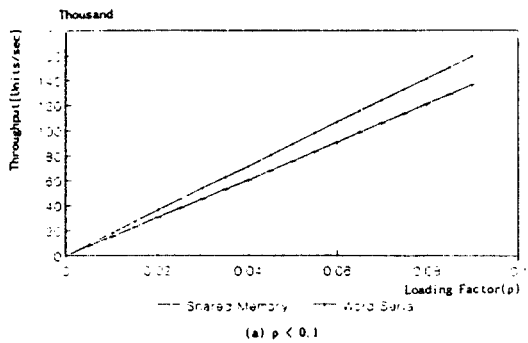


그림 4. 데이터 처리율

Fig. 4. Throughput



#### IV. 공유메모리 구조를 갖는 Slot0 Resource Manager의 구현

본 논문에서 구현한 slot0 resource manager의 프로세서는 VMEbus와 접속이 용이하고 32 bit의 어드레스와 데이터 버스를 제공하는 모토롤라 계열 MC68030 MPU를 사용하였다. 구현된 slot0 모듈은 논리적으로는 VMEbus 마스터 기능을 가지며, 버스 리퀘스터(requestor), 버스 중재기(arbiter), 인터럽트 핸들러(interrupt handler)등의 기능을 갖는다. 또한 on-board 형태의 1 Mbyte 메모리를 구현하였으며 그림 5에 블록도를 나타내었다.

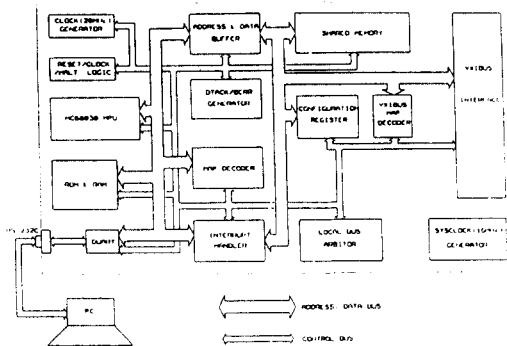


그림 5. 구현된 Slot0 Resource Manager의 블록도  
Fig. 5. Block Diagram of Slot0 Resource Manager

##### 1. 공유메모리의 구현

모듈에서 구현된 공유 메모리 프로토콜을 위하여 메모리 디바이스의 configuration 레지스터내에 카운터 레지스터를 구현하였다.

메모리 디바이스의 논리 어드레스를 10으로 하였으며 실제 어드레스는 다음 식에 의하여 \$ff02C280번지가 된다.

$$\text{Physical Address} = 10_{10} \times 64 + 49152$$

카운터 레지스터는 base 어드레스에서 0A<sub>16</sub> 번지에 하위 카운터 레지스터(counter low register), 0C<sub>16</sub> 번지에 상위 카운터 레지스터(counter high register) 각각 할당하였다.

카운터 레지스터는 8 bit octal up/down 카운터를 이용하여 구현되었다. 운용방법은 servant에서 수집·분석된 데이터의 word 수 또는 byte 수를 load 한 후 공유메모리로 데이터를 전송하게 된다. 이때 한번의

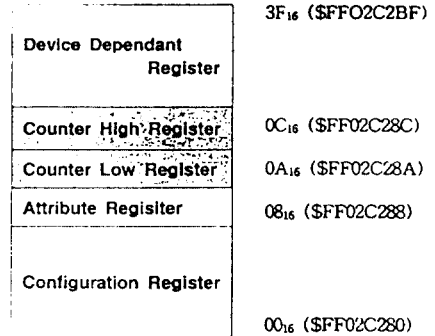


그림 6. 카운터 레지스터의 할당  
Fig. 6. Address Map of Counter Register

데이터 전송이 이루어 질 때마다 카운터는 down count를 하게 되고, count out이 되면 resource manager에게 인터럽트 요구를 발생시켜, resource manager는 인터럽트 처리부에서 공유메모리의 데이터를 처리하게 된다. 이때 발생하는 인터럽트는 autovector 처리를 하며 인터럽트 응답 사이클에서는 인터럽트 핸들러에 의해 발생하는 CNT\_JACK\* 신호에 의해 인터럽트 요구가 해제된다. 그림 7에 공유메모리 프로토콜의 흐름도를 나타내었다.

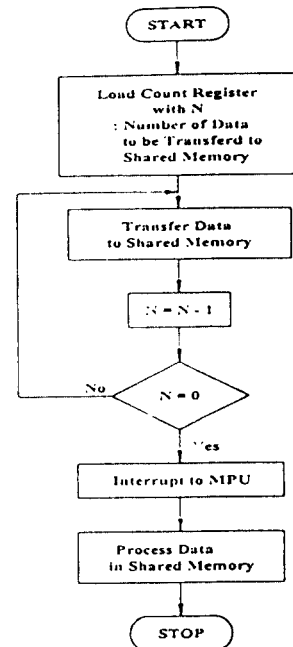


그림 7. 공유메모리 프로토콜의 흐름도  
Fig. 7. Flow Chart of Shared Memory Protocol

2. 공유메모리의 액세스

1) 로컬버스의 중재

모듈내에 버스중재는 SCB68172 BUSCON에서 제공되는 3개의 제어신호를 사용한다. 3 신호에 의하여 버스마스터에서 VXIbus를 액세스 할 때, 버스마스터에서 공유메모리를 액세스 할 때, VXIbus에서 공유메모리를 액세스 할 때 어드레스 버스, 데이터 버스, 콘트롤 버스가 각각 제어되며 표 9에 신호선의 값과 버스의 방향을 나타내었다.

표 9. 로컬버스 중재 신호선

Table 9. Local Bus Arbitration Signal

액세스	신호선	MASTENN	VMEENN	SLVSELN
마스터 → 공유메모리		L	H	L
마스터 → VXIbus		L	L	H
VXIbus → 공유메모리		H	L	L

2) 공유메모리 액세스

그림 8은 마스터에서 공유메모리를 액세스할 때와 VXIbus에서 공유메모리를 액세스할 때의 각각에 대한 실험결과 신호의 타이밍을 나타낸다.

공유메모리를 액세스하는 두 가지의 경우에 액세스 시간은 약 500 nsec와 1,100 nsec로 약 2배의 차이를 보이며 이 값을 3장의 해석적 모델의 입력 파라미터로 사용하였다.

V. VIX FFT 분석기 및 신호발생기의 구성 및 실험

1. 실험 환경

본 논문에서는 구현된 slot0 resource manager를 이용하여 VXIbus 시스템을 구성하였다. 사용된 계측기기로는 VXIbus 인터페이스 모듈과 DSP(Digital Signal Process) 모듈을 이용하여 구성한 FFT(Fast Fourier Transform) 분석기모듈과 100KHz에서 최대 1.1GHz까지의 장현파를 발생시키는 신호발생기 모듈을 사용하였다.

그림 8에 구성된 시스템의 블럭도를 나타내었다. 구현된 시스템에서 FFT 분석기 모듈은 VMEbus 전용 신호처리 모듈과 VXIbus 인터페이스 기능을 갖는 제어모듈로 이루어 졌으며, DSP 모듈과 인터페이스 모듈과의 통신은 표준 VMEbus를 사용한다. VXIbus 프레임과 컨트롤러 역할을 하는 PC 사이통신은 RS-232C를 이용한다.

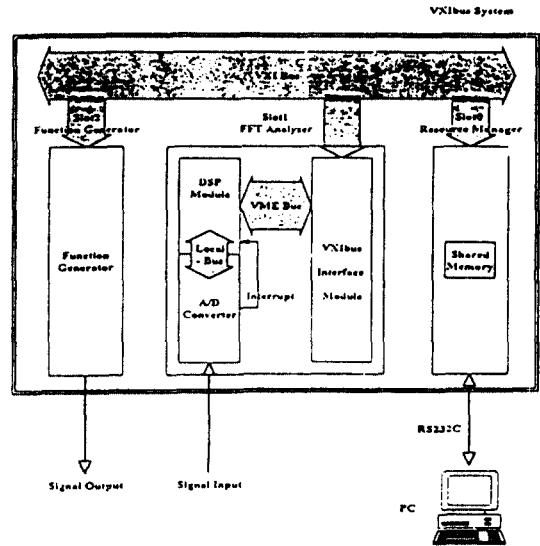


그림 8. 구현된 VXIbus 시스템의 블럭도

Fig. 8. Configuration of VXIbus System

표 10. 디바이스들의 어드레스 할당

Table 10. Address Map of Devices

디바이스	어드레스	논리 어드레스	실제 어드레스
Slot0 Resource Manager		0 <sub>10</sub>	\$\$FF02C000
Slot1 FFT 분석기 디바이스		1 <sub>10</sub>	\$\$FF02C040
Slot2 신호발생기 디바이스		2 <sub>10</sub>	\$\$FF020280
공유메모리 디바이스		10 <sub>10</sub>	\$\$FF02C280

표 10은 시스템에서 할당된 디바이스들의 논리 어드레스(logical address)와 실제 어드레스(physical address)를 보여준다.

2. FFT 분석기 디바이스

FFT 분석기 디바이스는 VXIbus 인터페이스 모듈과 VMEbus 전용 DSP 모듈로 구성되며, 인터페이스 모듈은 DSP 모듈의 제어역할을 담당하여 표준 VXIbus에서도 사용할 수 있도록 한다.

인터페이스 모듈은 전원이 인가된 후 자신의 메모리 초기화등을 행하고 DSP 모듈의 초기화를 실행한다. 또한 응답 레지스터의 write ready bit를 set하여 slot0 resource manager로부터 커멘트를 기다린다. Slot0 모듈로부터 커멘트를 받은 후, 커멘드가 RESET, INIT인가 MEAS인가에 따라 신호처리부 및 A/D 변



신호발생기는 다른 전자회로나 통신장비를 실험하고 조정하는데 사용되는 기준신호를 발생하는 계측기기로서 출력신호의 스프리어스가 적고 대역내에서의 직선성이 적고 출력주파수의 해상도 및 선택성이 우수할 것 등이 요구된다. 다음은 본 연구에서 사용한 신호발생기의 규격을 나타내고 있다.

- 주파수 범위 : 100kHz ~ 1.1GHz(정현파)
- 출력 레벨 : +10 ~ -100dBm
- 출력 임피던스 : 50 Ω
- 커맨드(SCPI 표준)

VXIbus 인터페이스부는 신호발생기로 명령어를 전송하는 역할을 담당하며 slot0 resource manager로부터 명령어가 신호발생기 모듈의 어드레스를 선택하면 인터페이스부에서는 신호발생기의 제어부로 인터럽트를 발생시켜 전송되는 명령 데이터를 읽게 된다. 이때 DTACK\*은 VXIbus에서 제공되는 16MHz 시스템 클럭을 64분주를 하여 약 8usec후에 발생시키며, 동시에 신호발생기의 제어부 인터럽트를 해제한다.

#### 4. 스펙트럼 분석실험

##### 가) 분석 방법

본 논문에서 수행한 스펙트럼 분석 방법은 A/D 변환부에서 샘플링된 데이터를 이용하여 radix-2 real FFT 알고리즘을 적용하였고 Hamming Window 함수를 사용하여 측정실험을 하였다. 또한 샘플링한 데이터를 처리할 때 non-overlap contiguous data segmentation을 적용하여 연속 측정시 유실되는 데이터가 없도록 하였다. 본 실험에서 사용한 데이터 포맷을 32 bit IEEE754 부동소수점을 사용하였으며, slot0를 통해 PC로 전송될 때 까지 그 값을 유지한다.

시스템에서 전원이 인가된 후로부터 스펙트럼 분석된 데이터를 컨트롤러로 디스플레이 하기까지의 절차를 다음에 나타내었다.

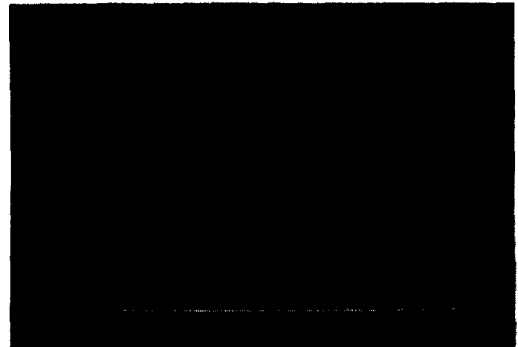
##### 나) 분석결과

앞의 절차에 의하여 정현파(sine wave) 및 구형파(square wave)에 대하여 반복 실험을 하였으며, 200 KHz 샘플링율에서 최대 100KHz 입력신호까지 FFT 분석실험을 실시하였다. 분석실험결과 샘플링율이 200KHz 이므로 최대 100KHz 입력신호까지는 가능하였지만 정현파 입력이 약 80KHz 일때까지의 분석

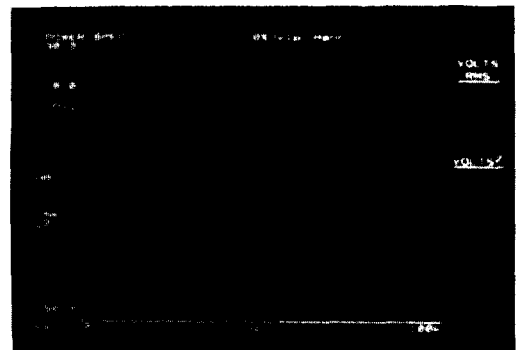
이 정확하게 됨을 보여주었다.

Point수는 32에서 최대 2048까지 증가시켜가면서 실험을 반복 수행하였으며 분석된 결과는 slot0 모듈의 RS232 케이블을 통하여 PC의 화면에 디스플레이를 하였다. 그러나 point수가 증가함에 따라 결과 데이터의 수가 많아지게 되므로 RS232를 통한 데이터 전송에 많은 시간이 소요되었다.

그림 10에서 11은 샘플링율 200KHz에서 point수 2048의 연산 파라미터로 하였을때 20KHz 입력신호의 정현파, 삼각파 및 구형파에 대한 결과 사진을 Hewlett-Packard 3562A Dynamic Signal Analyzer를 이용한 FFT 분석결과와 각각 비교하여 나타내었다. 표 12에는 입력주파수를 10KHz, 20KHz, 50KHz로 각각 주었을 때의 결과를 비교하였다. 표 12에는 입력주파수를 나타내는 기본주파수에 대하여 매우 정확한 값이 나타남을 알 수 있다.



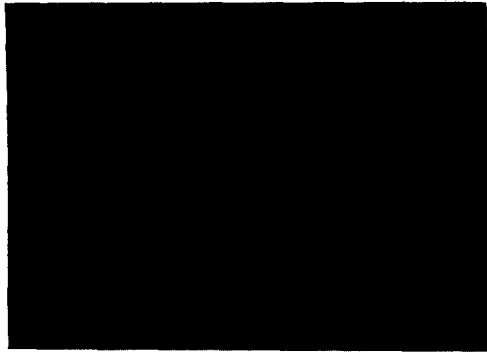
(a) VXIbus FFT Analyzer



(b) HP 3562A Dynamic Signal Analyzer

그림 10. 스펙트럼 분석 결과(정현파 20KHz)

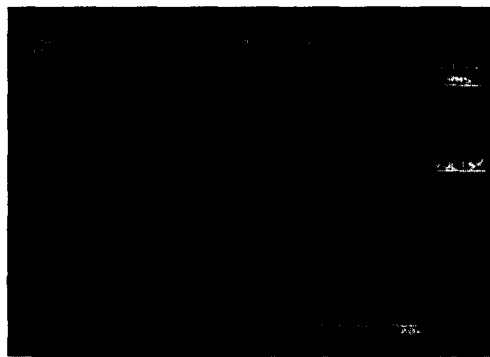
Fig. 10. Result of Spectrum Analysis(Sine Wave 20KHz)



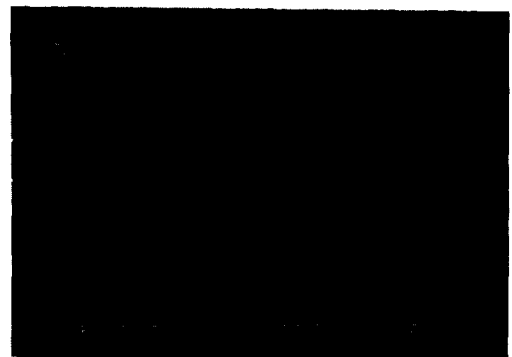
(a) VX1bus FFT Analyzer



(a) VX1bus FFT Analyzer



(b) HP 3562A Dynamic Signal Analyzer



(b) HP 3562A Dynamic Signal Analyzer

그림 11. 스펙트럼 분석 결과(삼각파 20KHz)  
Fig. 11. Result of Spectrum Analysis(Triangular Wave 20KHz)

그림 12. 스펙트럼 분석 결과(구형파 20KHz)  
Fig. 12. Result of Spectrum Analysis(Rectangular Wave 20KHz)

표 12. 스펙트럼분석 결과 비교

Table 12. Comparison of Spectrum Analysis

(단위 : dBV)

주파수(KHz)		0	10	30	50	70	90
정현파	A	-25	9				
	B	-29	9				
삼각파	A	-25	9	-20	-33	-37	
	B	-25	9	-18	-30	-34	
구형파	A	-25	11	-8	-11	-13	-15
	B	-25	11	-5	-8	-10	-13

(a) 10kHz (8.0 Vpp)

(단위 : dBV)

주파수(KHz)		0	20	60	100
정현파	A	-25	10		
	B	-29	10		
삼각파	A	-25	10	-20	-27
	B	-29	10	-17	-25
구형파	A	-25	11	-5	-12
	B	-25	11	0	-10

(b) 20kHz (8.0 Vpp)

(단위 : dBV)

주파수(KHz)		0	50
정현파	A	-25	10
	B	-29	10

(c) 50kHz (8.0 Vpp)

사진에서 주파수 SPAN을 0-100KHz까지로 하고 출력범위를 -40dBV에서 30dBV까지로 하였을 때의 결과로써 매우 정확하게 분석됨을 알 수 있으며, 80KHz 이상의 입력신호에서는 분석결과의 정확도가 떨어짐을 알 수 있었다. 그 이유는 Hewlett-Packard 3562 Dynamic Signal Analyzer의 샘플링율은 256 KHz이고 본 연구에서 구현된 FFT 분석 디바이스의 샘플링율은 200KHz 이므로 최대 입력주파수 값이 100KHz에 가까와 짐에 따라 샘플링율과 샘플링 과정에서 오는 차이때문인 것으로 분석된다.

### 5. 신호발생기의 실험

본 실험에서는 slot0 resource manager로부터 표 13과 같은 커멘드를 전송하였을 때 신호발생 디바이스로부터 발생하는 정현파를 측정하였다. 사용되는 커멘드는 CLEAR에서 현재 설정되어 있는 주파수와 레벨값이 해제 되며, 주파수를 설정하기 위하여는 FREQ 커멘드 뒤에 주파수 값을 갖는 파라미터가 오며 단위를 나타내는 KHz 또는 MHz가 뒤따라 온다. 신호 레벨값을 설정하기 위하여는 LEVEL 커멘드 뒤에 레벨값을 갖는 파라미터가 뒤따라 오고 다음에 단위로서 dBm 또는 mV가 온다. RF ON/OFF는 현재 설정된 값으로 신호를 발생하기 위한 커멘드로서 신호발생여부의 스위치 역할을 한다.

표 13. 신호발생 디바이스의 커멘드  
Table 13. Command of Function Generator

커 멘 드	내 용
CLEAR	현재 설정값을 해제
FREQ n KHz/MHz	발생 주파수 값을 설정
LEVEL n dBm/mV	출력 레벨 값을 설정
RF ON/OFF	주파수 발생 ON/OFF

위의 커멘드를 이용하여 실험한 결과 신호발생기의 출력주파수는 100KHz에서 1.1GHz까지 발생시킬 수 있었으며, 출력신호 레벨은 10에서 -100dBm까지 발생할 수 있었다. 또한 RF 필터를 통과한 후의 잡음 레벨이 -130dBm으로써 이러한 결과는 본 신호발생기 디바이스의 사양을 만족시키는 값이다.

## VI. 결 론

본 논문은 CAT의 일종인 VXIbus의 통신프로토콜

에 관한 연구이며, VXIbus의 기본적인 통신프로토콜인 word serial의 개선 프로토콜로서 공유메모리 프로토콜을 제안하였다.

Word serial 프로토콜의 성능저하요소는 UART (Universal Asynchronous Receiver/Transmitter) 방식의 데이터 전송으로 데이터의 양이 많아지는 경우 전체 시스템의 성능에 크게 영향을 미친다. 그러므로 본 연구에서는 slot0 resource manager 모듈내에 공유메모리구조를 갖는 시스템을 제안하고 이에 대한 성능 평가를 실시하였다. 그 결과 시스템 부하율이 0.7을 넘지 않는 경우에는 본 연구에서 제안한 공유메모리 프로토콜의 성능이 더 우수함을 알 수 있었으며, 최대로 시스템 부하율 0.01에서 약 20%의 성능향상을 보였다. 그러나 시스템 부하율이 0.7을 넘게되면 오히려 word serial 프로토콜의 성능이 우수함을 알 수 있었다. 그 이유는 공유메모리 프로토콜의 경우에 한 시스템내에 2개의 버스마스터가 존재하므로 부하율이 많아지면 잦은 버스충돌과 대기, 중재로 인한 성능저하요소가 발생하게 되지만, word serial 프로토콜의 경우에는 단일 버스마스터로 시스템 부하율이 증가하여도 버스충돌과 대기, 중재로 인한 성능저하요소는 발생하지 않는다. 그러나 실제의 시스템에서 시스템 부하율이 0.7이 넘지 않으며 본 논문에서 수행한 성능평가는 시스템 운용방법은 고려하지 않은 확률·통계적 해석을 위한 모델인 점을 감안하면, 실제의 시스템에서는 공유 메모리 프로토콜이 우수하다고 할 수 있다.

또한 on-board 형태의 공유메모리를 갖는 slot0 resource manager를 구현하였으며, 모듈내의 프로세서에서 공유메모리를 액세스할 때는 로컬버스를 이용하여 시스템 버스의 부하를 줄여 성능이 향상되도록 하였다.

구현된 slot0 모듈을 이용하여 VXIbus message-based 시스템을 구현하였으며 사용된 계측 디바이스는 FFT 분석 디바이스와 신호발생 디바이스를 이용하였다. 계측 디바이스로 사용한 FFT 분석 디바이스는 본 연구실에서 구현한 디바이스로 인터페이스 모듈과 VMEbus전용 DSP 모듈로 구성되어있으며, DSP 모듈은 A/D 변환부와 신호처리부로 구성되어 있었다.

실험은 샘플링율을 최대 200KHz까지 증가시켜가면서 아날로그 입력신호에 대한 스펙트럼분석을 실시하였다. 또한 point 수는 최대로 2048까지 증가시켜가면서 실험을 반복수행 한 결과 매우 정확하게 분

석이 됨을 알 수 있었다. 결과에서 주파수 SPAN을 0-100KHz까지로 하고 출력범위를 -40dBV에서 30 dBV까지로 하였을 때 매우 정확하게 분석됨을 알 수 있으며, 80KHz 이상의 입력신호에서는 분석결과의 정확도가 떨어짐을 알 수 있었다. 또한 신호발생실험에서는 출력주파수가 100KHz에서 1.1GHz까지 발생시킬 수 있었다. 출력신호 레벨은 10dBm에서 -100 dBm까지 발생할 수 있었다.

참 고 문 헌

1. Ron Wolfe, "Software communication protocol for VXI and GPIB : similar but different," VXI Journal, November 1990.
2. Larry Des Jardin, "VXI versus GPIB : is VXI actually faster?," VXI JOURNAL, pp.11-14, VOL. 1, NO.2, NOV. 1990.
3. VXIbus System Specification, Revision 1.3, VXI Consortium, July 14, 1989.
4. 김덕진, 노승환 외 3명, "Register Based VXIbus 시스템의 구현에 관한 연구," 한국통신학회논문지, 제17권, 제11호, 1992. 11.
5. "Applying the VXIbus Architecture," Hewlett-

- Packerd Company.
6. Peterson, J. L., "Petri net Theory and the Modeling of Systems," Prentice-Hall Englewood Cliffs, NJ 1981.
7. Michael K. Molloy, "Discrete Time Stochastic Petri Nets," IEEE Trans. on Software Engineering, Vol. SE-11, No.4, April 1985, pp.417-423.
8. Michael K. Molloy, "Performance Analysis Using Stochastic Petri Nets," IEEE Trans. on Comp., Vol. C-31, No.9, Sep. 1982, pp.913-917.
9. Marco Ajmone Marsan, Gianni Conte, Gianfranco Balbo, "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems," ACM Trans. on Computer Systems, Vol.2, May 1984, pp.93-122.
10. Marco Ajmone Marsan, Gianni Conte, Gianfranco Balbo, Performance Models of Multiprocessor Systems, The MIT Press
11. CAT(Computer Aided Test) 기술에 관한 연구, 한국과학재단 목적기초연구 제2차 중간보고서, 1992. 6.

본 논문은 한국과학재단 연구과제임 90-01-00-14



盧承煥(Soong Hwan Ro) 正會員  
 1962년 8월 19일생  
 1987년 8월 : 고려대학교 전자공학과 졸업(공학사)  
 1989년 8월 : 동대학원 졸업(공학석사)  
 1993년 8월 : 동대학원 졸업(공학박사)

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



姜敏鎬(Min Ho Kang) 正會員  
 1968년 7월 10일생  
 1991년 2월 : 고려대학교 전자·전산공학과 졸업(공학사)  
 1993년 8월 : 동대학원 졸업(공학석사)

현재: 현대전자 근무  
 ※주관심분야: 컴퓨터구조, 정보통신



金 惠 鎭(Duck Jin Kim) 正會員

1933년 12월 12일생

1957년 2월 : 서울대학교 전자공학과(공학사)

1962년 1월 : 일리노이 공과대학 전자공학과(공학석사)

1972년 2월 : 고려대학교 전자공학과 공학박사

1967년~1971년 : 서울대학교 전자공학과 공학박사

1971년~현재 : 고려대학교 전자공학과 교수

1985년 : 대한전자 공학회 회장

1990~현재 : 고려대학교 정보·통신 기술공동 연구소 소장