

論文97-34C-6-6

SPAX 병렬 컴퓨터에서의 온라인 무간섭 네트워크 성능 감시기

(An On-Line Non-Invasive Network Monitor for the SPAX Parallel Computer)

李承久 *

(Lee Sung Gu)

요 약

본 논문에서는 SPAX 병렬 컴퓨터를 위한 온라인 무간섭 네트워크 성능 감시기의 하드웨어 부분의 설계 및 테스트에 관해서 논술하였다. SPAX 병렬 컴퓨터에서는 4 개의 Intel P6 프로세서가 하나의 프로세싱 노드를 구성하며 최대 256개의 프로세서까지 사용할수 있다. 노드들을 연결해 주는 상호연결망은 Xcent-net 라고 불리는 이중 2계층 크로스바 구조의 네트워크이다. SPAX 병렬 컴퓨터에서 높은 성능을 얻기 위해서는 네트워크의 정상적이고 효율적인 동작이 필수적이다. 따라서 시스템 성능을 저하시키지 않으면서 온라인으로 Xcent-net의 성능을 감시하고 필요에 따라서 튜닝을 할수 있는 하드웨어 및 소프트웨어 장치는 매우 유용하게 사용될수 있다. 본 연구에서는 이러한 성능 감시를 위한 저렴한 실용적인 방법을 제시했으며 이 방법의 실용성을 입증하기 위해서 노드 모니터 보드와 PC 인터페이스 보드를 개발했고 성공적으로 테스트했다.

Abstract

This paper describes the design and test of an on-line non-invasive network performance monitor (hardware portion) for the SPAX parallel computer. The SPAX parallel computer supports up to 256 Intel P6 processors with 4 P6 processors constituting a processing node. The nodes are interconnected with a dual two-level crossbar network called Xcent-net. Since the performance of the SPAX parallel computer is highly dependent on the proper and efficient operation of the network, an on-line non-invasive network performance monitor (with hardware and software components) has been developed to aid in the monitoring and tuning of the Xcent-net. Successful testing of a prototype node monitor board and PC interface system shows that our monitor design provides a low-cost practical solution to this problem.

I. 서 론

성능 감시기 (모니터) 는 시스템의 동작을 관찰하는데 이용되는 툴이다. 성능 감시기는 시스템 성능의 통계를 보여줄 뿐만이 아니라 시스템 동작에 있어서의 중요한 타이밍 동작, 비정상적인 동작으로 인한 성능 저하 요인, 성능을 저하시키는 병목지점 분석, 시스템

자원의 효율적인 관리를 위한 정보등 다양한 정보를 제공할수 있다.

컴퓨터 기술이 고도화되고 병렬 처리 기술을 사용하게 됨으로써 효율적인 컴퓨터 시스템의 동작을 위해서는 시스템의 각종 파라미터를 감시할수 있는 툴들이 필요하게 된다^[1]. 대부분의 상용 및 연구용 중대형 컴퓨터에서는 이러한 감시기능을 해주는 여러가지 툴들을 제공해 준다. 예를 들자면 하드웨어 툴로는 Cray 컴퓨터에는 Hardware Performance Monitor가 있고 Intel iPSC/2 에는 Hypermon 이라는 툴과 성능 데이터 수집 전용 chip인 Multikron이 있다. 소프트웨어

* 正會員, 浦港工科大學校 電子電氣工學科
(Dept. of Electronic and Electrical Engineering,
Pohang Univ. of Science and Technology)
接受日:1997年3月24日, 수정완료일:1997年5月31日

틀로는 Intel iPSC/2상에서의 운영체제와 응용 프로그램의 성능감시를 위한 Crystal이 있고 Pablo 라이브러리와 Ctrace 라이브러리가 있다. 또한 Tandem사의 병렬 컴퓨터 시스템에서는 MEASURE^[2] 예를 들자면 Tandem사의 병렬 컴퓨터 시스템에서는 MEASURE^[2] 이라는 성능 감시용 틀이 있다. MEASURE를 사용하면 사용자는 각 CPU의 시간대별 사용도, 프로세스수등 원하는 시스템 파라미터를 감시해서 그 정보를 근거로 해서 프로세스를 한 CPU에서 다른 CPU로 옮기거나 다른 방법으로 시스템 튜닝 작업을 할수 있다. 또한 더 자세한 시스템 성능 정보를 제공해주고 부화 균등화 작업까지 지원해주는 틀로는 Quality Research 사가 Tandem시스템을 위해서 개발한 PROGNOSIS^[2] 틀이 있다.

(전술한 바와 같이 여러가지 성능 감시용 틀이 이미 개발되었으나 시스템 성능 감시의 특성상 범용적으로 사용할수 있는 틀은 없고 새롭게 개발되는 시스템을 위해서는 전용 틀이 새롭게 개발되어야 한다.) 본 논문에서는 이러한 틀을 개발한 연구결과에 대해서 논술하였다. 이 연구에서는 병렬 컴퓨터의 핵심적인 부분인 연결망을 전문적으로 감시할수 있는 시스템을 개발한 데에 특징이 있다. 이 연구에서 개발한 네트워크 성능 감시기는 다음의 디자인 요구사항을 만족시키도록 설계되었다.

- non-interference: 성능 감시기는 목적 시스템의 동작을 변화시키거나 성능을 달리 저하시키면 안 된다.
- on-line (real-time) monitoring: 목적 시스템을 계속 감시하면서 온라인으로 (실시간에) 시스템 성능에 관한 정보를 제공해야 한다.
- integration: 개발되는 성능 감시기는 목적 시스템의 개발단계에서 같이 개발되며 융합이 잘 되도록 설계한다.
- feedback: GUI (Graphical User Interface) 를 통해서 사용자가 보기 쉽게 시스템 성능 정보를 제공해야 하며 프로그램 내에서도 시스템 성능 정보를 사용할수 있어야 한다.

이 연구에서는 위의 4가지 조건을 만족시키는 네트워크 성능 감시기를 SPAX^[3] 병렬 컴퓨터를 위해서 개발했다. 개발된 네트워크 성능 감시기는 온라인 소프트웨어 부분이 포함된 완전한 감시기 시스템이며 본 논문에서는 감시기의 하드웨어 부분을 집중적으로 다

루었다. 본 논문은 다음과 같이 구성되어 있다. II장에서는 목적 시스템인 SPAX 병렬 컴퓨터에 관해서 기술하고 감시하고자 하는 네트워크에서의 패킷 종류 및 형태를 설명한다. III장에서는 본 연구에서 개발한 하드웨어 기반 네트워크 성능 감시기의 설계와 동작원리를 설명하고 IV장에서는 제작된 성능 감시기 시제품의 테스트 결과를 기술한다. 마지막으로 V장에서는 결론을 맺는다.

II. 목적 시스템 사양

1. SPAX 병렬 컴퓨터

SPAX (Scalable Parallel Architecture computer based on Xcent-net) 는 클러스터 구조를 기반으로 하는 병렬 컴퓨터로 4개의 Intel P6 프로세서가 하나의 노드를 구성하며 최대 256 개의 프로세서로 이루어진다^[3, 4]. 그림 1에서 볼수 있듯이 SPAX 는 여러개의 노드를 이중 2계층 크로스바 (Xcent-net)로 연결한 구조를 가지고 있다. 그림 1에서 M 은 메모리를 나타내며 XNIF 는 Crossbar Network Interface 의 약자로서 연결망과의 인터페이스 부분을 의미한다.

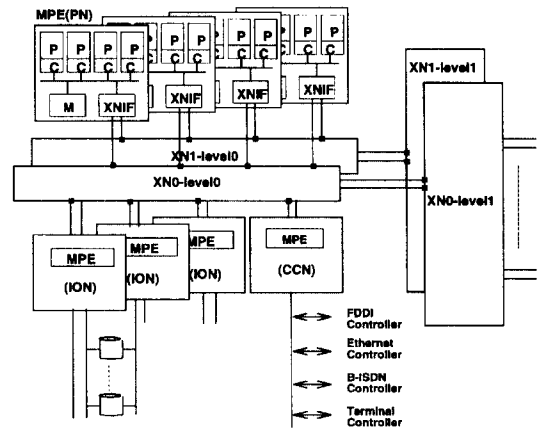


그림 1. SPAX 의 구조^[4].
Fig. 1. SPAX Architecture^[4].

노드 종류는 계산을 담당하는 PN (Processing Node), 입출력을 담당하는 ION (I/O Node) 및 통신 접속을 담당하는 CCN (Communication and Control Node) 이 있으며 최대 64개의 PN과 64개의 ION 및 CCN이 Xcent-net 로 연결이 되어 있다. 8개의 노드가 한개의 클러스터를 이루며 최대 16개의 클러스터

가 SPAX 병렬 컴퓨터를 구성한다. 각 프로세싱 노드는 SMP (Symmetric Multiprocessor) 구조로써 프로세서간의 통신은 공유 메모리를 통해서 이루어진다. 서로 다른 노드에 위치한 프로세서는 Xcent-net를 통해서 통신을 하게 되며 Xcent-net의 효율적인 운용만이 시스템의 높은 성능을 보장시킬수 있다.

2. Xcent-net를 통과하는 패킷 종류 및 형태

표 1에는 Xcent-net의 성능규격을 보여주고 있다. Xcent-net에서는 적응적인 virtual cut-through^[5, 6] 라우팅 방식을 사용하면서 근원지 전송 방식을 사용한다. 이 방식에서는 패킷을 '플릿' (플릿은 인접한 노드들간에 flow control 가 이루어지는 단위로써^[7]에서 소개된 이후 널리 사용되는 개념이다) 단위로 파이프라이닝 방식으로 보내면서 패킷의 머릿부분이 막히면 막힌 노드에서 전체 패킷을 임시저장하는 방식이다.

표 1. Xcent-net의 성능 규격^[4]

Table 1. The performance specifications for Xcent-net^[4].

| | |
|-----------|--------------------|
| 토폴로지 | 2중구조의 2계층 크로스바 |
| 최대 연결 노드 | 128 노드 (16 클러스터) |
| 최대 연결 포트 | 128개 이중 입력 및 출력 포트 |
| 전송 데이터 폭 | 2 비트 / 단일 포트 |
| 클록 속도 | 33.33 MHz |
| 최대 전송 대역폭 | 273.0 Gbps |

모든 메시지는 패킷 단위로 전송되며 별도의 시스템 제어용 연결망이 없으므로 Xcent-net를 통해서 모든 데이터 메시지와 제어 메시지를 보낸다. 패킷 종류로는 크게 일반 패킷과 망 제어 패킷이 있는데 망 제어 패킷은 Xcent-net의 특정 포트를 분리 또는 연결하도록 지시하는 패킷으로 본 연구에서 개발한 성능 감시기에서는 감시하지 않는 패킷이다. 일반 패킷으로는 제어 패킷 (control packet), 데이터 제어 패킷 (data control packet), 데이터 패킷 (data packet) 및 응답 패킷 (acknowledgement packet)등 4종류가 있다. 또한 응답 패킷에는 에러 검출에 관한 정보가 포함될수 있으며 에러 종류는 다음의 4종류가 있다: 패킷 전송 과정에서의 전송 프로토콜 에러를 나타내는 순서 오류, 인터레이스드 패리티에서 검출된 전송 오류인 패리티 오류, 수신 버퍼가 여유가 없음을 나타내는 버퍼 충만 오류, 그리고 다른 이유로써 발생하는 수신 불가 오류.

아래의 그림 2에서는 일반적인 패킷 형태를 보여주

고 있다. 패킷은 헤더와 데이터 부분으로 나뉘지며 헤더부분에 패킷종류, 수신 노드, 송신 노드, 에러유무등 패킷 내용을 제외한 모든 부분이 포함되어 있다. 헤더의 TAG 부분은 근원지 송신 방법에서 사용되는 부분으로써 연결망을 통해서 수신 노드로 전달되는 과정에서 소모되는 부분이다. 일반 패킷의 제어 정보 플릿의 형태는 그림 3에 나타나 있으며 본 성능 감시기에서 검출해서 데이터 수집에 사용하는 핵심적인 부분이다.

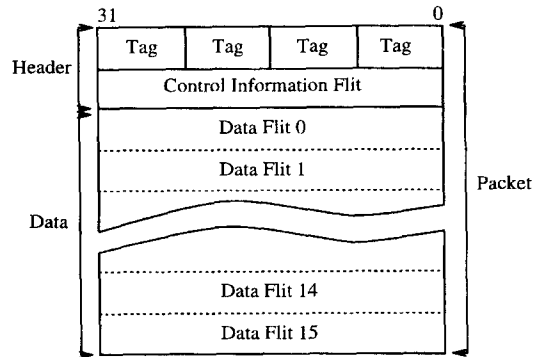


그림 2. 일반 패킷의 구조^[4]

Fig. 2. The structure of a normal packet.^[4]

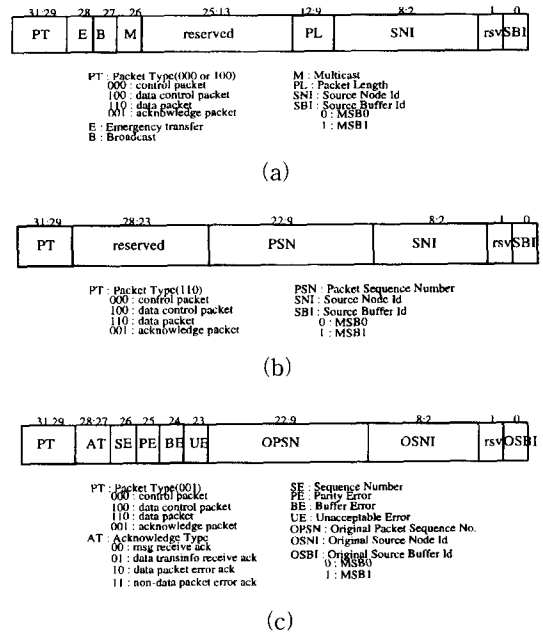


그림 3. 제어 정보 플릿: (a) 제어 또는 데이터 제어, (b) 데이터, (c) 응답 패킷^[4]

Fig. 3. Control information flit: (a) control or data control, (b) data, (c) acknowledgement packet^[4].

Ⅲ. 네트워크 성능 감시기의 디자인

본 연구에서는 Xcent-net를 온라인으로 감시하면서 시스템의 성능을 저하시키지 않는 감시기를 만드는 것을 목표로 했다. 감시기를 만드는 가장 저렴한 방법은 목적 시스템에서 실행되는 소프트웨어 감시기를 만드는 것이다. 그러나 SPAX 병렬 컴퓨터의 프로세서에서 실행되는 소프트웨어 감시기를 만들 경우에는 필연적으로 계산 시간을 빼앗게 되므로 시스템 성능이 약간 저하된다. 또한 전체 시스템에서의 통신에 관한 정보를 얻기 위해서는 모든 노드에서 수집한 데이터를 한곳에 모아야 하므로 시스템 감시에 의한 많은 양의 부가 통신을 유발하게 된다. 이러한 문제들을 감안해서 본 연구에서는 하드웨어 방법으로 Xcent-net를 감시하면서 별도의 경로를 통해서 데이터를 수집하고 별도의 PC, 워크스테이션 또는 SPAX 호스트에서 수집된 데이터를 온라인으로 가공해서 사용자에게 GUI형태로 보여주는 방식을 택했다.

네트워크의 성능을 감시하기 위해서는 네트워크를 통과하는 모든 패킷에 대한 정보를 수집하고 네트워크의 병목지점과 비정상적인 동작등을 검출하고 이 데이터를 가공해서 사용자가 분석하기 쉽게 보여야 한다. 또한 사용자의 요구사항에 의해서 특정한 종류의 패킷에 관한 정보도 제공할수 있어야 한다. 이러한 모든 기능을 가지기 위해서는 기초적인 데이터는 하드웨어로 수집하고 사용자가 보기 쉽게 데이터를 가공하는 작업은 소프트웨어로 해야할 것이며 본 논문에서는 하드웨어 부분을 다룬다.

1. 하드웨어 네트워크 성능 감시기의 전반적인 디자인

네트워크 성능 감시기의 하드웨어 부분에서는 기본적으로 Xcent-net를 통과하는 모든 종류의 패킷 갯수를 수집해야 한다. 수집하는 데이터는 패킷 갯수이므로 수집하고자하는 각종 데이터를 위해서 하드웨어 카운터를 사용하면 되며 각 노드의 수신 패킷 갯수, 각 패킷 종류의 갯수, 각 에러의 갯수등과 같은 종합된 데이터는 세부 데이터를 수집한 다음에 소프트웨어적으로 계산할수 있다. 그러므로 감시기 하드웨어에서 수집해야 하는 세부 데이터의 가지수는

수신 노드수 × 송신 노드수 × (패킷 종류 + 에러 종류) 이다.
모든 노드가 패킷의 수신 노드도 될수 있고 송신 노드

도 될수 있으므로 송수신 노드 갯수는 각각 128개이고 패킷 종류는 4가지이지만 에러는 2개 이상의 에러가 복합적으로 생길수 있으므로 유효한 에러 종류는 15가지다. 따라서 필요한 카운터 숫자는 총 $128 \times 128 \times (4 + 15) = 311,296$ 개다.

필요한 카운터의 숫자가 너무 많고 감시해야 하는 패킷이 Xcent-net에 전반적으로 분산되어 있는 문제가 있어서 감시기 하드웨어를 SPAX 시스템에서 분산시키는 방법을 사용했다. 구체적으로는 각 노드가 Xcent-net로 연결되는 지점에 하나의 노드 모니터 보드를 부착시켜서 최대 128개의 노드 모니터 보드를 직렬로 연결하고 PC 인터페이스 보드를 통해서 정보를 수집하도록 했다. 이러한 네트워크 성능 감시기의 구조를 그림 4에 나타냈다. 이러한 구조가 가능한 것은 SPAX의 backplane에 양쪽 연결이 가능하고 backplane 뒤쪽에 약간의 공간이 있기 때문이다.

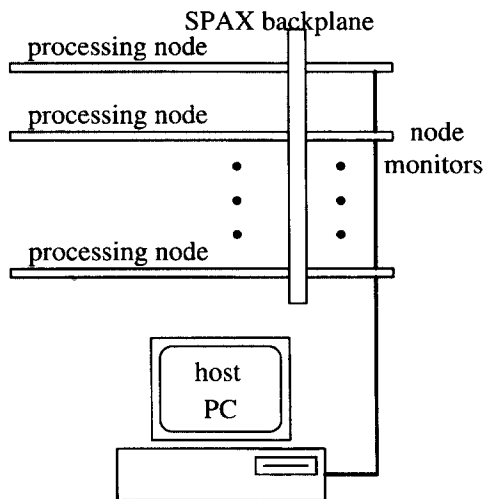


그림 4. 하드웨어 네트워크 성능 감시기의 블록도
Fig. 4. Block diagram of the hardware network performance monitor.

2. 노드 모니터 보드 디자인

각 노드 모니터 보드에는 128개의 송신 노드가 보낼수 있는 4종류의 패킷과 15종류의 에러조합 종류를 감안해서 $128 \times (4 + 15) = 2,432$ 개의 카운터가 필요하다. 각 수신 노드에 관한 정보는 각 노드 모니터 보드에 고유의 ID를 저장함으로써 각 노드 모니터에서 수집할수 있다. 그림 4에서 보인 것과 같이 모든 노드 모니터 보드는 직렬로 연결되어 있으며 각 연결지점에 버퍼를 둬으로써 파이프라인 방식으로 모든 정보를 하

나의 PC에서 수집할수 있다.

최종적으로 사용하게된 노드 모니터의 블록도를 그림 5에 나타냈다. 그림 5에서 볼수 있듯이 필요한 카운터는 고속 dual-port SRAM으로 구현했으며 파이프라이닝을 대대적으로 도입해서 Xcent-net 에서 사용하는 클럭 속도 (33.33 MHz)에 맞추어서 들어오는 모든 패킷을 온라인으로 감시할수 있도록 했다. 패킷의 동기화는 SYNC 신호에 의해서 이루어지며 유효한 첫 플릿은 VALID 신호가 내려가는 순간에 도착된다.

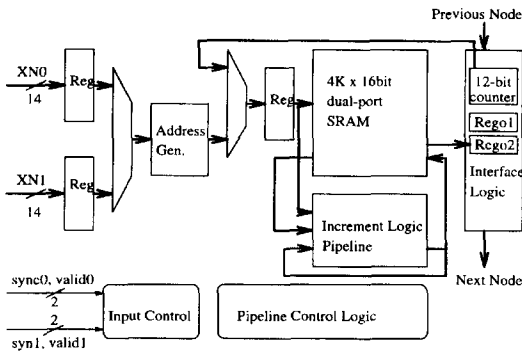


그림 5. 노드 모니터의 블록도
Fig. 5. Block diagram of the node monitor.

하나의 노드에서 패킷이 수신될 때에 도착하는 첫 플릿은 그림 3에 나타낸 제어 정보 플릿이며 이 플릿에 그 패킷에 관한 모든 정보가 포함되어 있다. 따라서 제어 정보 플릿을 register에 버퍼한 다음 송신노드 ID, 패킷 종류 및 에러 종류에 의해서 SRAM 주소를 생성하고 한 클럭 주기 뒤에 해당 카운터의 값을 SRAM에서 읽고 같은 주소로 하나 증가된 값을 저장하게 된다. 카운터의 값이 host PC로 보고될때 까지 정확도를 유지하기 위해서 SRAM에서는 16비트 데이터를 사용했다. 제어 정보 플릿을 이용한 SRAM 주소 생성 방법을 표 2에 나타냈다. 총 2,432개의 카운터가 필요하므로 표 2에수 볼수 있듯이 12개의 주소비트가 필요하며 총 용량이 64Kbit 인 4K × 16 형태의 SRAM이 필요하다.

표 2. SRAM 의 12비트 주소 형식
Table 2. 12-bit SRAM address format.

| Bit Position | Field |
|--------------|--------------------------|
| 11 --- 5 | Source Node ID |
| 4 | Error Indicator Bit |
| 3 --- 0 | Error Type / Packet Type |

만약 XN0 와 XN1의 이중 포트로 두개의 패킷이 동시에 도착할 경우에는 한 패킷을 한 클럭 주기동안 지연시켜야 한다. 또한 XN0 와 XN1 으로 도착하는 패킷이 같은 SRAM 주소를 생성할 경우에는 원래 SRAM 값에 2를 더하도록 (1을 두번 더하도록) 설계했다. 하나의 포트로 들어오는 두개의 패킷 사이에는 최소한 5클럭 주기가 경과되므로 공백이 생기는 클럭 주기 동안에 주기적으로 SRAM의 모든 데이터를 하나씩 Rego2 register로 저장시킨 다음 다른 모든 노드 모니터의 Rego1로 파이프라인으로 연결시켜서 모든 모니터 보드의 2,432개 카운터 값을 host PC로 전달하게 된다 (실제 디자인에서는 제어의 단순화를 위해서 각 노드 모니터는 4,096개의 값을 전달한다). Rego1과 Rego2의 double-buffer를 사용하고 앞단과 뒷단과 철저히 handshaking을 함으로써 데이터를 안전하게 전달하게 된다. PC로 데이터가 전달되는 순서는 N11 (node 1, data 1), N12, ... , N1n N21, N22, ... , N2n, ... , Nnn 의 고정된 순서로 고정된 간격으로 도착하게 되므로 대략적인 데이터 도착 시간 정보도 포함되어 있다. 자세한 시간 정보를 포함하는 것은 향후 연구에 포함시킬 계획이다.

PC 인터페이스 부분은 8255PPI를 사용한 간단한 I/O 인터페이스를 가진다. 하나의 데이터 세트는 4,096 × 128 노드 = 524,288 이며 사용자에게 온라인 (실시간) 감각을 주기 위해서는 약 1초이내에 새로운 데이터 세트를 받아서 처리해야 하므로 하나의 16비트 카운터 값은 2 μ 이내에 처리되어야 한다. 그러나 카운터 값의 overflow 를 근본적으로 불가능하게 만들기 위해서는 약 0.02 μs 이내에 하나의 16비트 카운터 값이 처리되던가 24비트 카운터를 사용해야 한다 (2 × 33.33MHz / 5 = 13.332Mhz 주파수로 하나의 SRAM 주소가 계속 증가될수 있으므로 1초 간격으로 이 값이 host PC로 보고될 경우 2²⁴ = 16Mega 까지의 카운팅 기능이 필요하다).

IV. 시제품 테스트

III장에서 기술한 방법으로 설계된 노드 모니터 보드의 시제품을 제작해서 실제 SPAX 병렬 컴퓨터에서 테스트를 했다. 6층 PCB로 제작된 이 시제품에는 한 개의 Actel 1280XL FPGA, 한개의 4K × 16 15-ns dual-port SRAM, 노드 ID 저장을 위한 8비트 DIP

switch, 그리고 Xcent-net 의 이중 포트에 연결하는 connector와 다음 노드 모니터로 연결하는 connector 등이 있다. 시제품 노드 모니터 보드 2개를 Xcent-net 에 연결하고 ribbon cable 로 노드 모니터 보드와 host PC를 연결해서 테스트를 하였다.

SPAX 시스템이 아직도 완전히 디버깅되지 않은 상태여서 25MHz 주파수로 동작하는 SPAX 시제품을 테스트 환경으로 사용하였다. SPAX 시제품은 하나의 클러스터로써 최대 8개의 노드 보드를 장착할수 있다. 이 시제품의 나머지 사양은 II장에 기술한 것과 같으나 아직은 디버깅 되고 있는 단계이다. 본 연구에서 개발한 네트워크 모니터를 테스트하는데에는 SPAX 시제품에 2 개의 프로세싱 노드를 장착해서 25MHz 주파수로 테스트하였다. (그러나 별도의 test pattern generator 를 이용한 테스트를 통해서 노드 모니터의 동작을 33.33MHz 에서도 검증하였다.) 노드 모니터 보드들간의 파이프라인을 통한 데이터 교환은 double-buffer와 handshaking을 통한 아주 안정적인 방식을 채택했으므로 본 테스트에서는 2개의 보드만 장착해서 테스트했으며 여러개의 노드 모니터 보드를 장착한 테스트는 SPAX 시제품이 여러개의 클러스터로 확장된 뒤에 하겠다.

나 있고 이 값과 송신 노드 ID (0) 를 사용해서 SRAM 주소 092H를 발생한 것을 볼수 있다. 그림 7에서는 SRAM의 패킷 카운터를 증가하는 모습을 잡았다. SRAM 주소 254H에 있는 데이터 AD1DH를 읽어서 같은 주소에 하나 증가된 값인 AD1EH를 저장하는 것을 볼수 있다. 이러한 방식으로 노드 모니터 보드의 테스트 가능한 모든 기능이 정상적으로 동작하는 것을 확인했으며 약간의 impedance matching 작업을 마친 후에는 모든 신호가 아주 깨끗하게 나오는 것을 확인했다.

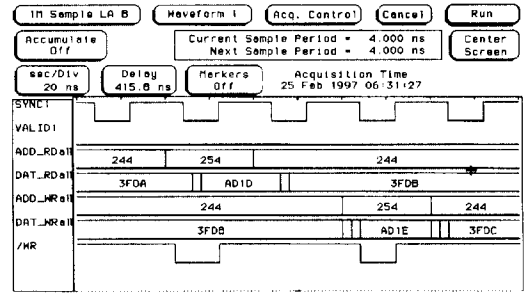


그림 7. 패킷 카운팅 테스트
Fig. 7. Packet counting test.

V. 결 론

본 논문에서는 SPAX 병렬 컴퓨터를 위한 온라인 무간섭 네트워크 성능 감시기의 설계 및 테스트에 대해서 기술하였다. 무간섭과 온라인 조건을 만족하기 위해서 하드웨어 감시기를 설계했으며 필요한 네트워크 데이터를 효과적으로 수집하기 위해서 SPAX의 128개 노드에 각각 분산된 노드 모니터 보드를 설계했다. 노드 모니터 보드는 SPAX 시스템의 backplane 의 뒷면에 연결되어서 SPAX의 연결망인 Xcent-net를 통과하는 모든 패킷에 관한 정보를 수집한다. 128까지 연결되는 노드 모니터 보드는 또한 직렬로 서로 연결되어서 host PC 시스템으로 모든 데이터를 보낸다.

목적 시스템인 SPAX의 Xcent-net 연결망이 33.33 MHz 주파수로 작동하도록 설계되어 있으므로 본 네트워크 모니터도 이 속도로 작동해야 한다. 따라서 33.33MHz 의 빠른 속도로 들어오는 모든 패킷을 검출하고 각 종류의 패킷 갯수 정보를 온라인으로 수집하고 사용자에게 보여 주기 위해서 고속 dual-port SRAM을 사용하고 파이프라이닝을 대대적으로 사용해야 했다. 하나의 SPAX 노드를 감시하는 노드 모니터

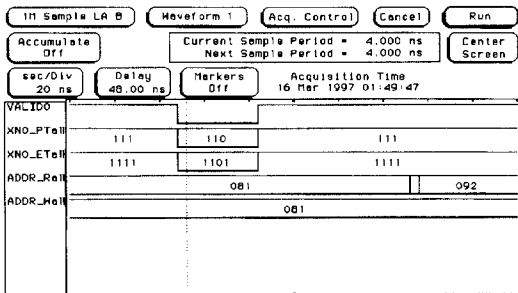


그림 6. 응답 패킷이면서 버퍼 충만 에러가 났을때의 테스트
Fig. 6. Test for an acknowledgement packet with a buffer full error.

SPAX 시제품에서는 여러 종류의 긴 테스트 패턴을 사용해서 테스트 했으며 이 테스트 결과의 일부를 그림 6과 7에 나타냈다. 그림 6은 응답 패킷이면서 버퍼 충만 에러가 났을때에 SRAM 주소를 생성하는 과정을 logic analyzer로 잡은 것을 보여준다. XNO\verb\|_PTell과 XNO\verb\|_ETell에 VALID0 신호가 내려가는 순간에 검출된 패킷 종류와 에러 종류가 나타

보드에서는 SRAM 이외의 논리는 전부 하나의 중간크기 FPGA chip으로 포함시켰다. 설계한 네트워크 감시기의 정상 동작을 확인하기 위해서 6층 PCB로 노드 모니터 보드를 제작해서 실제 SPAX 시스템에서 테스트를 했다. 여러가지 상황에서의 테스트를 실행하고 테스트 결과를 logic analyzer과 digital oscilloscope로 확인했다. 따라서 당초 목표한 대로 SPAX 병렬 컴퓨터를 위한 실용적인 무간섭 온라인 네트워크 성능 감시기를 개발하는데에 성공했다. 향후 연구로는 시간에 따른 네트워크 성능 데이터 수집 (타임스탬프 기록) 과 성능 감시기를 통해서 수집된 데이터의 사용방법을 연구할 계획이다.

참 고 문 헌

- [1] D. Haban and D. Wybranietz, "Monitoring and measuring parallel systems," in *Proc. 3rd Annual Parallel Processing Symposium*, pp. 499-513, Mar. 1989.
- [2] T. C. Inc., *Guardian 90 Operating System User's Guide*. Cupertino, CA: Tandem Computers Inc., 1989.
- [3] Y. W. Kim, S. W. Oh, and J. W. Park, "Design issues and system architecture of ticom iv, a highly parallel commercial computer," in *Proc. 3rd Euromicro Workshop on Parallel and Distributed Processing*, pp. 219-226, Jan. 1995.
- [4] M. H. Kim and M. H. Kang, "Memorandum on the packet formats and types for the xcent-net," tech. memo., ETRI Parallel Programming Laboratory, Aug. 1995.
- [5] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique," *Computer Networks*, vol. 3, pp. 267-286, 1979.
- [6] H. S. Lee, H. W. Kim, J. Kim, and S. Lee, "Adaptive virtual cut-through as an alternative to wormhole routing," in *Proc. 24th Int'l Conf. on Parallel Processing*, (Oconowoc, MI), pp. 68-75, Aug. 1995.
- [7] W. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Comput.*, vol. C-36, pp. 547-553, May 1987.

— 저 자 소 개 —



李承久(正會員)

1985년 University of Kansas 전기공학과 학사. 1987년 University of Michigan 전기공학 및 컴퓨터 과학과 석사. 1990년 University of Michigan 전기공학 및 컴퓨터 과학과 박사. 1990년 델라웨어 대학교 전기공학과 조교수 및 1991년 ~ 현재 포항공과대학교 전자전기공학과 부교수 근무. 주관심분야는 병렬 컴퓨터 및 결합포용 컴퓨터.