

고속 중형 컴퓨터 통합 시험 및 성능 분석을 위한 버스 감시기의 설계 및 구현

(Design and Implementation of a Bus Monitoring Instrument for the TICOM-III Integration Test and Performance Analysis)

韓宗錫 * , 宋庸鎬 **

(Jong-Seok Han, and Yong-Ho Song)

요약

버스를 기반으로하는 공유 메모리 다중처리 시스템에서 모든 데이터는 버스를 통하여 전송되기 때문에 버스에 구동되는 데이터를 검색하면 시스템 동작상의 오류를 발견할 수 있고 정확한 버스 성능을 예측할 수 있다. 본 논문은 주전산기 III의 시스템 통합 시험과 시스템 버스 성능 분석을 지원하는 도구인 버스 정보 처리기의 설계 및 구현에 관한 것이다. 버스 정보 처리기는 시스템 버스에 구동되는 데이터를 감시, 저장, 검색, 처리하여 시스템 통합 시험시에 필요한 다양한 트리거 조건과 함께 여러가지 정보들을 각 기능 및 목적에 따라 제공하고, 시스템 버스 성능 분석시에 필요한 버스 이용율, 읽기 쓰기 비율등 기본 정보를 제공한다.

Abstract

On a bus-based shared memory multiprocessing system, the system bus monitoring and analysis are crucial for system integration test and performance analysis. In this paper, the design and implementation of a bus monitoring instrument for the TICOM-III system are described. The instrument dedicated to TICOM-III, which is called the Bus Information Processing Unit, analyzes the bus state and measures the bus utilization. It performs many useful functions to help debugging the system, and offers a simple user interface.

I. 서론

상용화된 대부분의 다중처리 시스템은 구현이 용이하고 설계 비용이 적은 버스 구조를 기반으로 하고 있

다.

버스 구조는 설계 및 구현이 용이하여 대략 20개 이내의 프로세서를 갖는 중형급 이하 컴퓨터 시스템에 주도적으로 사용되어 왔다.

버스 구조와 공유 메모리를 기반으로 하는 다중처리 시스템에서 각 프로세서 보드, 입출력 처리기 보드 및 시스템 제어기 보드는 버스와 메모리를 공유하며 모든 데이터의 전송은 버스를 통하여 이루어 진다. 따라서, 시스템 버스를 통해 전송되는 데이터를 검색하면 시스템의 동작 상태를 파악하여 동작상의 에러 유무를 발견하고 실제적인 시스템 성능을 유추할 수 있다.

* 正會員, 韓國電子通信研究所 프로세서 研究室 (Processor Section, ETRI)

** 正會員, 三星電子 컴퓨터시스템 事業部 (SAMSUNG ELECTRONICS CO., LTD. Computer Systems Div. R & DI)

接受日字: 1995年3月4日, 수정완료일: 1995年7月29日

본 논문은 동기식 파이프라인드 버스인 HiPi+Bus (Highly Pipelined Plus Bus)¹¹⁾를 기반으로하는 고속중형 컴퓨터(주전산기-III)¹²⁾의 버스 데이터를 검색, 저장, 처리하여 시스템 버스 상태 분석을 통한 통합 시험 지원 및 성능 분석을 지원하는 도구인 버스 정보 처리기(Bus Information Processing Unit, BIPU)의 설계 및 구현에 관한 것이다.

종래의 비동기식 버스 점유형의 버스 상태 분석기¹³⁾나 동기식 파이프라인드 HiPi-Bus(Highly Pipelined Bus) 전용 버스 상태 분석기¹⁴⁾는 고속중형 컴퓨터에 직접 사용할 수 없고 그 기능 또한 단순한 것이어서 고속중형 컴퓨터에 적합하지 않은 문제점을 갖고 있다.

본 버스 정보 처리기는 다양한 트리거 환경을 가지고 버스의 데이터를 검색, 저장, 처리하는 기능과 시스템 버스의 사용을 측정등 시스템 버스 성능 분석 기능 그리고 응답기를 제공하여 요청기가 시스템 버스상에서 버스 정보 처리기에 검색과 저장을 명령하고 저장된 데이터를 요청할 때 요청기에 데이터를 전송하는 기능등을 제공하므로써 고속중형 컴퓨터에 보다 적합하게 사용할 수 있다.

(그림 1)은 프로세서 보드, 메모리 보드, 입출력 제어기 보드, 시스템 제어기 보드, 그리고 파이프라인드 버스로 구성된 고속중형 컴퓨터의 개략적인 구성을 보여준다.

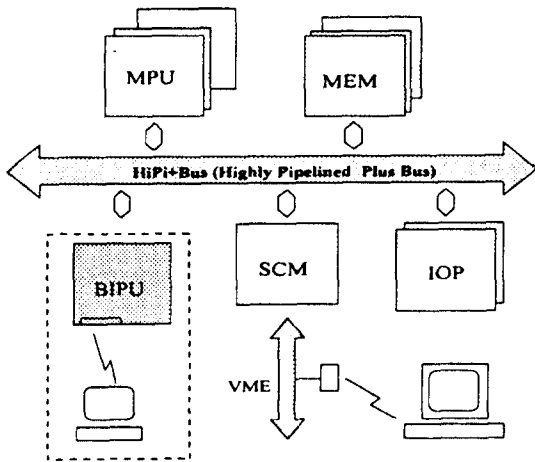


그림 1. 고속중형 컴퓨터의 개략 구성도
Fig. 1. Hardware configuration of the TICOM-III.

버스 정보 처리기는 기본적으로 고속중형 컴퓨터의 기본 구성에 포함되지 않으며 시스템의 통합 시험이나 시스템 버스의 성능 분석시등 필요에 따라 시스템 버스 백플레인에 장착되어 시스템 버스에 구동된 데이터를 검색, 저장, 처리하여 특정 상황의 시스템 버스의 상태를 분석하거나 성능을 측정하여 이를 사용자에게 제공한다.

2장에서는 버스 정보 처리기의 주요 기능에 대해 구체적으로 기술하며 3장에서는 버스 정보 처리기의 검색 신호선 및 주요 규격에 대해 기술하고 4장에서는 하드웨어 및 펌웨어의 내부 구조 및 세부 모듈 기능에 대해 기술한다. 5장에서는 버스 정보 처리기의 구현 환경 및 구현 방법, 사용자 인터페이스등을 기술하고 마지막으로 6장에서는 설계 및 구현시 고려사항을 간략히 기술한다.

II. 주요 기능

버스 정보 처리기는 고속중형 컴퓨터(주전산기-III)의 통합 시험 및 성능 측정에 필요한 도구로써 고속중형 컴퓨터의 시스템 버스인 HiPi+Bus에 구동되는 모든 신호들을 감시, 저장, 검색, 가공하여 사용자의 요구에 따라 여러가지 형태의 정보로 사용자에게 제공한다. 버스 정보 처리기의 중요한 기능들은 다음과 같다.

첫째, 통합 시스템 구축을 지원하는 도구로써 시스템 버스에 구동되는 모든 신호들을 감시, 저장, 검색, 분석하여 시스템 통합 구축시에 발생할 수 있는 여러가지 시스템 오류나 부적절한 동작상태들을 사용자(시스템 개발자 또는 통합 시험자)에게 알려주므로써 사용자로 하여금 이에 따른 적절한 조치를 취할 수 있도록 한다.

둘째, 시스템 성능 분석 지원 도구로써 어드레스 버스 사용을 및 데이터 버스 사용율등 기본적인 정보를 사용자에게 제공한다. 이는 시스템 버스에 구동되는 데이터들을 추적하여 실제 버스 사용율을 사용자에게 알려주므로써 정확한 시스템 성능을 예측할 수 있도록 지원한다.

셋째, 응답기(ResPonder) 기능을 제공하여 요청기(ReQuester)에서 검색 명령을 지시하고 이의 결과인 검색 메모리의 데이터를 요청기에서 접근 가능하도록 한다.

이 응답기 기능은 버스 정보 처리기의 전용 단말기를 사용하지 않고 시스템 소프트웨어에서 접근 가능하도록 하여 온라인 시스템 분석 기능을 지원하는 것이다.

1. 시스템 버스 검색 기능

버스 정보 처리기는 시스템 버스에 구동되는 모든 신호들을 감시하여 사용자가 지정한 트리거 조건(trigger condition)을 만족하는 신호들을 검색 메모리에 저장하고 이를 각 버스 신호선 모임으로 그룹화하여 사용자에게 제공한다.

사용자가 지정하는 트리거 조건은 버스 정보 처리기에서 제공하는 기능에 의존하기 때문에 사용자로서 금 보다 다양한 트리거 조건을 사용할 수 있도록 많은 기능을 제공하여야 한다.

본 논문의 버스 정보 처리기는 트리거 조건의 형태에 따라 버스 동작 기본 주기 트리거 조건과 버스 클럭 주기 트리거 조건을 제공하며, 트리거 조건의 로직에 따라 AND 트리거 조건 로직과 AND-OR 트리거 조건 로직을 제공한다. 또한, 트리거 조건을 만족하는 데이터의 저장 방법에 따라 단일 블록 저장 방법과 다중 블록 저장 방법을 제공한다.

시스템 버스에서의 데이터 전송은 버스 동작(bus transaction)으로 수행되며, 버스 동작은 기본 주기(basic cycle)로 구성된다.

기본 주기는 분리될 수 없는 일정한 순서로 진행되는 단계(phase)로 구성되며, 각 단계는 한개의 버스 클럭 주기(bus clock period)가 소요된다.

버스 동작 기본 주기 트리거 조건은 일정한 단계로 진행되는 기본 주기를 트리거 조건으로하여 순차적으로 발생하는 조건들을 비교 검색하는 조건으로서 시스템 버스 프로토콜에 따르는 하나의 버스 동작을 검색할 수 있다. 읽기 버스 동작을 위한 어드레스 기본 주기와 데이터 기본 주기, 그리고 쓰기 버스 동작을 위한 어드레스 데이터 기본 주기를 트리거 조건으로 제공한다.

버스 클럭 주기 트리거 조건은 동일 버스 클럭 주기에 구동되는 신호들을 트리거 조건으로하여 동일 시점에 발생하는 조건들을 비교 검색하는 조건으로서 각 신호선들의 연관 관계를 검색할 수 있다.

AND 트리거 조건 로직과 AND-OR 트리거 조건 로직은 각 트리거 항목의 로직 조합을 트리거 조건으

로하여 검색하는 조건이다.

AND 트리거 조건 로직은 모든 항목이 AND 로직을 만족하는 경우의 조건을 지정하며, AND-OR 트리거 조건 로직은 일부 항목은 AND 로직을 만족하고 일부 항목은 OR 로직을 만족하는 경우의 조건을 지정한다.

트리거 조건을 만족하는 데이터의 저장 방법중 단일 블록 저장 방법은 하나의 트리거 만족 데이터를 중심으로 연속된 전후의 다른 데이터들과 함께 물리적으로 하나인 단일 블록에 저장하는 방법이며, 다중 블록 저장 방법은 물리적으로는 하나이지만 논리적으로 다중화되어 있는 다중 블록에 각 블록마다 하나의 트리거 만족 데이터를 저장하여 동일한 트리거 조건을 만족하는 복수개의 데이터를 저장하는 방법이다.

단일 블록 저장 방법은 트리거 만족 데이터를 중심으로 많은 양의 주변 데이터를 검색할 수 있어 순차적인 상황 판단에 유용하며, 다중 블록 저장 방법은 각 트리거 만족 데이터의 주변 데이터 양은 적지만 복수개의 동일 트리거 만족 데이터들을 비교할 수 있어 반복적인 상황 판단에 유용하다.

<표 1>은 트리거 조건의 형태, 트리거 조건의 로직, 그리고 트리거 데이터의 저장 방법에 따라 분류한 도표이다.

표 1. 형태와 로직, 그리고 저장방법에 따른 트리거 조건의 분류

Table 1. A taxonomy of trigger conditions.

트리거 조건 형태	버스 동작 기본 주기 트리거 조건
	버스 클럭 주기 트리거 조건
트리거 조건 로직	AND 트리거 조건 로직
	AND-OR 트리거 조건 로직
저장 방법	단일 블록 저장 방법
	다중 블록 저장 방법

1) 버스 동작 기본 주기 단일 블록 저장 기능

어드레스 기본 주기, 데이터 기본 주기 또는 어드레스 데이터 기본 주기를 트리거 조건으로 하여 각 기본 주기에 대해 검색하거나 어드레스 기본 주기와 데이터 기본 주기를 하나의 트리거 조건으로 하여 버스 동작(bus transaction)을 검색하는 기능을 제공한다.

- 어드레스 기본 주기

- 데이터 기본 주기
- 어드레스 기본 주기 + 데이터 버스 기본 주기 (읽기 버스 동작)
- 어드레스 데이터 기본 주기(쓰기 버스 동작)

각 기본 주거나 버스 동작안에서 각 신호선의 AND 로직이나 AND 와 OR 로직을 조합한 트리거 로직을 제공한다. 트리거 조건을 만족하는 데이터를 중심으로 일정량의 다른 버스 데이터를 저장하는데, 연속된 128 K 버스 클럭 주기의 데이터를 저장할 수 있다.

2) 버스 동작 기본 주기 다중 블록 저장 기능

각 기본 주거나 버스 동작안에서 각 신호선의 AND 로직이나 AND 와 OR 로직을 조합한 트리거 로직을 제공한다. 동일 트리거 조건에 대하여 이를 만족하는 복수개의 버스 데이터들을 저장하는 기능으로 최대 1024 개의 블록을 제공한다. 각 블록당 트리거 조건을 만족하는 하나의 버스 데이터를 포함한 128 버스 클럭 주기의 데이터를 저장할 수 있다.

3) 버스 클럭 주기 단일 블록 저장 기능

한 버스 클럭에 구동되는 각 신호들의 AND 트리거 로직이나 AND & OR 조합 트리거 로직을 만족하고 버스 정보 처리기의 버스 인터페이스에 처음으로 감지 되는 버스 신호선을 저장한다. 트리거 조건을 만족하는 데이터를 중심으로 일정량의 다른 버스 데이터를 저장 하는데, 연속된 128 K 버스 클럭 주기의 데이터를 저장할 수 있다.

4) 버스 클럭 주기 다중 블록 저장 기능

한 버스 클럭에 구동되는 각 신호들의 AND 트리거 로직이나 AND & OR 조합 트리거 로직을 만족하는 복수개의 버스 데이터들을 저장하는 기능으로 최대 1024 개의 블록을 제공한다. 각 블록당 트리거 조건을 만족하는 하나의 버스 데이터를 포함한 128 버스 클럭 주기의 데이터를 저장할 수 있다.

2. 시스템 성능 분석 기능

시스템 성능 예측에 있어 가장 중요한 요소인 어드레스 버스 사용율 및 데이터 버스 사용율, 그리고 재시도와 캐쉬 관련 상태 버스 구동 빈도수에 관한 정보를 제공한다. 어드레스 버스 사용율 및 데이터 버스 사용율은 시스템 버스가 수용할 수 있는 최대 성능의 한계를 예측 가능하게 해주기 때문에 버스를 사용하는 고속중형 컴퓨터의 개발에 있어 반드시 측정되어야 할 중요한 요소이다. 버스 정보 처리기는 시스템 버스에

구동되는 신호들의 빈도수를 조사하여 일정 기간 동안의 버스 사용율에 관한 정보를 제공한다. 이러한 기능을 제공하기 위하여 버스 정보 처리기의 하드웨어에서 각 신호에 대한 16 비트 계수기 기능을 제공한다.

3. 응답기 데이터 전송 기능

버스 정보 처리기는 응답기 기능을 제공하여 요청기로부터 검색 저장 요청을 전달받고 검색 저장 기능을 수행하며, 수행 결과로써 얻은 검색 메모리의 데이터를 시스템 버스를 통해 다시 요청기에 전송하는 기능을 제공한다.

버스 정보 처리기에 의해 검색되고 저장된 데이터들은 이 데이터 전송 기능을 통해 시스템에 연결된 터미널이나 네트워크에 연결된 터미널을 통해서 사용자에게 제공 가능하게 되며, 따라서 사용자는 버스 정보 처리기의 전용 단말기를 사용하지 않고도 원하는 결과를 얻을 수 있게 된다.

버스 정보 처리기에 의해 검색되는 데이터들은 일정 시점을 기준으로한 데이터들이기 때문에

단순히 이 데이터들만으로는 정확한 시스템 성능을 예측하기 어렵다. 또한, 이 기능은 사용자에게 많은 샘플링을 허용하여 보다 정확한 시스템 성능 예측이나 다양한 상황에서의 시스템 동작을 규명할 수 있도록 한다. 사용자는 필요한 시점에 검색 저장된 데이터들을 화일에 저장하고 저장된 여러 화일들을 비교, 검토하여 보다 정확한 시스템의 동작이나 시스템의 성능을 예측할 수 있다.

Ⅲ. 검색 신호 및 주요 규격

1. 검색 신호

HiPi+Bus는 중재 버스, 데이터 전송 버스, 인터럽트 버스 및 유틸리티 버스로 세분되며 총 293 개의 신호선으로 구성된다. 버스 정보 처리기는 버스에 구동되는 신호들 중에서 클럭과 슬롯 어드레스, JTAG Boundary Scan 신호선을 제외한 282 개의 신호들과 1 개의 외부 트리거 요소 신호를 포함한 총 283 개의 신호들을 검색한다. 또한, 이러한 신호들이 구동되는 정확한 시점에서 유효 데이터를 검색하기 위해 BCLK 의 클럭 신호선과 슬롯 어드레스 신호선 GA <4.0>의 5 비트를 사용한다.

<표 2>는 버스 정보 처리기에서 검색하는 신호들을 보여준다.

표 2. 검색 신호선들

Table 2. Snooped system bus signals.

Bus	Mnemonic	Size	Name
중재버스	ABRQ<12..0>*	13	Address Bus Request
	ABINH*	1	Address Bus Arbitration Inhibition
	WRINH*	1	Write Cycle Inhibition
	DBRQ<8..0>	9	Data Bus Request
	DBINH*	1	Data Bus Arbitration Inhibition
	PCW*	1	Priority Change Window
데이터전송 버스 (어드레스 버스)	A<31..4>*	28	Address
	AP<3..0>*	4	Address Parity
	SI<7..0>*	8	Source Identification
	SIP*	1	Source Identification Parity
	AS<2..0>*	3	Address Space
	TT<4..0>*	5	Transfer Types
	STP*	1	Space+Types Parity
	BE<15..0>*	16	Byte Enable
	BEP<1..0>*	2	Byte Enable Parity
	AE*	1	Address Cycle Enable
데이터전송 버스 (데이터 버스)	D<127..0>*	128	Data
	DP<15..0>*	16	Data Parity
	DK<7..0>*	8	Destination Identification
	DIP*	1	Destination Identification Parity
	DE*	1	Data Cycle Enable
데이터전송 버스 (상태버스)	AACK<1..0>*	2	Address Acknowledge
	SHD*	1	Hit on Shared Line
	DTY*	1	Hit on Dirty Line
	SNK*	1	Snoop No Acknowledge
	ITV*	1	Intervention
	LCR*	1	Hit on Interlocked Region
	DACK*	1	Data Acknowledge
	CDK*	1	Cache Data Acknowledge
	SPIN<3..0>*	4	Spin Queue Order
	BSY<7..0>*	8	Busy Status Line
인터럽트 전송버스	IBSYNC*	1	Interrupt Bus Sync.
	IBD<7..0>*	8	Interrupt Bus Data
	IBDP*	1	Interrupt Bus Data Parity
유틸리티 버스	RST*	1	System Reset
	SFAIL	1	System Fail
외부 신호	EXT*	1	External Input
	total	283	

2. 하드웨어 규격

버스 정보 처리기는 20 MHz의 MC68020 프로세서를 장착하며 5.12 MBytes 용량의 검색 메모리(trace memory)를 갖는다. 또한, 프로세서의 동작을 규정하는 256 KBytes 의 EPROM과 지역 데이터를 처리하기 위한 512 KBytes 의 지역 메모리(local memory)를 갖는다. 데이터 전송 프로토콜은 시스템 버스의 동기형 팬디드 프로토콜을 사용하며, 16.5 MHz의 클럭 속도를 포함한 모든 규격은 시스템 버스

의 규격에 따른다.

(표 3)은 버스 정보 처리기의 하드웨어 규격을 기술한다.

표 3. 버스 정보 처리기의 주요 규격

Table 3. Hardware specifications of the BIPU.

프로세서(processor)	
프로세서 클럭속도	MC 68020 20 MHz (50 nsec)
검색 메모리(trace memory)	
메모리특성	Static RAM
역세스시간	25 ns
용량	5.12 MBytes
검색 주기	128 K clock × 40 Bytes
ROM(read only memory)	
ROM 특성	EPROM
역세스시간	120 ns
용량	256 KBytes
RAM (random access memory)	
RAM특성	Static RAM
역세스시간	25 ns
용량	512 KBytes
버스 인터페이스(Bus Interface)	
소자특성	BTL(Backplane Transceiver Logic)
데이터 전송 버스 (Data Transfer Bus)	
프로토콜 제어방식	Pended protocol 동기형 (Synchronous)
클럭속도	16.5MHz (60.6nsec)

IV. 내부 구조 및 세부 모듈 기능

1. 하드웨어 구조 및 세부 모듈 기능

버스 정보 처리기의 하드웨어는 시스템 버스와의 인터페이스를 위한 버스 정합 모듈, 시스템 클럭의 분배를 위한 내부 클럭 생성기 모듈, 버스에 구동된 신호들을 저장하기 위한 검색 메모리 모듈, 요청기들과의 데이터 전송을 위한 응답기 모듈, 구동된 신호들의 검색과 저장을 명령하는 프로세서 모듈 그리고 응답기 모듈이나 프로세서 모듈로 부터 검색과 저장 명령을 받고 트리거 조건을 비교하여 검색 메모리에 데이터를 저장하는 기능 제어 모듈등이 서로 연관 관계를 가지고 수행된다.

(그림 2)는 버스 정보 처리기의 내부 구조를 보여준다.

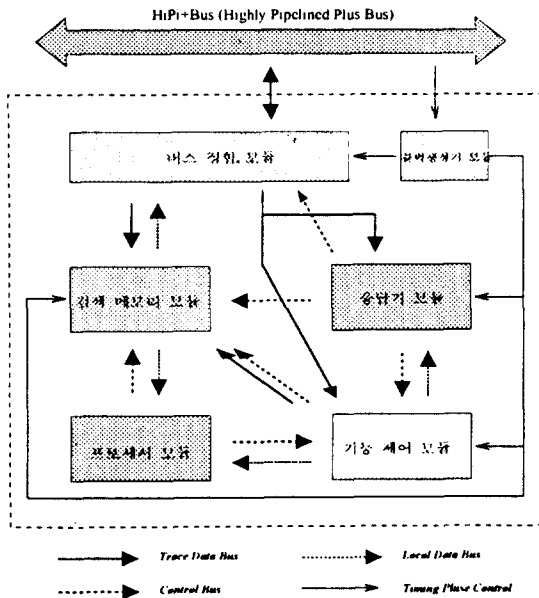


그림 2. 버스 정보 처리기의 내부 구조
Fig. 2. Architecture of the Bus Information Processing Unit.

1) 버스 정합 모듈

시스템 버스에 구동된 데이터를 매 주기마다 감지하여 검색 메모리 모듈과 기능 제어 모듈 그리고 응답기 모듈로 전송한다. 감지 시점과 전송 시점은 클럭 생성기 모듈에서 생성된 클럭에 의해 제어된다. 또한, 응답기 모듈의 제어신호에 의해 검색 메모리 모듈의 데이터를 시스템 버스에 구동하기도 한다.

2) 검색 메모리 모듈

매 주기마다 버스 정합 모듈에서 전송되는 버스 데이터와 매 주기마다 기능 제어 모듈에서 전송되는 시간순서 정보를 함께 저장한다. 저장할 때 저장 위치의 주소는 기능 제어 모듈에서 제공하며 저장된 데이터를 읽을 때의 주소는 프로세서 모듈이나 응답기 모듈에서 제공한다.

3) 기능 제어 모듈

프로세서 모듈과 응답기 모듈간의 중재를 수행하며 중재 결과를 각 모듈에 알리고 검색 명령을 받아 이를 수행한다. 검색 명령 수행중에 검색 메모리 모듈로 시간순서 정보와 저장 위치의 주소를 제공하며, 버스 정합 모듈에서 전송된 데이터를 기능 제어 모듈 내의 레지스터의 내용과 비교하여 검색 종료 시점을 결정한다. 기능 제어 모듈 내의 레지스터는 프로세서 모듈이나

응답기 모듈중 중재에서 이긴 모듈에서만 제어가 가능하다. 또한, 기능 제어 모듈은 버스 사용을 계수기를 구동하여 프로세서 모듈로 하여금 시스템 버스 성능에 대한 자료를 구할 수 있도록 한다.

4) 프로세서 모듈

사용자와의 통신 통로를 제공하기 위한 사용자 정합 기능을 수행하며 사용자의 명령에 따라 기능 제어 모듈의 레지스터 내용을 제어하고 기능 제어 모듈에 검색 명령을 지시한다. 또한, 검색 메모리 모듈에 저장된 데이터를 사용자의 요구에 따라 읽고 처리하여 사용자에게 보여준다. 프로세서 모듈은 기능 제어 모듈 내의 버스 사용을 계수기 값을 읽어 사용자에게 시스템 버스의 성능에 관한 정보를 제공한다.

5) 응답기 모듈

버스 정합 모듈로 부터 전송된 데이터를 해석하여 요청기로 부터 검색 명령이나 데이터 전송 명령이 있는지를 파악하고 각 명령에 따라 기능 제어 모듈에 검색 명령을 지시하거나 검색 메모리 모듈에서 데이터를 읽어 버스 정합 모듈에 전송하므로써 버스 정합 모듈로 하여금 시스템 버스에 유효 데이터를 구동시키도록 한다.

6) 클럭 생성기 모듈

클럭 생성기 모듈은 시스템 버스에서 시스템 클럭을 입력으로 받아 버스 정보 처리기에서 필요로하는 각 제어 신호를 생성하고 이를 각 모듈에 제공한다.

2. 펌웨어 세부 모듈 기능

버스 정보 처리기의 펌웨어는 시스템 버스 검색 기능과 시스템 성능 분석 기능을 제공하기 위해 초기화 모듈, 입력 모듈, 검색 모듈, 검색 결과 출력 모듈, 그리고 버스 이용율 표시 모듈등으로 구성된다.

1) 초기화 모듈

Exception Handler, DUART, Register, SRAM등을 각각 그 동작에 맞게 초기화한다.

2) 입력 모듈

버스 정보 처리기의 DUART를 통해 연결된 터미널을 통하여 버스 정보 처리기가 수행할 작업에 대해 명령을 입력하는 모듈이다. 이 모듈은 메뉴 방식을 사용하여 구현되며, 따라서 각 명령에 대해 구체적인 구문을 인지할 필요없이 버스 정보 처리기를 쉽게 사용가능하도록 한다. 사용자가 입력할 수 있는 명령은 크게 다음과 같이 나눌 수 있다.

- 트리거 조건의 입력
- 검색 작업 개시
- 검색 작업 중지
- 검색 결과의 출력
- 트리거 조건의 해제
- 버스 이용율의 출력

3) 검색 모듈

기능 제어 모듈에 검색 작업의 개시를 지시하고 검색 작업의 종료를 기다린다. 또한 검색 도중 사용자가 특정 키를 눌러 작업을 중단시킬 경우, 검색 작업을 중단하고, 주 메뉴로 돌아간다.

4) 검색 결과 출력 모듈

검색 메모리에 저장된 데이터를 데이터의 특성에 맞는 형태로 화면에 출력한다. 즉, 어드레스 버스, 데이터 버스, 인터럽트 버스별로 데이터를 출력한다. 데이터 출력 상태에서 사용자는 특정 데이터를 검색할 수 있으며, 전후로 스크롤시키면서 버스 데이터를 살펴 볼 수 있다.

5) 버스 이용율 표시 모듈

일정 기간의 버스 이용율을 히스토그램 형태로 화면에 출력한다. 즉, 버스의 정보가 버스 정보 처리기의 검색 메모리에 저장되면, 펌웨어에서 이를 분석하여 주소 버스, 데이터 버스, 인터럽트 버스별로 각각의 사용율을 계산한다. 또한 단일 전송대 블럭 전송의 비율, 입기대 쓰기 전송의 비율, 그리고 버스상에 장착된 요청기별로 버스에 대한 사용율등을 계산한다.

V. 구현 및 사용자 인터페이스

1. 구현 환경

버스 정보 처리기 구현의 주 도구(main tool)로는 Cadence사에서 제공하는 Design Framework II를 사용하였다¹⁵⁾. Design Framework II의 스키메틱 설계 도구(schematic design entry)인 Composer를 사용하였으며, 시뮬레이션을 위해 Verilog-XL을 이용하였다. 또한, PCB 설계 준비단계로 Packager를 이용하였으며, PCB 설계와 검증을 위해 Allegro를 사용하였다.

본 구현에서는 4 개의 EPLD 설계를 수행하였는데 모두 Altera사에서 제공하는 Max+plusII 설계 도구를 PC와 Workstation에서 이용하여 설계하였다¹⁶⁾

EPLD 설계의 결과물로 Verilog-HDL 화일을 생성하여 Verilog-XL에서 보드의 부분 통합 시뮬레이션을 수행하였다.

2. 하드웨어 구현

1) 버스 정합 모듈

시스템 버스인 HiPi+Bus의 버스 정합 규격에 따르는 BTL(Backplane Transceiver Logic) 소자¹⁷⁾를 사용해 구동 신호를 송신 또는 수신한다. 버스 정보 처리기의 버스 정합 모듈은 응답기 데이터 전송 기능을 위해 데이터 중재 버스, 데이터 버스 및 상태 버스의 신호를 송수신하는 부분과 어드레스 중재 버스, 어드레스 버스, 인터럽트 버스 및 유틸리티 버스의 신호를 수신하는 부분으로 구현하였다.

2) 검색 메모리 모듈

검색 메모리 모듈은 10개의 메모리 뱅크로 구성되어 283 비트의 검색 신호와 32 비트의 시간 순서 정보를 하나의 단위로 하여 최대 128 K 개를 저장한다. 각 메모리 뱅크는 128 K 개의 32 비트 데이터를 저장하도록 4개의 1 M SRAM으로 구현되었다.

3) 기능 제어 모듈

기능 제어 모듈은 공유자원 제어를 위한 중재 기능과 트리거 조건 검색 비교 기능 그리고 검색 메모리로의 쓰기 동작을 위한 어드레스 제어 기능등을 수행한다. 기능 제어 모듈내에 중재 제어를 위한 전역 상태 천이기(state machine)와 기능 제어를 위한 기능 제어 상태 천이기가 하나의 EPLD 소자(EPM7192)로 구현되었다.

4) 프로세서 모듈

프로세서 모듈은 사용자 인터페이스를 제공하는 펌웨어를 수행하기 위한 MC68020 마이크로 프로세서, 256 K 바이트의 ROM 및 512 K 바이트의 RAM 부분과 사용자와의 물리적 인터페이스를 위한 DUART 부분으로 구성된다.

프로세서 모듈내에 어드레스를 번역하여 해당 제어 신호를 생성하는 어드레스 번역 제어기가 하나의 EPLD 소자(EPM7128)로 구현되었다.

5) 응답기 모듈

응답기 모듈은 시스템내의 요청기와의 인터페이스를 제공하여 요청기로부터 검색 명령과 데이터 전송 요청 명령을 받아 수행한다. 응답기 모듈의 기능은 프로세서 모듈의 기능과 유사하며 단지, 시스템 버스를 통한 인

터페이스를 제공한다는 것이 다르다.

응답기 모듈의 기능 수행을 위한 기능 제어 상태 천이기와 데이터 전송을 위한 중재 제어 상태 천이기가 각각 하나의 EPLD 소자(EPM7192, EPM7032)로 구현되었다.

6) 클럭 생성기 모듈

클럭 생성기 모듈은 시스템 버스에 구동되는 시스템 클럭을 입력으로 받아 내부 제어 클럭을 생성시킨다. PLL(phase-locked lookup) 소자와 지연 소자들을 사용하여 부하 균형을 고려한 제어 클럭을 생성시켰다.

3. 사용자 인터페이스

버스 정보 처리기의 펌웨어는 사용자와의 인터페이스를 위해 Event 지정 명령, Go 명령, Dump 명령, Histogram 명령등을 수행하며, 또한 사용자 요구에 따라 각 명령에 따르는 부 명령등을 수행한다.

〈그림 3〉은 버스 정보 처리기의 메뉴 구성을 나타낸다.

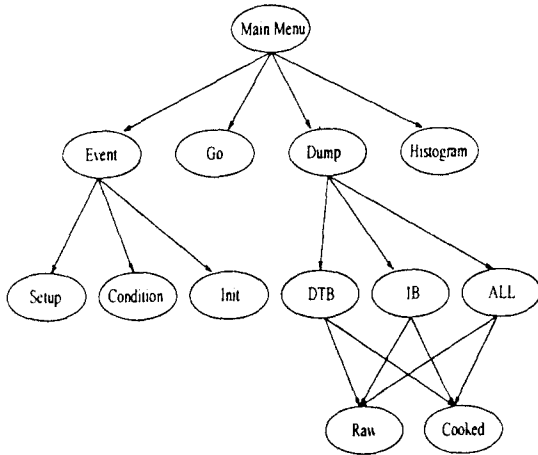


그림 3. 버스 정보 처리기의 메뉴 구성
Fig. 3. Menu structure of the BIPU.

1) Event 지정 명령

사용자가 버스상에서 발생하는 신호들중 특정한 사건이 발생하는 경우를 보고자 할 경우, 해당되는 사건을 검색조건으로 지정하는 명령이다. 이 명령에서는 신호 그룹별 혹은 신호 그룹 내부적으로 여러개의 트리거 조건을 AND, OR로 결합시켜 사용할 수 있다.

〈그림 4〉는 사용자 인터페이스를 제공하는 주 메뉴 화면을 보여준다.

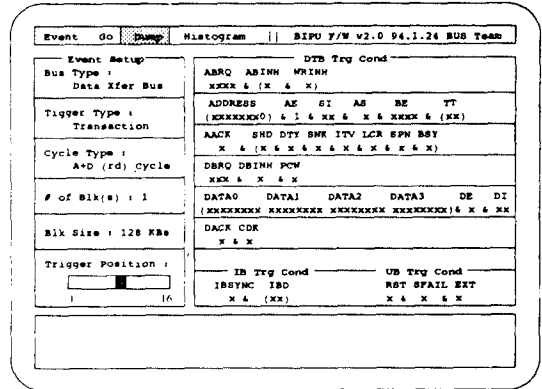


그림 4. 버스 정보 처리기의 메인 메뉴
Fig. 4. Main menu of the BIPU.

2) Go 명령

사용자가 지정한 트리거 조건의 검색을 시작시키는 명령이다. 만일 버스에서 지정한 조건이 발견되면 검색 작업을 마친다. 그러나 전체 검색 메모리를 최대 1024 개까지의 블록으로 나누어 각 블록에 대해 검색 결과를 하나씩 저장함으로써 동일한 조건에 대해 최대 1024회 저장할 수 있다.

3) Dump 명령

검색 결과로 얻어진 검색 메모리의 내용을 사용자가 볼 수 있는 형태로 화면에 출력한다. 버스 신호의 종류에 따라 데이터 전송 버스와 인터럽트 버스로 나누어 출력하며, 명령에 따라 데이터 전송 버스와 인터럽트 버스, 유틸리티 버스를 동시에 출력한다.

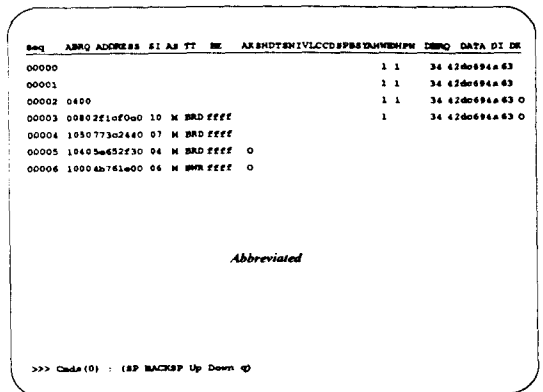


그림 5. 데이터 전송 버스의 해석 형태 출력 화면
Fig. 5. Cooked-format display of the Data Transfer Bus.

또한, 사용자가 쉽게 인지할 수 있는 해석 형태의

출력 모드와 2진수 혹은 16진수로 출력되는 순수 데이터 형태의 출력 모드를 제공한다.

<그림 5>는 여러가지 출력 형태중 데이터 전송 버스의 해석 형태 출력 화면을 보여준다.

4) Histogram 명령

시스템 버스의 이용율을 화면으로 출력시키는 명령이다. 버스 이용율은 히스토그램 형태로 출력된다. 대개의 경우 히스토그램이라하면 다음과 같은 3가지의 출력 모드를 갖는다.

- Instance Mode
- Accumulate Mode
- Average Mode

Instance Mode는 버스상에서 주어진 시간 동안의 이용율을 그대로 출력하는 모드이다. 그러나 Accumulate Mode는 Histogram 명령이 실행된 이후의 이용율을 평균하여 출력한다. 따라서 Histogram 명령의 실행시간 전체에 대해 평균 이용율을 얻을 수 있다. Average Mode는 바로 이전의 이용율과 현재 시간의 이용율 양자간의 평균치로 출력한다. 즉, Accumulate Mode에서 측정 시간을 축소한 모드라고 볼 수 있다.

일정 시간별로 버스에 대한 사용율의 변화를 쉽게 파악하는대는 주로 Instance Mode가 사용된다. 그리고 이 모드는 하드웨어의 장시간 테스트에도 아주 편리하게 사용될 수 있다. 버스 정보 처리기에서는 Instance Mode를 기본적으로 하여 지원한다.

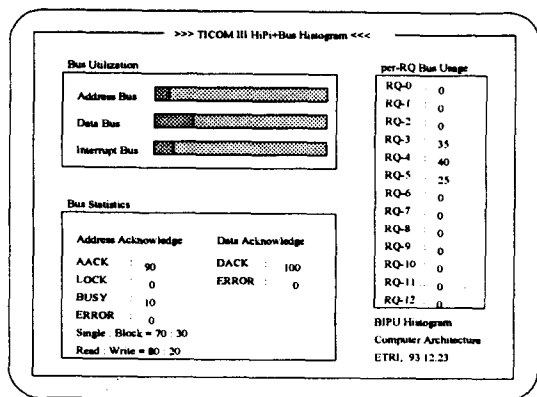


그림 6. Histogram 메뉴

Fig. 6. Histogram menu of the BIPU.

<그림 6>은 어드레스 버스 이용율, 데이터 버스 이용율, 그리고 인터럽트 버스의 이용율과 각 보드의 버

스 이용율, 읽기 동작과 쓰기 동작의 비율등을 출력하는 히스토그램 화면을 보여준다.

VI. 결 론

버스 정보 처리기 설계 및 구현의 기본 목적은 시스템 통합 시험 지원과 시스템 성능 분석 지원도구로써 시스템 버스상에 구동된 데이터를 정확히 저장 검색하여 처리한 결과를 사용자에게 제공하는데 있다.

본 논문의 버스 정보 처리기는 시스템 버스 검색 기능과 시스템 버스 성능 분석 기능, 그리고 응답기 데이터 전송 기능등을 제공하며, 이를 위해 첫째, 다양한 트리거 조건을 제공하여 시스템 버스에 구동되는 신호들의 의미있는 대부분의 조합을 가지고 다양한 환경에서의 시스템 동작 상태를 알 수 있도록 설계 및 구현되었으며, 둘째, 많은 양의 트리거 데이터를 검색할 수 있도록 충분히 큰 용량의 검색 메모리를 제공하여 구현하였고 셋째, 트리거 조건을 만족하는 신호 그룹을 저장하는 방법을 가변적으로 하여 검색 메모리의 효율성을 높이고 사용자의 다양한 욕구를 충족시켜 보다 정확한 시스템 동작을 예측 가능하도록 설계 및 구현하였다.

또한, 검색된 데이터를 처리하여 기본적인 버스 이용율 및 각 보드의 버스 사용율, 읽기 동작과 쓰기 동작의 비율등 시스템 버스의 성능을 쉽게 알 수 있도록 성능 분석 기능을 구현하였으며,

응답기 모듈을 제공하여 시스템 버스를 통한 인터페이스가 가능하도록 구현하였다.

버스 정보 처리기 보드 시험과 시스템 통합 시험시 실제 활용을 통하여 응답기 데이터 전송 기능을 제외한 모든 기능이 정상적으로 동작됨을 확인하였으며, 공동 연구 업체에서 시스템 통합 시험을 위하여 계속 사용하고 있다. 응답기 데이터 전송 기능은 요청기와 연동하여 추후 시험될 예정이다.

향후 고속 중형 컴퓨터의 시스템 통합 시험이나 시스템 버스 성능 분석시에 높은 신뢰성을 가지고 동작되는 버스 정보 처리기를 이용하여 유용한 결과를 얻을 수 있을 것으로 생각된다.

참 고 문 헌

[1] 심원세의 2인, "HiPi+Bus에서 사용된 데이

- 터 전송제어 기법'', 한국전자공학회 학술대회 논문집, July 1994, pp 623 - 625
- [2] 신상석외 6인, ``고속중형 컴퓨터(주전산기 III) 시스템 설계'', 한국정보과학회 춘계 학술대회 논문집, April 1993, pp 255 - 258
- [3] 신상석외 4인, ``VME-BSA(VME Bus State Analyzer) 개발에 관한 연구'', 한국전자공학회 학술대회 논문집, July 1987, pp 1111 - 1115
- [4] WonSae Sim et. al., ``Implementation of Bus Status Analyzer on the Bus-based Multiprocessor System'', *Poceedings of the JTC-CSCC' 91*, Hiroshima, Japan1991, pp 427 - 432.
- [5] Cadence, Design Framework II User's Manual, Cadence Design System Inc., April 1993.
- [6] Altera, *MAX+plusII User's Manual*, Altera Corporation, April 1991.
- [7] National, *High-Performance Bus Interface Designer's Guide*, National Semiconductor Corporation, 1992.

저 자 소 개



韓宗錫(正會員)

1966년 7월 13일생. 1989년 2월 아주대학교 전자계산학과 학사. 1990년 3월 - 1992년 2월 연세대학교 본대학원 전산과학과 석사. 1992년 2월 - 현재 한국전자통신연구소 프로세서연구실 근무. 관심분야는 병렬처리 컴퓨터구조, 상호연결망, ASIC 설계 등임.

국전자통신연구소 프로세서연구실 근무. 관심분야는 병렬처리 컴퓨터구조, 상호연결망, ASIC 설계 등임.



宋庸鎬(正會員)

1989년 서울대학교 공과대학 컴퓨터공학과 학사. 1991년 서울대학교 공과대학 컴퓨터공학과 석사. 1991년~현재 삼성전자 정보컴퓨터 전임연구원. 관심분야는 Computer Architecture, Performance Evaluation 등임.

컴퓨터 전임연구원. 관심분야는 Computer Architecture, Performance Evaluation 등임.