TMS320C6701기반의 고속 병렬신호처리보드의 설계 및 구현

김진호*, 전창호*, 박성주*, 이동호*
(* : 한양대학교 전자컴퓨터 공학부)

An Implementation of a High Speed Parallel DSP Boards using TMS320C6701

J. H. Kim*, C. H. Jeon*, S. J. Park*, D. H. Lee*
(* : School of Electrical and Computer Engineering, Hanyang Univ.)

{jhkim, dhlee}@image.hanyang.ac.kr, chjeon@cse.hanyang.ac.kr, parksj@mslab.hanyang.ac.kr

요 약

근본적으로 방대한 양의 실시간 연산을 요구하는 영상 신호처리, 소나, 레이다와 같은 시스템에서는 시스템의 성능을 최대화하기 위해 병렬 신호처리 시스템의 사용이 불가피하다. 본 논문은 방대한 양의 데이터를 실시간으로 처리할수 있는 병렬 신호처리보드를 설계 및 구현하였다. 이 보드는 DSP칩간의 통신과 보드간의 통신이 가능하며, DSP칩이 마스터가 되어 EMIF(External Memory Interface Port)포트를 통해 다른 DSP칩의 지역메모리를 액세스 할수 있다. 또한 외부의 호스트 프로세서가 보드 내의 DSP칩에 프로그램을 다운로딩 할 수 있다. 보드간의 통신은 PCI 버스를 통하여 이루어지며, DSP칩간의 통신과 DSP칩과 그의 지역메모리와의 통신은 지역버스를 통해 직접적으로 이루어진다. 보드에서 가장 핵심인 DSP-to-PCI제어기는 하드웨어 언어인 VHDL로 설계하였으며, 시뮬레이션 환경은 Synopsys & ALTERA MaxplusII를 사용하여 검증하였으며, 최종적으로 CPLD(Complex Programable Logic Device)칩을 사용하여 구현하였다.

I. 서 론

최근에는 고속 신호처리 시스템을 구현하는데 DSP칩을 다중으로 연결하는 병렬 시스템 구조가 많이 채택되고 있다. 그 이유는 DSP소자들의 개별적인 연산능력이 우수하긴 해도 단일 프로세서구조로 높은 성능요구조건을 충족시키기 어렵기 때문이다. 특히 영상 신호처리, 소나, 레이더와 같은 신호처리분야를 구현하기 위해서는 몇 GFLOPS(Giga Floating Point Operation Per Second)의 시스템 신호처리 성능을 요구하기 때문에 필연적으로 DSP의 병렬 신호처리가 필수적이며, 이에 관하여 현재까지 많은 병렬 신호처리 연구와 병렬 신호처리 시스템이 개발되어져 왔다[12]. 본 논문에서는 고속으로 데이터신호를 처리하기 위해 2개의 DSP칩을 사용하여 병렬 신호처리보드를 설계 및 구현하였다. 본 논문의 구성은 구현된 병렬 신호처리보드의 구성과 각 블록에서의 역할을 2장에서 상세히 설명하였고, 3장에서는 CPLD칩으로 구현한 DSP-to-PCI제어기에 대하여 설명하였으며, 4장에서 본 논문의 결론을 맺는다.

II. 병렬 신호처리보드의 구성

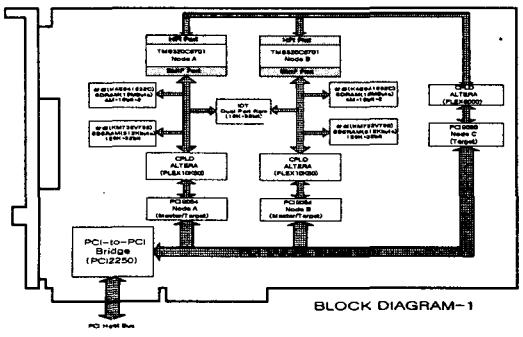

그림 1. 병렬 신호처리보드 전체 블록도

<그림 1>은 구현된 병렬 신호처리보드의 전체 하드웨어 구성을 나타내는 블록도이다. 병렬 신호처리보드는 크게 DSP신호처리부, 지역메모리부, 공유메모리부, PCI 인터페이스부로 구성되어 있다. DSP를 통하여 고속으로 처리된 신호는 DSP의 EMIF포트를 통해 지역메모리인 SDRAM, SBSRAM에 임시 저장되며, PCI버스를 통해 저장된 데이터를 호스트로 보낸다. 보드간의 통신은 PCI버스를 통하여 이루어지고 프로세서간의 통신은 이중포트 메모리를 통하여 지역버스에서 직접적으로 이

루어진다[9][15]. 중요부분의 자세한 설명은 다음과 같다.

1. DSP신호처리부(TMS320C6701)

병렬 신호처리보드의 연산소자로 TI사의 TMS320C6701은 고성능의 부동소숫점 DSP칩으로서 167MHz의 클럭속도와 1GFLOPS의 산술연산능력을 가지고 있다. 그리고 1Mbit의 프로세서 내부메모리와 초당 400Mbyte의 대역폭을 가지며, 32비트 외부 메모리 인터페이스를 지원한다. EMIF포트를 통하여 직접적으로 지역메모리(SDRAM, SBSRAM)를 액세스할 수 있으며, DSP가 마스터일 때 다른 DSP칩의 지역메모리를 액세스할 수 있다. 또한 HPI(Host Port Interface)포트를 통하여 외부의 호스트가 DSP에 프로그램을 다운로딩할 수 있다[1][3].

2. 지역메모리부(SDRAM, SBSRAM)

지역메모리인 SDRAM과 SBSRAM은 DSP에서 고속으로 처리된 데이터를 임시 저장하기 위해 사용된다. 평상시에는 DSP가 SDRAM과 SBSRAM을 직접적으로 제어하다가 외부에서 SDRAM이나 SBSRAM을 사용하겠다는 홀드 요구신호가 들어오면 DSP는 지역메모리와의 연결을 끊고 이때부터 DSP-to-PCI제어기가 지역메모리를 제어한다[4][5].

3. 공유메모리부(Dual Port Ram)

이중포트메모리는 두 DSP칩의 공유메모리로 사용되며, 동시에 메모리 접근이 가능한 장점이 있다. 따라서 병렬 신호처리보드에서 DSP칩간의 통신을 PCI버스를 거치지 않고 지역버스에서 직접적으로 할수 있다[14].

4. PCI인터페이스부

PCI버스를 통한 DSP칩간 또는 보드간 DMA전송을 지원하며, PCI버스 마스터가 DSP칩의 지역메모리를 액세스할수 있다. 또한 시스템 호스트 프로세서가 PCI버스를 통하여 DSP에 프로그램을 다운로딩할수 있다. PCI버스는 33MHz로 동작되며, 32bit 데이터 버스폭을 지원하므로, 이론적으로 최대 132MB/S를 전송할수 있다[10][11][13].

III. DSP-to-PCI 제어기

DSP-to-PCI제어기는 크게 마스터모드, 타겟모드, 그리고, HPI(Host Port Interface)모드로 구성되며, CPLD(Complex Programmable Logic Device)칩으로 구현하였다[8].

1. 마스터 모드

지역호스트 프로세서인 DSP가 시스템 호스트 프로세서와는 독립적으로 다른 PCI타겟에 대하여 데이터를 송수신할수 있도록 하며, 이를 위해서 마스터용 DMA 제어기가 포함된다.

(1) DMA에 필요한 레지스터들

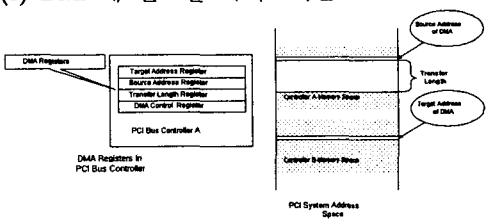

그림 2. DMA 관련 레지스터 및 메모리 영역과의 관계

PCI브릿지인 PCI9054는 DSP가 직접 DMA전송을 개시할수 있도록 여러개의 레지스터를 구비하고 있다. 일반적으로 DMA를 구현하기 위해서는 다음과 같은 값을 셋팅하는 것이 일반적이다[1][6].

- Target Address Register : 이 값은 현재 마스터인 DSP가 액세스하려는 PCI버스상의 타겟 디바이스의 주소를 말한다.
- Source Address Register: 이 값은 현재 마스터 프로세서의 메모리 영역의 액세스 대상 시작 주소를 알려준다.
- 전송 길이 레지스터 : 이 값은 전송하려는 데이터의 총 크기를 나타낸다.
- DMA 콘트롤 레지스터 : 일반적인 DMA제어를위한 레지스터이다. 인터럽트 혹은 시스템에 의한 DMA의 개시와 같은 기능을 포함한다.

지역 호스트인 DSP가 직접 DMA기능을 수행한다는 것은 먼저 PCI9054가 지원하고 있는 위 레지스터를 직접 세팅한다는 것을 의미한다. 물론 위 레지스터들은 시스템 호스트 프로세서에 의해서도 설정이 가능하다. 위의 레지스터 세팅이 완료되면 DMA 전송을 개시할 수 있다. 개시 역시 시스템 프로세서를 이용하며 소프트웨어적으로 DMA 콘트롤 레지스터의 DMA시작 제어 비트를 세팅함으로써 개시할 수 있다. 하드웨어적인 방법으로는 DSP가 직접 PCI9054의 외부 핀으로 나와 있는 DMA요구 핀을 이용하여 직접 DMA를 개시할 수 있다. <그림 2>는 DMA관련 레지스터와 그내용을 간단히 보여주고 있다.

(2) DMA를 위한 PCI9054와 DSP와의 인터페이스

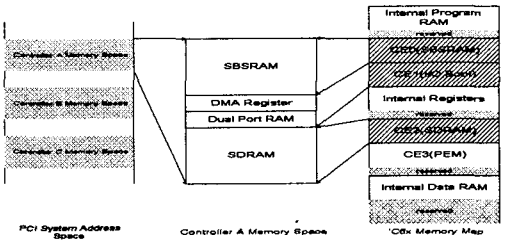

그림 3. 시스템과 콘트롤러 그리고 TMS320C6701의 메모리 맵 예

DSP에 의한 PCI9054의 레지스터 엑세스는 간단히 이루어 질 수 있다. 먼저 PCI9054측에서는 CCS#라는 외부 입력핀을 이용하여 외부 호스트가 PCI9054의 내부 레지스터를 엑세스한다는 것을 감지한다. DSP는 DMA 관련 레지스터를 엑세스 하기 위해서 이들 레지스터를 자신의 메모리 맵에 할당 하여야 한다. <그림 3>에서처럼 DSP의 메모리 맵은 그 용도가 미리 정해져 있으며 외부 메모리 영역은 칩 인에이블 신호인 CE0~CE3을 이용하여 액세스하여야 한다. 예를 들어 SDRAM은 CE2가 활성화되는 영역에 매핑되어 있다. 따라서 우리가 엑세스하려하는 PCI9054의 내부 DMA관련 레지스터는 CE1 영역을 통해 액세스한다[6].

(3) PCI9054와 DSP간 DMA 전송과정

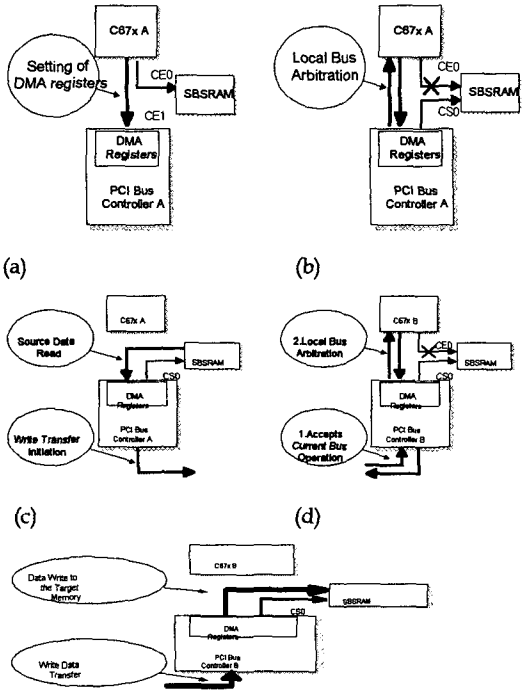

그림 4. TMS320C6701에 의한 DMA전송과정

<그림 4>는 DSP가 지역 호스트로써 직접 DMA전송을 처리하는 과정을 나타내고 있다. <그림 4> (a)는 앞에서 설명한 PCI9054의 DMA관련 레지스터를 DSP가 설정하는 과정이다. 이 때 DSP는 레지스터를 엑세스 하기 위하여 CE1과 함께 레지스터의 주소를 이용한다. 레지스터 세팅이 끝나면 이 후의 모든 데이터 전송 문제는 PCI9054가 관리한다. 이를 위해선 먼저 컨트롤러는 DSP로부터 지역버스를 사용하기 위한 사용권을 획득하여야 한다. <그림 4>의 (b)에서처럼 이 과정에는 홀드 및 홀드응답신호가 이용된다. 버스 사용권을 넘겨준 DSP는 EMIF의 출력 신호들을 모두 Hi-Z상태로 만들어 준다. 따라서 이 후부터 PCI9054는 자신의 주소와 데이터 버스 및 기타 제어선을 이용하여 지역 메모리를 엑세스할 수 있다. <그림 4>의 (c)는 PCI9054가 직접 지역 메모리로부터 데이터를 읽어들이는 과정이다. 이와 동시에 PCI9054는 PCI버스 사용권을 중앙 아비터로부터 할당 받는다. (a)부터 (c)까지는 TMS320C67x에 의한 DMA 쓰기 과정을 나타내었으며 (d)와 (e)는 반대로 읽기 과정을 보여주고 있다.

2. 타겟 모드

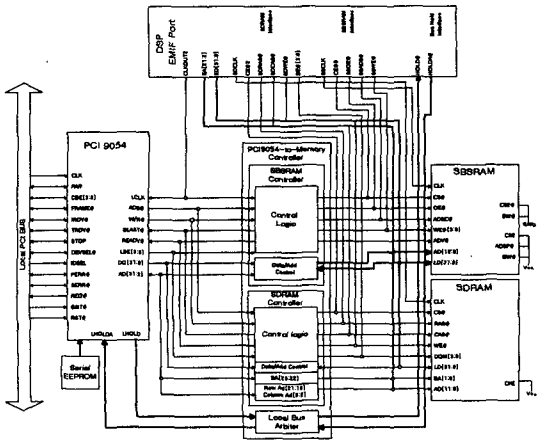

그림 5. PCI9054-to-EMIF인터페이스 제어기(타겟모드)

타겟모드일때는 PCI버스상의 외부의 마스터 디바이스가 DSP의 지역메모리인 SDRAM과 SBSRAM을 액세스 할 수 있도록 동작한다. <그림 5>는 타겟모드로 PCI9054-to-EMIF 인터페이스에 대한 전체구조를 나타내었다. PCI9054가 DSP칩의 지역메모리를 액세스하기 위해서는 우선적으로 홀드신호를 이용하여 지역버스의 사용권을 DSP의 EMIF에게 요청해야 한다. 이 홀드신호가 제어기에 인가되면 제어기는 이 신호를 다시 인버트 시켜서 EMIF포트에 보내는데, DSP는 내부에서 정해져 있는 메모리 요구의 우선순위 에 따라서 지역버스의 사용권한을 가장 우선순위에 있는 PCI9054에게 준다. 이때 EMIF포트에 나와 있는 모든 신호들은 Hi-Z 상태로 되며(단 클럭은 제외) PCI9054는 지역버스 사용을 허가하는 홀드응답신호를 받고 난 다음부터야 비로서 지역버스를 사용할수 있다. 지역버스를 점유하는 동안은 PCI9054의 홀드신호가 High상태에 있으며, 이에 따라 EMIF포트로 들어가는 홀드신호는 Low상태로 있게 되며, 이 상태가 계속 지속되는 동안은 PCI9054가 지역버스를 계속 사용할 수 있다. <그림 5>에서는 PCI9054가 SDRAM과 SBSRAM을 모두 액세스할 수 있도록 제어기가 설계되어 있는 것을볼수 있다. 이 2개의 지역메모리를 액세스하기 위해서 PCI-to-SDRAM 제어기와 PCI-to-SBSRAM 제어기가 포함되어 있으며, 먼저 SDRAM과 SBSRAM 둘중에 어느것을 액세스할지를 결정하기 위해서 PCI9054에서 각 메모리에 할당된 주소를 제어기가 판단하여 SDRAM과 SBSRAM의 CS신호를 Low로 인가한다. 예를 들어 PCI9054가 SDRAM을 액세스하고 싶다면 SDRAM에 대하여 PCI9054에 할당된 주소를 제어기에 보내고, 이를 받은 제어기는 이 주소를 해독하여 SDRAM에 CS신호를 Low로

인가하고, 동시에 SBSRAM에게 CS신호를 High로 인가하여 SBSRAM의 모든 신호를 Hi-Z로 만들어서 SDRAM을 액세스할 수 있도록 한다[6][8].

3. HPI(Host Port Interface)모드

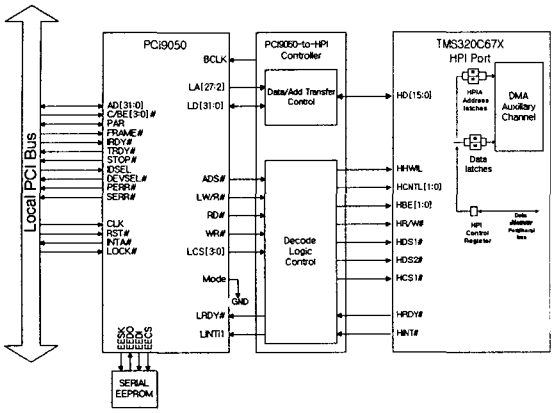

그림 6. PCI9050-to-HPI 인터페이스 제어기(HPI 모드)

DSP칩의 HPI 포트는 시스템 호스트 프로세서가 DSP에게 프로그램을 다운로딩할수 있도록 지원한다. 또한 호스트 프로세서는 HPI를 통해 DSP칩의 전체 메모리를 액세스할 수 있다. 여기에 PCI9050은 총 4개까지의 DSP칩의 HPI 포트를 PCI버스와 연결해주는 PCI브릿지의 역할을 한다. DSP-to-PCI 제어기는 PCI9050과 HPI 포트간의 원할한 인터페이스를 위해서 HPI모드를 지원한다[1][7]. <그림 6>은 PCI9050-to-HPI 인터페이스 제어기에 대한 전체블럭도를 나타내었다. PCI9050-to-HPI 인터페이스 제어기는 DSP의 전체 메모리를 액세스할 수 있도록 동작한다. PCI9050에서 나오는 2개의 CS#[1:0](Chip Select)신호는 2개의 DSP칩중에 어느 것을 선택할지를 결정하고, CS#신호가 Low로 떨어지면 (예를 들어 CS# "10"=>1번 DSP칩의 HPI 포트와 연결) 연결된 DSP칩을 제외한 다른 DSP칩의 HPI는 Hi-Z상태가 된다. HPI 포트의 특성상 주소와 데이터가 16비트 버스로 다중화되어 있기 때문에 PCI9050에서 나오는 주소[27:2]와 데이터[31:0]를 제어기가 받아서 16비트씩 두번 연속해서 보내주어야 한다. 또한 여기서 고려해야 할 점은 PCI9050은 클럭의 동기에 맞추어 동작되는 반면 HPI 포트로 들어가는 신호는 비동기로 동작되기 때문에 데이터의 읽기/쓰기를 제어신호 (HSTROBE)에 의해서 동작된다는 점을 고려해야한다[1][7].

IV. 결 론

본 논문에서는 고속의 신호처리를 위하여 2개의 DSP칩(TMS320C6701)을 사용하여 병렬 신호처리보드를 설계 및 구현하였다. DSP-to-PCI제어기는 하드웨어 언어인 VHDL로 설계하였으며, 시뮬레이션 환경은 Synopsys & ALTERA MaxplusII를 사용하여 검증하였으며, 최종적으로 CPLD칩을 사용하여 구현하였다. DSP-to-PCI제어기는 PCI버스환경에서 동작하는 DSP기반의 병렬 신호처리시스템에서 유용하게 쓰일것으로 기대된다.

참 고 문 헌

[1] TMS320C6701 Peripherals Reference Guide Texas Instruments, 1998.

[2] PCI2250 PCI-to-PCI Bridge User Guide Texas Instruments. 1999.

[3] TMS320C6X Evaulation Module Technical Reference Texas Instruments, 1999.

[4] K4S641632D 1M*16bit*4bank CMOS SDRAM Datasheet Samsung.

[5] KM732V789 128K*32bit SBSRAM Datasheet Samsung.

[6] PLX Technology PCI9054 Data Book.

[7] PLX Technology PCI9050 Data Book.

[8] ALTERA FLEX10K50, FLEX6000 Data Book

[9] Spectrum Signal "Datona" Dual C6701 PCI Board Technical Reference.

[10] PCI System Architecture Fourth Edition, MindShare ,INC

[11] PCI Specification Revision 2.1, PCI Special Interest Group, 1997

[12] 박광철, 이동호 "TMS320C67x 기반 병렬시스템의 설계 및 성능분석" 98"추계 신호처리학술발표논문집.

[13] 정수운, 이동호 "VHDL을 이용한 PCI로컬버스 제어기의 설계" 98"추계 신호처리학술발표논문집.

[14] IDT70V26S/L 256K(16K*16) 3.3V Dual-Port RAM Datasheet.

[15] Mario Palumbo, "A New Standard for Multiprocessing DSP Systems", Spectrum Signal Processing