2014년 4월 전자공학회 논문지 제 51권 제 4 호
Journal of The Institute of Electronics and Information Engineers Vol. 51, NO. 4, April 2014 http://dx.doi.org/10.5573/ieie.2014.51.4.033

논문 2014-51-4-5

효과적인 다채널 직접 메모리 접근 관리를 통한 멀티포트 메모리 컨트롤러의 성능 향상 방법

(Performance Improvement Method of Multi-Port Memory Controller Using An Effective Multi-Channel Direct memory Access Management)

천 익 재*, 여 준 기*, 노 태 문**, 이 문 식*

(Ik-Jae Chun©, Chun-Gi Lyuh, Tae Moon Roh, and Moon-Sik Lee)

요 약

본 논문에서는 모바일 시스템 환경에서 멀티포트 메모리 컨트롤러의 특성을 고려한 직접 메모리 접근 컨트롤러를 사용하여 고속 데이터 전송을 효과적으로 수행하는 메모리 액세스 방법을 보인다. 제안된 직접 메모리 접근 컨트롤러는 여러 개의 직접 메모리 접근 채널을 제어 할 수 있는 통합 채널 관리 기능을 제공하며, 그 채널들은 물리적으로 분리되어 서로 독립적으로 동작한다. 제안된 직접 메모리 접근 방법을 통한 데이터 전송을 이용함으로써 읽기 동작에 대하여 72%, 쓰기 동작에 대하여 69%의 데이터 전송 성능 향상을 얻었다. 특히, 4 채널 접근 모드에 대해서 제안된 방법이 기존 직접 메모리 접근 방법에 비하여 63% 적은 전체 전송 사이클을 가짐으로써 전송 성능 향상에 기여할 수 있음을 보인다.

Abstract

This paper presents an effective memory access method for a high-speed data transfer on mobile systems using a direct memory access controller that considers the characteristics of a multi-port memory controller. The direct memory access controller has an integrated channel management function to control multiple direct memory access channels. The channels are physically separated and operate independently from each other. Experimental results show that the proposed direct memory access method improves the transfer performance by up to 72% and 69% on read and write transfer cycles, respectively. The total number of transfer cycles of the proposed method is 63% less than in a commercial method under 4-channel access.

Keywords : High-Speed Data Transfer, DMA Control, Memory Controller, Mobile SoC

Ⅰ. 서 론

오늘날 대부분의 모바일 단말 및 휴대용 멀티미디어 장치에서 사용되는 시스템 온 칩(system on chip: SoC)들은 마이크로프로세서, 메모리 모듈, 주변기기, 사용자 정의 IP (intellectual property) 들과 같은 다양한 장치들로 구성된다. 이러한 장치들은 서로 다른 장치들을

* 정회원, ** 평생회원, 한국전자통신연구원
(Electronics and Telecommunications Research Institute)
© Corresponding Author(E-mail: ijchun@etri.re.kr)
※ 본 연구는 미래부가 지원한 2013년 정보통신·방송 (ICT) 연구개발사업의 연구결과로 수행되었음.
접수일자: 2013년9월11일, 수정일자: 2014년3월13일
수정완료: 2014년4월3일

제어하고 데이터를 전송하기 위해 상호 유기적으로 연결된다. 이를 위해 대부분의 SoC 들은 일반적으로 AMBA AHB/APB[1] 같은 버스 인터커넥션을 채택하고 있다. 그러나 AMBA AXI[2]와 같은 네트워크 기반의 인터커넥션이 도입되면서, 버스 및 네트워크 인터커넥션 뿐만 아니라 다른 인터페이스를 가지는 장치들이 하나의 시스템으로 혼합되고 있다. 따라서 SoC를 구성하는 장치들 간의 연결은 점점 더 복잡해지고 있으며 그들 간의 데이터 전송은 더 어려워지고 있다.

이러한 SoC들은 다양한 유형의 대용량 데이터를 빠르게 처리하는데 초점을 맞추고 있으며, 처리해야할 데이터의 크기는 점점 더 증가하고 있다. 시스템 성능의 측면에서 외부 메모리와 내부 장치 간의 고속 데이터 전송은 빠른 데이터 처리를 위해서 매우 중요한 요소로 나타난다. 이 때 데이터 전송의 처리량은 직접 메모리 액세스(direct memory access: DMA) 컨트롤러와 상호 프로토콜의 성능에 가장 크게 의존하게 된다.

지금까지 DMA 컨트롤러에 대한 신속하고 효율적인 데이터 전송에 대한 여러 연구가 진행되어왔다[3~7]. C.-H. Yu 등은 사용자가 구성한 X와 Y 위치 정보를 실제 물리적인 데이터 주소로 변환하는 2D coordination setting interface를 제안 하였다[3]. G. Ma 와 H. He는 블록 데이터 이동을 수행하는 매개 변수 집합을 사용하여 병렬 AHB/APB 동작 및 3D 전송 방법을 제안하였고[4], K.-J. Lin 등은 실시간 I / O 처리량을 보장하고 CPU (control processing unit)에 걸리는 인터럽트를 줄일 수 있는 일정관리가 가능한 DMA 방법을 제시하였다[5]. Texas Instrument에서는 랜덤 액세스 메모리(RAM)와의 연결 방법을 통해 전송 요청의 우선순위를 정하고 우선순위가 높은 요청을 잘 처리할 수 있도록 하는 DMA 컨트롤러를 제안하였다[6]. 또한, S. Srinivasan과 D. B. Stewart는 DMA 동작이 연속하지 않는 소스 주소 또는 대상 주소에도 불구하고 데이터 이동을 계속할 수 있도록 미리 프로그램 된 DMA 방법을 제안하였다[7].

위에서 보인 연구들은 타겟 시스템에서 구현된 이미지나 영상 처리와 같은 각 어플리케이션의 데이터 구조를 고려하거나, 또는 채널 전송 및 DMA 컨트롤러 자체의 제어 방법 개선에 초점을 맞추고 있다. 따라서 전자의 DMA 컨트롤러들은 읽기/쓰기 등의 패턴 변화 및 주소 변환 등을 통한 데이터의 효율적인 전송 방법에 중점을 두고 있으며, 후자의 DMA 컨트롤러들은 DMA 채널을 CPU 또는 사전 정의 된 매개 변수에 의해 개별적으로 제어하여 제어 부하를 줄이는데 중점을 두고 있다. 그러나 시스템에 있어서 메모리 컨트롤러 또한 데이터 전송에 있어서 매우 중요한 역할을 수행하고 있지만, 지금까지 DMA 컨트롤러에 대한 연구는 메모리 컨트롤러 자체의 액세스 특성을 고려하지 않고 있다.

본 논문에서는 물리적으로 분리된 DMA 채널을 통해 메모리 컨트롤러의 메모리 액세스 성능을 향상시킬 수 있는 간단하고 효율적인 채널 제어 방식과 다채널 DMA 방법을 제안한다. 제안된 방법은 직접 외부 메모리를 제어하고 멀티 프로토콜 인터커넥션 및 다중 인터페이스 포트를 제공하는 메모리 컨트롤러의 특성을 고려한다. 본 논문에서는 이러한 메모리 컨트롤러를 멀티 프로토콜 기반 멀티포트 메모리 컨트롤러 (multi-port memory controller: MPMC)라 부르기로 한다. 우리는 시스템의 데이터 전송 성능이 시스템을 구성하는 버스 또는 네트워크와 같은 인터커넥션의 프로토콜 유형과 밀접하게 연관되어 있다는 것을 알고 있다. 따라서 본 논문에서는 MPMC를 구성하는 서로 다른 상호 연결 프로토콜에 의해 발생되는 데이터 전송 차단 효과를 줄임으로써 데이터 전송 성능을 향상시킬 수 있는 DMA 채널 제어 방법을 보이고자 한다. 제안 된 방법은 MPMC 자체의 수정없이 DMA 채널을 통한 MPMC의 멀티포트의 효율적인 제어를 통해 메모리 버스 활용도를 극대화 할 수 있다.

본 논문은 다음과 같이 구성된다. Ⅱ장에서 MPMC의 기본적인 특성을 보이고, Ⅲ장에서 제안된 다채널 DMA 컨트롤러의 구조를 보이고 Ⅳ장에서 제안된 DMA 컨트롤러의 다채널 전송방법에 대하여 설명한다. Ⅴ장에서는 실험결과를 보이고, 마지막으로 Ⅵ장에서 결론을 맺도록 한다.

Ⅱ. MPMC의 기본 특성

일반적으로 디지털 시스템의 데이터 전송에 사용되는 버스 기반의 인터커넥션을 가지는 SoC에 있어서, 마이크로프로세서, 액정 표시 컨트롤러 및 스트림 처리 모듈과 같은 여러 개의 마스터 장치들은 각 장치들이 요구하는 서비스 품질(quality of service: QoS)을 얻기 위해 메모리 액세스를 동시에 요구할 지도 모른다. 이

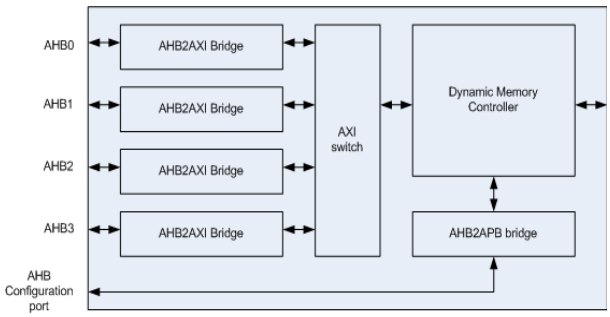

그림 1. MPMC의 구조 예
Fig. 1. An example of an MPMC.

를 위해 각 마스터 장치들이 연결된 인터커넥션은 각 장치들을 중재하여 메모리가 제공할 수 있는 전송 대역을 알맞게 할당해야 한다. 이를 위해 메모리 컨트롤러는 외부 메모리를 액세스하는 각 마스터 장치에 일정하고 안정된 서비스 품질을 제공하기 위해, 멀티포트 인터페이스와 함께 메모리 접근 권한을 제어할 수 있는 기능을 갖기도 한다[8~9]. 이러한 메모리 컨트롤러들은 내부적으로 서로 다른 두 개의 인터커넥션 프로토콜을 사용하고 있으며, 다수의 인터페이스 포트를 사용한다. 우리는 이러한 메모리 컨트롤러를 MPMC라 하고 그 예를 그림 1에서 보인다.

그림 1에서 보이는 바와 같이 MPMC는 크게 멀티 프로토콜 데이터 전송 모듈과 동적 메모리 컨트롤러로 구성되어 있다. 멀티 프로토콜 데이터 전송 모듈은 버스 기반의 인터페이스와 네트워크 기반의 중재기를 기반으로 하고, 동적 메모리 컨트롤러는 외부 동적 메모리 모듈을 제어한다. 버스 기반의 인터페이스의 경우 AHB2AXI 브릿지[1~2]가 사용되며 네트워크 기반의 중재기로는 AXI 스위치[2]가 일반적으로 사용된다. 마스터 장치에 의한 메모리 액세스 정보는 버스 기반의 인터페이스에서 동적 메모리 컨트롤러에 네트워크 기반의 중재기를 통해서 전송된다. 이때 버스 기반의 인터페이스는 네트워크 기반의 중재기에 마스터 장치의 주소, 데이터 및 버스트 모드를 포함하여 액세스 정보를 전달한다. 이러한 전송에서 버스 기반의 인터페이스는 현재의 데이터 전송이 완료될 때까지 다음 액세스 정보 전송을 차단하는 특성을 보이게 된다. 그러나 네트워크 기반의 중재기는 현재 데이터 전송 상태와 관계없이 버스 기반의 인터페이스에서 동시에 수신한 액세스 정보를 순차적으로 동적 메모리 컨트롤러에 전송할 수 있다. 동적 메모리 컨트롤러는 외부 동적 메모리와의 데이터 통신을 수행하며, 이때 동적 메모리 컨트롤러가 현재 데이터 전송이 끝나기 전에 외부 메모리에 다음 액세스 정보를 제공하는 경우 현재 데이터 전송이 완료되면, 외부 메모리는 곧바로 다음 데이터를 전송할 수 있게 된다. 이러한 동작형태는 외부 메모리의 데이터 대역폭 즉 메모리 버스의 사용률을 증가시키는 효과를 제공한다.

위에 설명한 바와 같은 이종 인터페이스가 동시에 적용된 메모리 컨트롤러에 있어서 버스 기반의 인터페이스와 네트워크 기반의 인터페이스 간의 프로토콜 불일치는 버스 기반 인터페이스가 다음 액세스 정보의 전달을 막음으로써 동적 메모리 컨트롤러가 외부 메모리에 다음 액세스 정보를 미리 제공하지 못하도록 한다. 이러한 액세스 정보의 블로킹 문제는 전송 데이터의 대역폭을 저하시키며, 메모리 컨트롤러의 데이터 전송 성능이 네트워크 기반의 중재기에 의해서가 아니라 버스 기반의 인터페이스 특성에 더 크게 의존하도록 한다. 이러한 현상은 멀티채널 액세스에 비해 단일 채널 액세스에서 더 분명히 나타나며, 우리가 사용하고자 하는 멀티포트를 갖는 메모리 컨트롤러에 있어서 다른 액세스 포트를 사용하지 않는 상황에서 단일 채널 액세스를 사용하여 point-to-point 전송을 하는 경우에도 데이터 대역폭을 늘릴 수 없는 원인이 된다.

그러나 메모리 컨트롤러의 멀티포트를 통해 동시에 제공되는 버스 기반 인터페이스의 모든 액세스 정보를 순차적으로 네트워크 기반의 중재기를 통해 동적 메모리 컨트롤러에 제공하게 되면, 동적 메모리 컨트롤러가 현재 데이터 전송이 끝나기 전에 외부 메모리에 순차적으로 다음 액세스 정보를 전송할 수 있게 된다. 이를 통해 우리는 MPMC와 외부 메모리 간의 메모리 버스의 대역폭을 증가시킬 수 있으며, 본 논문은 이러한 특성을 제안된 DMA 컨트롤러를 통해 반영하고자 하였다.

그림 2와 3은 단일 채널 액세스와 4 채널 액세스에 있어서의 4 포트 MPMC의 전송 특성을 보인다. 그림에서 보이는 바와 같이 외부 동적 메모리에 현재의 전송 동작이 끝나기 전에 다음 액세스 정보를 전달 할 수 있다면 MPMC와 외부 동적 메모리 사이의 버스 사용률과 데이터 대역폭을 증가시킬 수 있음을 알 수 있다. 이때 연속적인 액세스 정보의 전달은 그림 3에서와 같이 MPMC의 다른 포트들에 접근함으로써 얻어낼 수 있다.

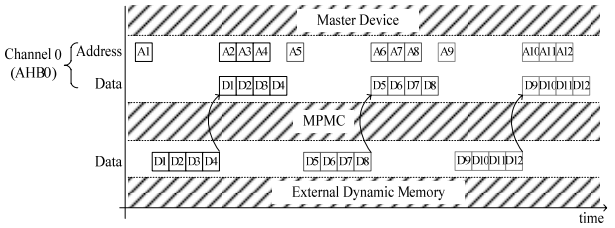

그림 2. 4 포트 MPMC의 단일 채널 액세스의 데이터 읽기 특성

Fig. 2. Data read characteristics of single-channel access for a four-port MPMC.

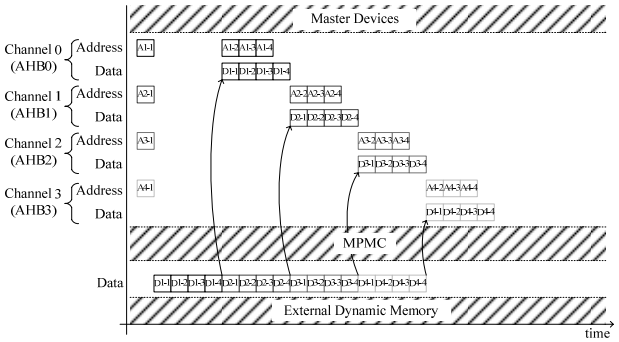

그림 3. 4 포트 MPMC의 4 채널 액세스의 데이터 읽기 특성

Fig. 3. Data read characteristics of four-channel access for a four-port MPMC.

이것은 우리가 외부 메모리와 장치 간 데이터 전송에 있어서 MPMC의 모든 사용 가능한 포트들을 사용할 때 메모리의 데이터 대역폭과 버스 사용률을 높일 수 있다는 것을 의미한다. 또한 사용 가능한 메모리 컨트롤러의 다른 포트들을 활용하게 되면 메모리 컨트롤러의 단일 포트를 사용하여 point-to-point 데이터 전송을 수행할 때보다 더 효과적이고 효율적인 데이터 전송이 가능함을 알 수 있다.

Ⅲ. 다채널 DMA 컨트롤러 구조

시스템에서의 메모리 컨트롤러 및 주변 장치 사이에서 발생되는 데이터 전송에 있어서 그 성능은 매우 중요한 문제 중 하나이다. 그 데이터 전송 역할은 일반적 DMA 컨트롤러의 DMA 채널을 통해 달성된다. 본 논문에서는 point-to-point 데이터 전송에 있어서의 성능 향상을 위해서 MPMC의 멀티 포트를 제어하는 다채널 DMA 컨트롤러를 제안한다.

그림 3에 나타낸 바와 같이, 우리는 MPMC와 내부 주변장치 사이의 point-to-point 데이터 전송에 있어서 DMA 컨트롤러가 MPMC에 접근 가능한 액세스 채널을 최대한 사용할 때 데이터 전송 대역폭을 증가시킬 수 있음을 알고 있다. 따라서 제안된 다채널 DMA 컨트롤러는 외부 동적 메모리의 액세스를 위해 MPMC의 멀티포트에 연결된 DMA 채널 관리를 그 주요 기능으로 한다. 그림 4는 4개의 멀티포트를 갖는 MPMC와 데이터 전송을 수행하기 위해 물리적으로 분리된 4개의 DMA 채널을 갖는 DMA 컨트롤러의 구조를 보인다.

스위치 인터페이스(Switch Interface)는 시스템 내부의 멀티프로세서에 의해 DMA 컨트롤러를 동시에 제어할 수 있도록 2개의 AHB 슬레이브 인터페이스를 제공한다. 레지스터 뱅크(Register Bank)는 DMA 컨트롤러를 제어하고 DMA 채널들의 데이터 전송 동작 수행을 제어하기 위한 레지스터 세트를 갖는다. 레지스터 제어 유닛(Register Control Unit)은 멀티채널 관리 유닛(Multi-Channel Management Unit)으로부터 제공되는 사용 가능한 DMA 채널 정보 및 전송 영역 분할 정보를 이용하여 DMA 채널의 데이터 전송 동작을 제어하고 채널 제어를 위한 레지스터 뱅크의 레지스터들을 갱신하는 역할을 수행한다. 멀티채널 관리 유닛은 MPMC와 연결된 DMA 채널들이 버스를 사용할 수 있는지 확인하고 버스 사용이 가능한 DMA 채널의 버스 사용권한을 버스 중재기(arbiter)로부터 획득한다. 이와 동시에 DMA 채널 정보와 전송 데이터의 영역 분할 정보를 레지스터 제어 유닛으로 전달하여 DMA 채널 동작을 제어하도록 한다. DMA 마스터 인터페이스 모듈(DMA Master I/F Module)은 MPMC와의 인터페이스를 위한 4 채널 AHB 마스터 인터페이스와 시스템의 주변장치

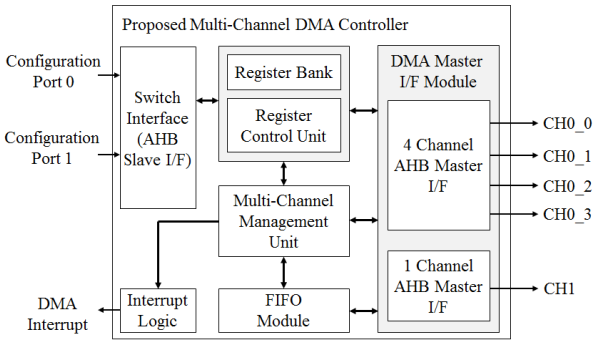

그림 4. 제안된 다채널 DMA 컨트롤러의 구조

Fig. 4. Architecture of the proposed multi-channel DMA controller.

와 인터페이스를 위한 1 채널 AHB 마스터 인터페이스를 지원한다. 인터럽트 로직(Interrupt Logic)은 DMA 컨트롤러의 내부 동작에 따른 인터럽트를 외부로 전달하는 기능을 수행한다.

Ⅳ. 고속 다채널 데이터 전송 방법

상용 DMA 컨트롤러[10]를 포함한 기존의 DMA 컨트롤러들은 일반적으로 단일 채널 액세스에 기반을 두고 있다. 이것은 DMA 컨트롤러가 장치와 장치 간 point-to-point 전송을 수행하는데 있어서 하나의 물리적 채널을 사용함을 의미한다. 따라서 이러한 DMA 컨트롤러가 MPMC와 사용될 때 point-to-point 데이터 전송에 있어서 성능 저하를 갖는 단일 채널 데이터 전송을 수행하게 된다. 그러나 우리는 앞의 설명에서 다른 장치에 의해 사용되지 않는 MPMC의 모든 포트들을 사용함에 의해서 point-to-point 전송의 효율을 높일 수 있음을 보았다. 이를 위해 본 논문에서는 MPMC를 위한 물리적으로 분리된 다중 액세스 채널을 지원하고 이러한 다중 액세스 채널을 하나의 채널처럼 관리하는 멀티채널 관리기능을 갖는 DMA 컨트롤러를 제안한다.

제안된 DMA 컨트롤러는 외부 메모리의 블록 데이터 영역을 전송할 때 MPMC의 접근 가능한 DMA 채널의 수에 따라 그 데이터 영역을 나누어 전송을 수행한다. 이때 데이터의 영역 분할은 영역 어드레스의 열 또는 행을 기준으로 분할된다. 이렇게 나누어진 데이터 영역은 물리적으로 분리된 DMA 채널에 할당되고 각 DMA 채널은 할당된 영역의 데이터를 독립적으로 액세스하게 된다.

우리가 전송하고자 하는 메모리의 블록 데이터 영역에 대한 전체 행의 수를 n, MPMC의 액세스 가능한 포트의 수를 M, 전송하고자 하는 행의 영역을 $row[S]$, 사용 가능한 DMA 채널을 CH_r이라 한다. 이때 r은 사용 가능한 DMA 채널의 일련번호를 의미하며 0부터 M-1의 범위를 갖고, S는 전송하고자 하는 행을 의미하며 0부터 n-1의 값을 갖는다. 전송하고자 하는 행의 영역 $row[S]$는 $r = S(\mathrm{mod}\, M)$ 인 CH_r에 할당된다. 여기서 액세스 가능한 포트라 함은 다른 마스터 장치에 의해 사용되지 않는 MPMC의 유휴 포트를 의미한다. 결과적으로 DMA 컨트롤러는 다음의 식 (1)을 통해서 영역 분할을 수행하고 분할된 영역을 사용가능한 DMA채널에 할당한다.

$$CH_{S(\mathrm{mod}\, M)} = row[S] \text{ for } S = \{0, 1, 2, ..., n-1\} \quad (1)$$

앞에서 설명한 바와 같이 본 논문은 기본적으로 외부 메모리의 row 어드레스의 반복적인 변화에 의한 row 어드레스 스트로브 신호를 줄이기 위해 블록 데이터 영역을 row 어드레스를 기준으로 분할한다. 나누어진 블록 영역에 대한 정보는 제안된 DMA 컨트롤러의 각 DMA 채널들에 자동으로 설정된다. 따라서 외부 프로세서에 의한 DMA 설정 오버헤드는 크지 않다. 이는 외부 프로세서가 전송하고자하는 전체 블록 데이터 영역에 대한 기본 전송 정보, 즉 소스 및 목적지 어드레스, 데이터 전송 크기, 전송 모드 등에 대한 정보를 제공하면 각 DMA 채널의 설정은 DMA 컨트롤러의 멀티채널 관리 유닛이 수행하기 때문이다. 설정된 DMA 채널들은 서로 독립적으로 MPMC의 버스 인터페이스를 통해 메모리에 접근하게 된다. 만약 외부 메모리의 블록 데이터 영역이 단지 연속적인 column만으로 이루어진다면, 데이터 영역의 분할은 column을 기준으로 나눌 수도 있다.

그림 5와 6은 외부 동적 메모리와 제안된 DMA 컨트롤러 사이의 MPMC를 통한 멀티채널 데이터 전송의 예를 보인다. 그림 5는 제안된 DMA 컨트롤러가 3개의 채널을 사용하여 MPMC를 통해 메모리에 접근하는 것을 보이고, 그림 6은 제안된 DMA 컨트롤러가 4개의 채널을 사용하여 MPMC에 접근하는 것을 보인다. 이를 위해 제안된 DMA 컨트롤러는 MPMC에 접근 가능한 DMA 채널을 선택하고, 다음으로 외부 동적 메모리의 액세스할 영역을 분할한 후, 분할된 블록 데이터 영역

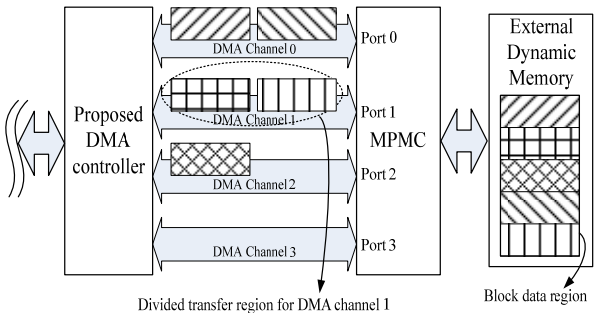

그림 5. DMA 컨트롤러와 MPMC 사이에서의 3 DMA 채널을 통한 멀티채널 데이터 전송

Fig. 5. Multi-channel data transfer using three DMA channels between the proposed DMA controller and the MPMC.

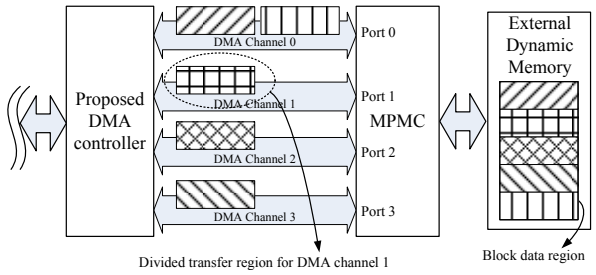

그림 6. DMA 컨트롤러와 MPMC 사이에서의 4 DMA 채널을 통한 멀티채널 데이터 전송

Fig. 6. Multi-channel data transfer using four DMA channels between the proposed DMA controller and the MPMC.

을 사용가능한 각 DMA 채널에 할당하여 메모리 접근을 수행한다.

V. 실험 결과

제안된 DMA 컨트롤러는 Verilog-XL을 사용해서 설계되고 검증되었다. 검증 환경은 200 MHz 시스템 클럭이 사용되었고, 32-bit 외부 SDRAM (synchronous dynamic random access memory)이 MPMC에 연결되었다. 그리고 MPMC를 위해서 ARM사의 PL242[8]를 사용하였으며 시스템 클럭에서 동작하는 내부 SRAM (static random access memory)이 외부 SDRAM과 데이터 전송을 위한 내부 주변장치의 하나로 사용되었다. 제안된 DMA 컨트롤러는 시스템 클럭으로 동작하고 SDRAM은 시스템 클럭의 1/2 클럭으로 동작한다. 성능 검증을 위한 MPMC와 DMA 컨트롤러 간의 연계 환경은 그림 7에서 보인다.

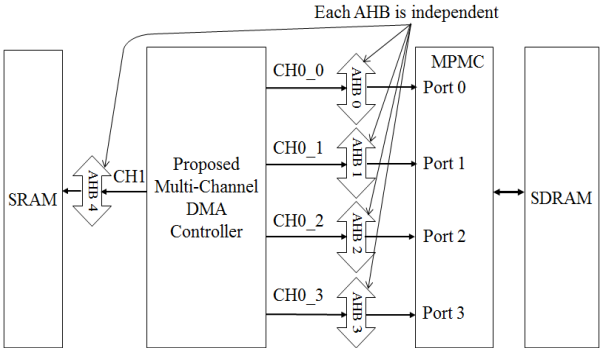

그림 7. DMA 컨트롤러와 MPMC 연계 성능 검증 환경

Fig. 7. Environment for the performance verification of the link between the DMA controller and the MPMC.

표 1. 제안된 DMA 컨트롤러와 MPMC에서의 단일 DMA 채널과 4 DMA 채널 동작의 비교

Table 1. Comparison of the read/write cycles of both a single DMA channel and four DMA channels between the proposed DMA controller and the MPMC.

Burst Mode	Write Bandwidth (Gbps)		Read Bandwidth (Gbps)	
	1 Ch. Access	4 Ch. Access	1 Ch. Access	4 Ch. Access
4 burst	1.07	3.76	1.73	5.61
8 burst	1.83	4.82	2.61	5.61
16 burst	2.84	4.82	3.76	5.61

표 2. MPMC와 DMA 액세스 채널에 따른 내부 SRAM의 데이터 전송 대역폭

Table 2. Data transfer rate to internal SRAM according to the DMA access channels for the MPMC.

Burst Mode	Read Operation (cycle)		Write Operation (cycle)	
	1 Ch. Access	4 Ch. Access	1 Ch. Access	4 Ch. Access
4 burst	384	109	237	73
8 burst	224	85	157	73
16 burst	144	85	109	73

표 3. 제안된 DMA 컨트롤러와 상용 DMA 컨트롤러의 MPMC와 연계된 제어부하 및 전송 성능 비교

Table 3. Comparison of the control overhead and transfer performance considering the MPMC between our DMA controller and a commercial DMA controller.

Category	Proposed DMA controller		Commercial DMA controller	
	1 Ch. Access	4 Ch. Access	1 Ch. Access	4 Ch. Access
The Number of Register Configuration (*A*)	7	7	33	33
The number of Interrupts (*B*)	1	1	1	4
Only Data Transfer Cycles (*C*)	384	109	384	109
Total Transfer Cycles (*D*)	512	237	616	641
Data Transfer performance at 200MHz	0.80 Gbps	1.73 Gbps	0.66 Gbps	0.64 Gbps
Control overhead ((*D*-*C*)/*D*)	0.25	0.54	0.38	0.83

표 1은 MPMC와 DMA 컨트롤러 사이에서 단일 채널과 4 채널 동작 시 나타나는 데이터 읽기/쓰기 사이클의 비교를 보인다. 표 1의 비교를 위해서 본 논문에서는 8 x 8 블록의 256 바이트 데이터 전송에 대해서 4, 8, 16의 버스트 모드(burst mode) 설정에 따른 DMA 컨트롤러 동작을 수행하였다. 그 결과로써 4개의 DMA 채널을 사용하는 4 버스트 모드에 있어서 단일 채널 DMA 전송에 비해 제안된 DMA 컨트롤러는 읽기/쓰기 전송 사이클에 있어서 읽기는 72%, 쓰기는 69%의 전송 성능 향상을 보인다.

이와 더불어, 우리는 내부의 SRAM에 저장되는 데이터에 대한 대역폭을 검증하였다. 그 결과는 표 2에서 보이고 있으며, 본 논문에서 예상한 바와 같이 데이터 대역폭은 사용되는 MPMC의 DMA 채널이 증가함에 따라 향상됨을 알 수 있다. 또한 버스트 모드를 제어함으로써 전송 효율을 더 높일 수 있음을 알 수 있다.

표 1과 2에서 보이는 바와 같이 제안된 DMA 컨트롤러는 메모리의 블록 데이터를 효과적이고 빠르게 전송할 수 있다. 이를 위해 우리는 단일 채널을 가지는 상용 DMA 컨트롤러나 이미 개발된 DMA 컨트롤러들을 다수 사용하여 앞에서 설명한 다채널 DMA 컨트롤러를 구성할 수도 있다. 그러나 이러한 단순히 물리적으로 분리된 다수의 DMA 채널 사용은 각 채널의 제어를 독립적으로 수행해 주어야 하며 그 결과 개별 DMA 채널의 레지스터 설정 및 인터럽트 처리와 같은 제어 프로세스는 증가하게 된다. 본 논문에서 제안된 DMA 컨트롤러는 자동화된 채널 제어를 통해 레지스터 설정 시간을 줄임으로써 제어 오버헤드를 줄이는 것과 함께 전송 성능을 향상시키는 효과를 제공한다. 이러한 결과는 표 3에서 보인다.

표 3에서는 그림 7의 환경에서 상용 듀얼포트 DMA 컨트롤러[10]와 제안된 DMA 컨트롤러를 이용하여 4 포트 MPMC와 연계 동작 수행에 따른 DMA 컨트롤러들 간의 제어 오버헤드 및 데이터 전송 성능을 비교한다. 그 결과들은 DMA 컨트롤러 자체의 성능이 아니라 MPMC와 연계되어 나타나는 외부 SDRAM에서 내부 SRAM으로의 데이터 전송성능을 나타낸다. 이는 DMA 컨트롤러의 데이터 전송 성능이 MPMC가 SDRAM을 액세스해 데이터를 제공하는 성능에 좌우됨을 의미한다. 표 3의 결과는 point-to-point 전송에 있어서 MPMC의 특성을 고려하여 기존 MPMC의 수정 없이 MPMC를 액세스하는 DMA 컨트롤러의 채널 개수를 증가시키는 방법만으로 손쉽게 데이터 전송 사이클의 향상을 도모할 수 있으며, DMA 채널의 설정을 자동화함으로써 레지스터 제어 횟수 및 인터럽트 횟수를 감소시켜 전체 전송 성능을 향상시킬 수 있음을 보인다.

상용 DMA 컨트롤러는 MPMC와의 인터페이스를 위해 AMBA AHB 프로토콜을 사용하는 DMA 컨트롤러를 대상으로 하였고, 물리적으로 하나의 DMA 채널을 가지고 있으며 내부적으로 8개의 논리적 채널을 갖는다. 4개의 DMA 채널 동작비교에 있어서는 기존의 DMA 컨트롤러들이 단일 채널 DMA 동작에 기반하고 있기 때문에 우리는 4개의 상용 단일 채널 듀얼포트 DMA 컨트롤러를 조합하여 물리적으로 4 채널을 만들어 MPMC에 연결하였다. 이 때 4 채널을 구성하는 각 DMA 채널들은 독립적으로 제어된다.

성능 검증은 4 포트 MPMC에 연결된 외부 SDRAM의 읽기 동작에 대하여 8 x 8 블록 (256 bytes)의 전송을 기준으로 하였다. 이 때 8 x 8 블록은 row를 기준으로 8개의 영역으로 나누어지고 이를 각 DMA 채널에 할당하였다. 버스트 모드는 4로 설정되고, 데이터 전송 단위는 32 비트이다. 소모되는 제어 오버헤드 계산에 있어서는 DMA 채널 설정을 위한 싱글 데이터 전송에 대하여 AMBA 버스의 아비트레이션(arbitration)을 고려해 4 사이클을 기본 단위로, 인터럽트 처리에 대해서는 프로세서 내부 처리를 고려하여 100 사이클을 기본 단위로 설정하였다. 따라서 표 3의 전체 전송 사이클 (total transfer cycle) = $4A+100B+C$로 계산된다.

표 3에서 보이는 바와 같이 제안된 DMA 컨트롤러는 상용 DMA 컨트롤러에 비하여 1 채널을 이용한 DMA 전송에 있어서는 45%, 4 채널을 이용한 DMA 전송에 있어서는 76% 까지 레지스터 설정과 인터럽트 처리를 위한 제어 오버헤드를 줄일 수 있었으며, 설정된 8 x 8 기준 블록에 대한 4 채널을 이용한 전체 전송 사이클(D)은 제어 오버헤드의 차이로 인해 상용 DMA 컨트롤러에 비하여 37%의 사이클만을 소모함을 볼 수 있다.

이러한 제어 오버헤드의 차이는 4 채널을 구성하는 상용 DMA 컨트롤러가 서로 독립적으로 동작함으로써 외부 SDRAM 메모리의 전송 영역에 대한 DMA 채널 설정을 각 채널별로 독립적으로 수행해야 하며 DMA 채널의 인터럽트 처리를 각 채널에 대해 순차적으로 처리해야하기 때문이다. 반면 제안된 DMA 컨트롤러는

전송을 위해 설정된 영역에 대하여 전송이 완료될 때까지 각 채널의 설정을 자동으로 설정하고 관리할 수 있어 지속적인 DMA 채널 설정에 따른 부하 및 프로세서가 처리해야 하는 인터럽트를 줄일 수 있다는 장점을 갖는다.

Ⅵ. 결 론

본 논문은 MPMC의 데이터 전송 특성을 고려하는 효과적인 데이터 전송 방법에 대하여 제안하였다. 제안된 DMA 컨트롤러는 point-to-point 전송에 있어서 기존의 단일 채널 DMA 컨트롤러를 사용한 일반적인 데이터 전송 구조보다 더 빠른 속도로 데이터 전송을 수행할 수 있었다. 이러한 결과로 부터 본 논문은 MPMC의 수정 없이 다채널 DMA 컨트롤러의 효과적 개선을 통해 MPMC의 동작 효율을 향상시킴으로써 외부 메모리와 내부 장치 간 데이터 전송성능을 손쉽게 향상시킬 수 있음을 보였다.

REFERENCES

[1] ARM, *AMBA Specification (Rev 2.0)*, http://www.arm.com
[2] ARM, *AMBA AXI Protocol Specification (v1.0)*, http://www.arm.com, 2004.
[3] C.-H. Yu, C.-K. Liu, C.-H. Kang, et al., "An efficient DMA controller for multimedia application in MPU based SoC," IEEE Int. Conf. on Multimedia and Expo, pp. 80-83, 2007.
[4] G. Ma and H. He, "Design and implementation of an advanced DMA controller on AMBA-based SoC," IEEE Int. Conf. on ASIC, pp. 419-422, 2009.
[5] K.-J. Lin, C.-H. Huang, and C.-C. Lo, "Design and Implementation of a Schedulable DMAC on an AMBA-Based SOPC Platform," IEEE Asia Pacific Conference on Circuits and Systems, pp.279-282, Dec. 2006.
[6] Texas Instrument, *TM320C6000 peripherals reference guide*, http://www.ti.com
[7] S. Srinivasan and D.B. Stewart, "High speed hardware assisted real time interprocess communication for embedded microcontrollers," Proc. of the 21st Real-Time Systems Symposium, pp. 269-279, 2000.
[8] ARM, *PrimeCell AHB SDR and NAND Memory Controller (PL242) Technical Reference Manual*, Dec. 2006.
[9] Synopsys, *DesignWare DDR2/3-Lite memory Controller IP*, http://www.synopsys.com
[10] ARM, *PrimeCell DMA Controller (PL080) (Rev r1p3)*, http://www.arm.com

저 자 소 개

천 익 재(정회원)

1998년 충남대학교 전자공학과 학사 졸업.
2000년 충남대학교 전자공학과 석사 졸업.
2006년 충남대학교 전자공학과 박사 졸업.
2006년～현재 한국전자통신연구원 선임연구원
<주관심분야 : 통신모뎀, 디지털신호처리, NoC 설계, SoC 설계>

여 준 기(정회원)

1998년 경북대학교 컴퓨터공학과 학사 졸업.
2000년 한국과학기술원 전자전산학과 석사 졸업.
2004년 한국과학기술원 전자전산학과 박사 졸업.
2004년～현재 한국전자통신연구원 혼성신호처리 연구실 실장/선임연구원
<주관심분야 : 컴퓨터비젼, 재구성형 프로세서, 저전력 디지털 신호처리>

노 태 문(평생회원)

1984년 경북대학교 전자공학과 학사 졸업.
1986년 경북대학교 전자공학과 석사 졸업.
1998년 경북대학교 전자공학과 박사 졸업.
1998년～현재 한국전자통신연구원 책임연구원
<주관심분야 : 신호처리, 반도체, SoC 설계>

이 문 식(정회원)

1997년 성균관대학교 제어계측공학과 학사 졸업.
1999년 광주과학기술원 기전공학과 석사 졸업.
2005년 광주과학기술원 기전공학과 박사 졸업.
2008년～2009년 미국 Stanford 대학교 Post-Doc.
2005년～현재 한국전자통신연구원 무선근접통신 연구실 실장/책임연구원
<주관심분야 : 5G 이동통신, D2D 통신, M2M 통신>