

논문 2005-42SD-4-2

MPEG 시스템용 다중 작업에 적합한 양방향 버스 구조 (Bi-directional Bus Architecture Suitable to Multitasking in MPEG System)

전 치 훈*, 연 규 성*, 황 태 진*, 위 재 경**

(Chi-hoon Jun, Gyu-sung Yeon, Tae-jin Hwang, and Jae-Kyung Wee)

요 약

본 논문은 OCP(Open Core Protocol)에 호환되는 파이프라인 구조를 가진 시스템 버스와 MPEG 시스템에 적합한 메모리 버스로 구성된 계층 구조를 가지는 새로운 동기 세그먼트 버스를 제안한다. 이 구조는 MPEG 시스템의 모바일 제품에 사용되는 영상 데이터 처리를 위한 메모리 인터페이스에 기반을 둔 버스 구조와 멀티 마스터와 멀티 슬레이브를 사용하여 고성능의 다중 처리를 위한 양방향 다중 버스 구조(bi-direction multiple bus architecture)를 가진다. 효율적인 데이터 처리를 위하여 파이프라인 스테이지와 결합된 마스터와 슬레이브의 주소번지가 latency를 결정하며, 시스템의 특성에 따라서 각각의 IP 코어를 배치하였다. 제안된 버스는 저전력 구현을 위하여 세그먼트 버스 구조를 가지고, 멀티미디어 SoC 시스템의 성능 저하 없이 다중 작업이 가능한 구조를 가지며 확장이 가능하다. 제안된 버스 구조는 AMBA와 비교 하였을 때 bandwidth는 3.7배 증가 하였고 latency는 0.25배 감소하였다.

Abstract

This paper proposes the novel synchronous segmented bus architecture that has the pipeline bus architecture based on OCP(open core protocol) and the memory-oriented bus for MPEG system. The proposed architecture has bus architectures that support the memory interface for image data processing of MPEG system. Also it has the segmented bi-directional multiple bus architecture for multitasking processing by using multi-masters/multi-slave. In the scheme address of masters and slaves are fixed so that they are arranged for the location of IP cores according to operational characteristics of the system for efficient data processing. Also the bus architecture adopts synchronous segmented bus architecture for reuse of IP's and architecture of developed chips. This feature is suitable to the high performance and low power multimedia SoC system by inherent characteristics of multitasking operation and segmented bus. Proposed bus architecture can have up to 3.7 times improvement in the effective bandwidth and up to 4 times reduction in the communication latency.

Keywords : bus, hierarchy, arbitration, bi-direction, multitasking, MPEG

I. 서 론

VLSI 기술의 발전으로 보다 많은 양의 회로를 단일 칩에 집적할 수 있게 되었고, 이는 SoC (System-On-Chip) 시대의 도래를 가능하게 했다. 고집적의 시

스템을 단 기간에 개발하기 위해서는 IP (Intellectual Property)로 통용되는 기존에 잘 설계된 IP 코어를 이용하는 것이 절실히 요구된다^[1]. 이러한 재사용 가능한 IP 코어와 IP 코어 사이의 통신을 위하여 칩 내부에서의 On-chip 통신은 주요한 이슈로 부각되었다. On-chip 통신을 위하여 많은 버스^{[2] [4]}가 제안되었고, 이 버스들은 Standard 버스^{[2] [5]}와 Wrapper 기반의 버스^[6]로 분류된다.

Standard 버스들은 IP 코어 사이의 쓰기 읽기 동작을 어떻게 할지에 대한 접속 프로토콜로 정의되고, IP 코

* 학생회원, ** 정회원 숭실대학교 전자공학과
(School of Electronic Engineering, Soongsil University)

※ 본 연구는 SIPAC의 지원을 받아 수행되었음.
접수일자: 2005년1월27일, 수정완료일:2005년3월25일

어들은 이 프로토콜들 중에 하나에 따르게 된다. 이 프로토콜은 다른 SoC 버스를 사용하는 곳에 재사용할 수 있으나, 이에 따른 내부 회로 변화가 필수적이다.

Wrapper 기반의 접근은 IP 코어들 각각의 인터페이스 프로토콜을 사용하여 버스 프로토콜에 독립적으로 동작하며, wrapper는 IP 코어와 IP 코어 사이의 통신을 처리한다. 따라서 wrapper 기반 인터페이스 프로토콜에 따른 IP 코어들은 SoC에 집적화할 수 있고, Standard 버스와 같은 backbone 버스들과 다르다. 하지만 단순한 wrapper 하드웨어 구조는 latency가 증가하기 때문에 최적화해야만 하고, 성능향상을 위하여 회로 추가가 필요하다. 이러한 IP 코어의 재사용의 효율을 높이고자 VCI(Virtual Component Interface)와 실제 많이 사용하고 있는 OCP(Open Core Protocol)에 적용 가능한 wrapper 기반의 버스가 제안되었다. 기존의 버스만을 사용한 시스템 설계는 시스템의 특성에 따라 성능이나 효율이 달라진다. 멀티미디어 시스템의 MPEG와 같은 데이터의 양이 많고, 메모리 점유와 접근이 빈번한 특성을 갖는 시스템의 경우, 그 시스템에 효율적인 버스가 필요하다.

본 논문에서는 MPEG 시스템에 적합한 시스템 특성이 있는 버스구조를 제안한다. 첫째, MPEG 시스템의 모바일 제품에 사용되는 영상 데이터를 처리하며, 다중 작업이 가능하다. 영상 데이터의 처리를 위해 외부 메모리로 SDRAM이 쓰인다고 가정하였다. 둘째, 시스템에 사용되는 마스터와 슬레이브의 특성(데이터 크기, 성능, latency 등)을 알기 위해 latency를 고정 시켰다. 이를 통해 각 IP 코어들의 배치와 주소번지를 결정하고 통신의 효율성을 극대화 시켰다. 셋째, IP의 재활용을 위하여 OCP 프로토콜을 지원하였다.

본 논문의 제 II장에서는 기존의 버스의 시스템에 MPEG 시스템 적용 시의 문제점과 제안된 버스구조와의 차이점과 제안된 버스 구조의 통신 프로토콜 특징에 대하여 다룬다. 제 III장에서는 제안된 버스구조의 검증 방법과 시뮬레이션의 결과와 제 IV장에서는 제안된 버스 구조의 대한 간단한 요약과 기대효과를 제시하며 결론을 맺는다.

II. 본 론

1. MPEG 시스템 적용 시 기존버스의 문제점

새로운 버스제안에 앞서 일반적으로 가장 많이 쓰이

는 ARM사의 AMBA와 Sonics사의 Sonics 버스에 MPEG 시스템에 적용 했을 시의 문제점을 알아보고, 그 외의 samba-bus^[7]와 segment-bus^[8]의 특징을 비교하고 새로운 버스의 특징을 제안하고자 한다.

ARM사의 AMBA의 경우 고성능동작을 위한 AHB와 외부 주변장치의 저 성능의 동작을 하는 APB 버스로 구성된다. AMBA를 MPEG 시스템에 적용 했을 때의 AMBA의 특징과 문제점은 다음과 같다. 첫째, AMBA는 ROM이나 RAM과 같은 On-chip IP들로 구성된 시스템에 좋은 성능을 보이는 반면, 외부 메모리와 같은 긴 latency나 다양한 latency로 구성된 시스템에서 성능 저하를 보인다. MPEG 시스템과 같이 메모리 접근이 빈번한 경우 SDRAM은 AHB의 고성능 버스와 인터페이스를 하게 되므로 마스터에서 SDRAM에 대한 요청 시 응답에 긴 latency를 가진다. AMBA는 하나의 IP가 버스를 점유 할 때 다른 IP들은 사용할 수 없는 버스 구조이므로 긴 latency를 가지는 SDRAM을 사용할 경우, 하나의 마스터가 버스 전체 (AHB)를 사용하게 되고, 다른 마스터가 버스를 공유하여 사용할 수 없게 되어 효율성이 저하된다. 이러한 버스 전체를 점유하게 되는 문제점을 해결하기 위하여 AMBA는 다중 마스터가 단일 슬레이브에 전송이 가능한 Multi-layer방식을 지원한다. 하지만 구조의 특성상 마스터의 수가 늘어나면 버스의 면적이 증가하게 되고, 배선 연결이 복잡해지는 선로 연결 복잡성(wirability) 문제가 발생한다. 둘째, AMBA는 다중 마스터와 단일 슬레이브의 동작에서 하나의 arbiter로 우선순위를 결정하는 Central arbitration 구조를 가지므로 arbiter의 역할이 중요해지고 복잡해진다.

Sonics의 버스는 각 IP들의 Time schedule(TDMA)을 이용하여 arbitration을 한다. Sonics의 버스를 사용하여 SDRAM을 사용할 경우, 마스터는 요청 후에 데이터의 입출력이 일어날 때까지 수 사이클 이상의 시간을 필요로 한다. Sonics 버스는 SDRAM 인터페이스가 직접 해당하는 마스터에게 읽기나 쓰기를 요구하게 함으로서 성능 향상을 가져올 수 있으며 AMBA와 비교하여 좀 더 최적화가 쉬운 버스 구조를 가진다. 또한 OCP를 지원함으로써 IP의 재활용이 용이하다. 하지만 SDRAM의 효율을 높이려면 TDMA등의 경우 여러 개의 동작요구(requests)들을 모아 burst의 연관성을 찾아서 수행해야 하는 정교한 SDRAM 인터페이스의 구현이 필요로 하는 문제점과 설계 전에 각 IP들의 정확한

정보들을 알고 있어야 한다.

본 논문에서 제안한 버스는 기존의 세그먼트 버스^[7]와 비동기 SAMBA-BUS^[8]에 비하여 다중 작업이 가능한 양방향 버스를 제시하고, 기존의 칩 설계와 호환성을 가지는 동기 시스템을 위해 설계되었다.

2. 제안하는 버스 구조

가. 전체 구조

앞 절에서 설명한 기존 버스의 문제점을 해결하고자 본 논문에서는 MPEG 시스템의 모바일용으로 데이터 처리를 위한 메모리 시스템 버스와 다중 작업 (Multitasking) 처리가 가능한 고성능 시스템을 위한 계층적 버스 구조를 제안하였다.

제안하는 버스구조는 다음과 같은 특징을 가진다. 첫째, MPEG 시스템의 모바일 제품에 사용되는 영상 데이터 처리를 한다. 둘째, MPEG와 같은 데이터 폭 (data-width)이 큰 시스템은 메모리 버스와 통신하고, MPEG 성능 저하 없이 다중작업이 가능한 시스템 버스 구조를 가진다. 또한 시스템 버스에서는 양방향 구조를 가지며 OCP를 지원한다. 셋째, 각 IP 코어의 어드레스 번지를 latency에 따라 고정하여 데이터 통신 효율을 높이고, 고성능 버스 채널의 제어를 위하여 채널 제어 방법을 단순화 하였다. 주소번지 지정은 IP의 latency와 시스템의 특성에 따라서 배치가 달라진다.

그림 1은 제안된 전체 버스 구조를 나타낸다. 제안된 버스 구조는 시스템 버스와 메모리 버스로 이루어진 계층적인 버스 구조이다. 시스템 버스는 양방향 버스 구조와 다중작업을 위한 구조로 제안되었고, 메모리 버스는 MPEG 시스템의 각 IP들과의 인터페이스를 위한 버스로서 메모리의 점유율이 높은 MPEG 시스템의 특성을 고려하여 SDRAM과의 원활한 통신 프로토콜을 가지는 멀티 마스터와 단일 슬레이브 구조를 가지고 있다.

제안된 버스 구조는 Local Arbiter(LArbiter), Central Arbiter(CArbiter), MBUS Arbiter(MArbiter)의 3가지의 arbitration 방식을 적용하였다. Local arbiter는 그림 1에서와 같이 각 마스터에 연결되어 있고, 설계 전 고려된 마스터들의 사용 빈도에 따라서 TDMA 방식으로 arbitration을 한다. Central arbiter는 local arbiter에서의 요청에 대하여 버스의 채널 상태에 따라 마스터의 승인 여부를 결정하는 채널 제어 블록이다. Local

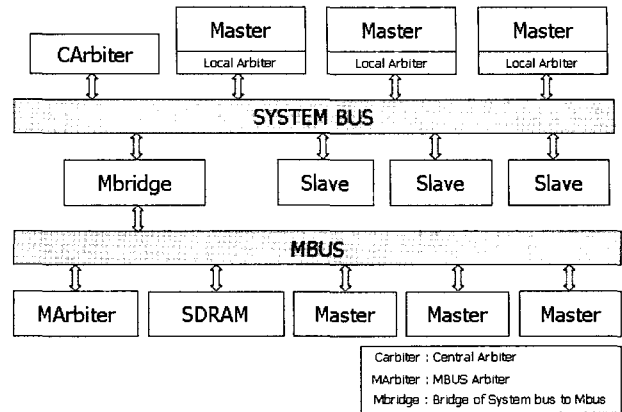


그림 1. 제안된 버스 구조.
Fig. 1. Proposed bus architecture.

arbiter와 central arbiter의 전송 및 제어 신호는 시스템 버스(System Bus)의 어드레스, 데이터, 명령어의 전송 버스와는 별개로 직접 연결된 버스 구조를 가지고 있어 짧은 latency를 갖는다. 또한 Local arbiter와 central arbiter는 분리된 구조로 설계하여 마스터의 추가 확장이 가능하도록 하였다. 메모리 버스의 arbiter인 Marbiter는 MPEG 각 IP 코어의 요청에 따라 MPEG 시스템이 원활히 동작하도록 arbitration을 하게 된다. 시스템 버스와 MBUS는 Mbridge를 통하여 통신한다. 여기서 Mbridge는 시스템 버스에서는 슬레이브로, 메모리 버스에서는 마스터로 동작 하며, Local arbiter와 central arbiter는 독립적으로 동작한다.

나. 메모리 버스 (MBUS) 구조

MPEG은 데이터 폭이 크고, 메모리의 접근 빈도가 높은 특성을 가지고 있다. 이러한 이유로 SDRAM과의 인터페이스에 높은 비중을 두는 그림 2와 같은 버스 구조를 가져가는 것이 SDRAM 통신의 효율을 높일 수 있다. 또한, 이 구조는 다중 마스터와 단일 슬레이브 구조로 구현되었다. 그림2 에서 Mbridge는 메모리 버스에서 마스터로서 동작하며, 이때 SDRAM은 마스터에서 요청 후 실제로 데이터 입출력이 일어날 때까지 수 마이크로 이상의 시간이 걸리는 긴 latency를 갖고 있다. SDRAM 인터페이스가 해당 마스터에게 읽기/쓰기를 요구하고, 데이터 arbitration 신호는 미리 대기하고 있을 수 있다. 따라서 메모리를 접근할 타이밍을 알고 있어야 하므로 MPEG의 각 블록들의 동작과 데이터의 전송 타이밍을 arbiter가 정확히 알고 있어야 한다. SDRAM에서 open 된 row 에 대해서는 무작위 접근

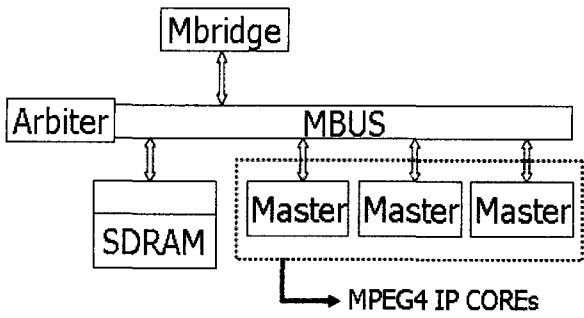


그림 2. 제안된 메모리 기반 버스 구조 (MBUS)
 Fig. 2. Proposed memory based bus architecture (MBUS).

(random access)이 되지만 그렇지 않을 경우, precharge 와 row-open 시간이 요구된다. 이 경우 Bank-interleaving을 burst 동작과 결합할 경우 효율을 극대화 할 수 있다. Bank-interleaving은 하나의 IP에서 동작요구(request)가 일어날 경우는 최적화할 수 있지만 다중 IP들과 동작을 하는 경우 효율은 떨어진다. 이러한 SDRAM기반의 메모리 버스에서 메모리 접근빈도를 예상하여 메모리 접근 가능한 타이밍에 다중 작업을 가능하게 하는 버스를 다음 절에 설명한다.

다. 시스템 버스 구조

제안된 버스 구조는 IP의 재사용이 용이 하도록 OCP(Open Core Protocol)에 호환되는 프로토콜 인터페이스를 가지고 있고, 성능 향상을 위하여 파이프라인 구조로 되어있다. 또한, Mbridge의 회로 추가로 시스템 버스와 메모리 버스에 연결이 가능하다.

그림 3은 마스터에서 슬레이브로의 전송 요청 시 다중 작업을 위한 시스템 버스의 대칭 구조를 보여준다. 마스터에서 요청이 일어 날 경우 Local arbiter에서 arbitration 후 “Dones”, “Requests”, “Accepts”들의 직접 제어 신호들과 central arbiter의 채널 신호로 버스 채널을 제어 하게 된다. 이후 승인된 마스터에 한하여 양방향 버스 구조를 가지는 사각형 점선의 채널 게이트 블록을 통하여 Command, Address, Data의 전송이 이루어진다. 이 블록은 Mux, F/F, 3-State 버퍼로 구성되어 있으며, 성능 향상과 데이터의 정확한 전송을 위하여 F/F를 삽입한 파이프라인 구조를 가지고, 배선효율을 높이고자 양방향 전송이 가능한 구조로 설계 되었다. 이 양방향 구조의 채널 게이트 블록에서 사용되는 3-State 버퍼는 central arbiter에서 제어하며 버스의 채널을 열고 닫는 스위치의 역할을 수행한다. Local

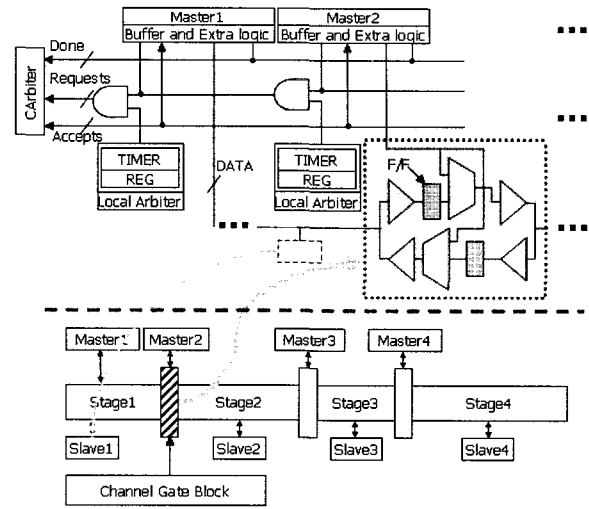


그림 3. Command, Address 및 Data의 양방향 파이프라인 버스 구조.
 Fig. 3. Bi-direction and pipeline bus architecture for command, address, and data.

표 1. 다중 작업을 위한 Master ID와 Slave ID의 고정

Table 1. Fixed Master ID and Slave ID for Multitasking.

Master ID	001	010	011	100
Slave ID	001	010	011	100
Bus Channels	001	010	011	100

arbiter에서 요청된 어떤 동작에 대하여 각 마스터와 슬레이브의 주소번지를 central arbiter에서 ROM 형태로 기억하고 있다가 이전 마스터와 슬레이브의 주소번지를 central arbiter에서 비교하여 3-state 버퍼를 스위칭하는 방식으로 전송을 제어한다. 각 마스터들은 설계 전에 미리 고려된 버스 점유율과 사용 빈도에 따른 시간 정보에 따라서 local arbitration 후 central arbiter에서 버스 채널의 상태를 체크 후 전송을 시작한다.

본 구조의 local arbiter는 마스터가 할당된 시간에만 버스를 점유할 수 있고, 파이프라인 동작이 가능하다. 또한 central arbiter와 직접 연결되어 latency를 줄이기 때문에 성능 향상을 얻는다. 그림 3에서 TIMER와 REG는 설계 전 고려된 각 IP 코어의 버스 점유율에 따른 시간 정보를 저장하는 블록이다.

표 1은 다중 작업을 위한 마스터와 슬레이브의 주소번지를 나타낸다. Mater ID와 Slave ID는 설계 전에 고려된 IP의 특성인 마스터와 슬레이브의 관계와 latency에 따라서 결정된다. 표의 Bus Channels는 버스 채널의

번호를 나타내고, 파이프라인 스테이지 수에 비례한다. Central arbiter는 표에 주어진 Master ID와 Slave ID로 각 스테이지의 3-State 버퍼의 스위치를 On/Off 시킴으로 버스의 채널을 제어한다. 만약 이전의 수행중인 Master ID가 001이고 Bus Channels이 010 (Slave ID가 010)이라고 하고, 현재 요청된 Master ID가 001보다 크고, Bus Channels이 011보다 크다면 다중 작업이 가능하다. 이것은 central arbiter의 ROM 형태로 각 Master ID, Slave ID(Bus Channels)가 기억되어있어 ID만 가지고 central arbiter는 arbitration을 수행 할 수 있는 간단한 구조이다. 이는 수식 (1)로 표현 할 수 있다.

$$BAC = \begin{cases} CMid > PMid \text{ and } CSid > PSid \\ CMid < PMid \text{ and } CSid < PSid \end{cases} \quad (1)$$

이 수식에서 BAC(Bus Accept Channel)은 버스 채널의 사용 가능 여부, CMid(Current Master ID)은 현재 사용 요청 중인 Master ID, PMid(Previous Master ID)은 이전 사용 중인 Master ID, CSid(Current Master ID)은 현재 사용 요청 중인 Slave ID, PSid(Previous Slave ID)은 이전 사용 중인 Slave ID이다. 최대 가능한 다중 작업은 총 4개이며, 각각의 마스터와 슬레이브의 배치에 따라서 그 시스템의 효율이 결정된다.

이러한 작업을 위한 채널 게이트 블록에서 3-state 버퍼의 on/off를 위한 스위치는 latency를 줄이고, 시스템의 성능 향상을 위하여 직접 제어를 한다. 높은 효율을 위해서는 버스 점유율에 따라 그림 3과 같은 파이프라인 스테이지에 마스터와 슬레이브를 배치하고, 표 1에서와 같이 ID를 지정해야한다. 같은 Master ID와 Bus Channel를 가지는 경우가 가장 빈번히 사용되는 IP들로 배치되는 것이 가장 효율적이다. 또한, 마스터가 다른 마스터를 요청할 때는 수식(1)에서와 동일하게 Bus Channels(각 IP 코어의 ID)의 상태에 따라서 버스 접근을 허용한다.

그림 4는 제안된 버스에 할당된 신호의 형식을 나타내는 것으로, 파이프라인 구조의 전송을 하는 동작요구(Request) 모드와 응답(Response) 모드를 위한 버스로 나누어진다. 동작요구 모드는 마스터에서 슬레이브로의 전송을 의미하고, 응답 모드는 슬레이브에서 마스터로의 전송을 나타낸다. 이 두 가지 전송 모드는 앞서 설명한 양방향 버스의 구조를 가지고 있다. 동작요구 모드와 응답 모드의 구조는 Command, Address, Data로 동일한 버스 구조로 설계되었으며 서로 대칭적이다. 이

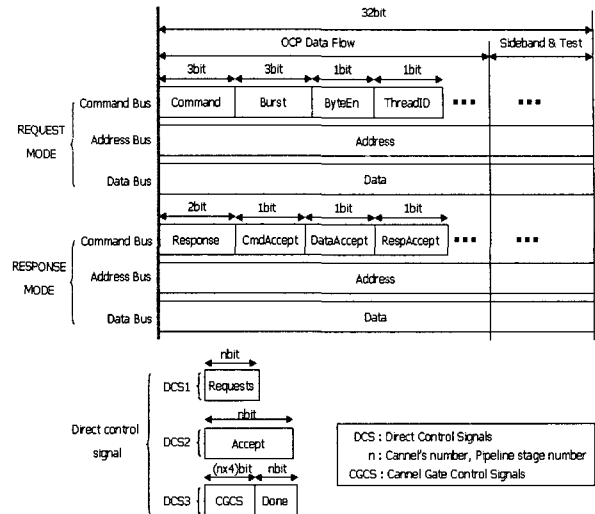


그림 4. 파이프라인 버스와 직접 제어를 위해 할당된 신호와 제어 정보 형식.
Fig. 4. Allocation of pipeline buses and direct control signals and their control information formats.

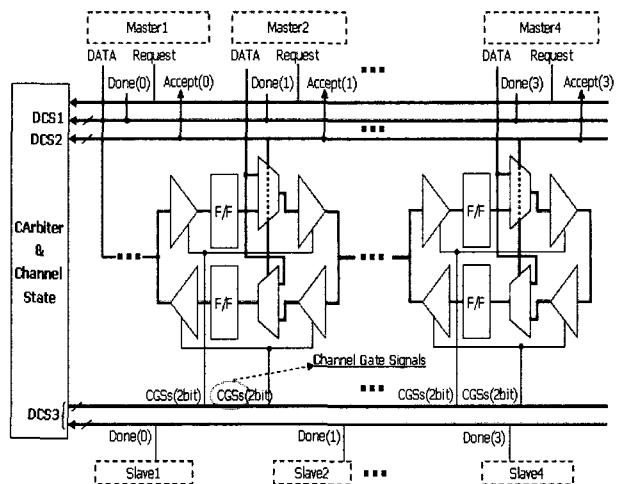


그림 5. 버스 제어를 위한 직접 제어 신호의 구성.
Fig. 5. Direct control signal control architecture for bus control.

버스들은 그림 4와 같은 OCP를 지원하는 신호들로 구성되어 있다.

이와 같은 버스 제어를 위한 신호의 형태를 그림 4에서 보여준다. DCS1, DCS2, DCS3 (Direct control signals, Done)은 DCS1은 직접 마스터와 local arbiter가 연결되는 경우, DCS2은 local arbiter와 central arbiter가 연결되는 경우, DCS3은 central arbiter와 슬레이브가 연결되는 경우로 central arbiter와 channel gate block(그림3 참고)를 통하여 연결된다. 각 비트는 파이프라인의 스테이지수에 의해 결정된다. 본 논문에서 구현된 구조는 4 마스터와 4 슬레이브의 IP를 가지

며, 4 스테이지 파이프라인의 4개의 채널을 가지는 버스로 구현 되어있다. 이 구조는 파이프라인의 추가(채널의 추가) 없이 Local arbiter의 수정만으로 사용되는 마스터와 슬레이브의 확장과 축소가 가능한 이점을 가진다.

그림 5는 버스의 직접 제어를 위한 신호의 구성을 나타낸다. 이 신호들은 버스 프로토콜과 독립적으로 구현되었고, 짧은 latency를 위하여 직접 연결 방식을 채용하였다. 이러한 버스의 직접 제어를 위한 신호들은 그림 5과 같은 DCS1, DCS2, DCS3의 3가지의 형식을 가지고 있다. 마스터에서 동작요구가 있으면, Master ID와 Slave ID의 신호를 entral arbiter로 전송하고, 버스의 채널을 확인 후 승인 신호 "Accept"를 마스터로 보내주어 수락을 하면 Command, Address와 Data의 전송을 수행한다. 이때 "Accept"신호를 보내준과 동시에 DCS3의 채널의 스위치를 On/Off 하는 신호인 CGSs를 보내주게 된다. 전송을 완료하게 되면, Command의 읽기 혹은 쓰기에 동작에 따라서 DCS3의 Done 신호를 central arbiter로 보내주며 채널을 막고 전송을 완료하게 된다.

그림 6은 마스터에서 마스터로의 데이터의 전송 흐름과 제어 순서를 나타낸다. 만약 Master1에서 Master3로 어떠한 전송이 이루어지려면, ①에서 동작요구가 이루어지고, Arbiter에서 우선순위와 채널의 상태를 확인 후, Master3의 전송을 인정하는 신호와 채널 게이트 제어 신호인 ②를 보내게 된다. 이후에 데이터의 전송이 이루어지는 ③이 전송된다. 데이터의 전송이 완료되면 ④의 done 신호를 arbiter에 보내고, 채널을 닫아 전송을 완료 하게 된다. 그림 7은 다중 작업을 위한 데이터의 흐름과 제어 순서를 나타낸다. 그림의 예제는 Master1에서 Slave3으로, Master3에서 Slave4로 동시에 데이터를 전송하는 순서를 간략히 보여준다. Master1에서 요구 신호인 ①을 CArbiter에 보내고, Master3에서 ①'신호를 CArbiter에 보내게 되면, CArbiter에서 채널의 상태에 따라 수락(accept)신호와 채널 게이트 제어 신호인 ②와 ②'를 보내주게 된다. 이때 ①과 ①'는 Local arbiter에서 우선순위를 정하기 때문에 동시에 요구를 하지 않아 central arbiter에서는 우선순위를 고려하지 않아도 된다. 이후에 데이터의 전송인 ③과 ③'을 각각 전송하게 된다. 전송 완료 후 ④의 Done 신호를 Carbiter로 보낸 후 채널을 닫

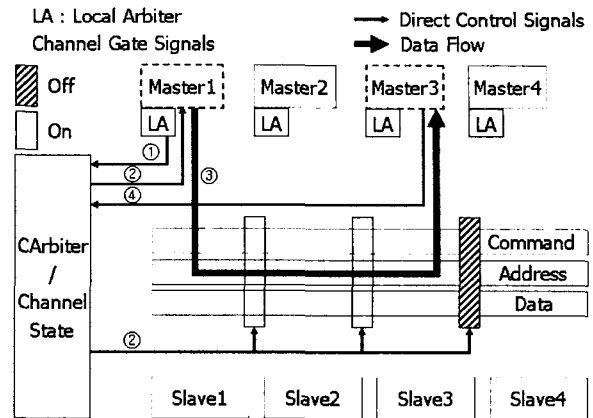


그림 6. Master1에서 Master3으로 데이터 와 직접 제어를 위한 신호의 흐름.
Fig. 6. Data and direct control signal flow from Master1 to Master3.

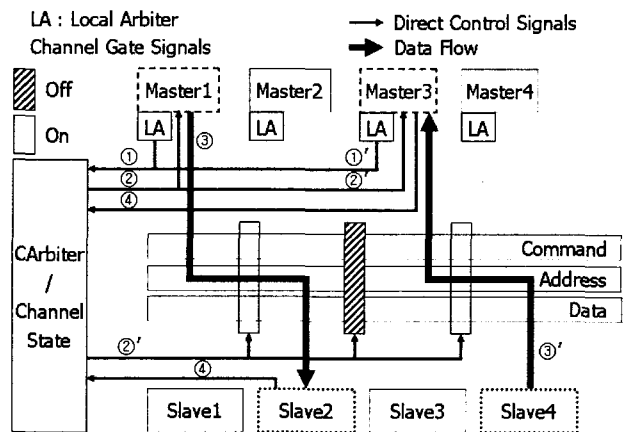


그림 7. 다중작업을 위한 데이터와 직접 제어를 위한 신호의 흐름.
Fig. 7. Data and direct signal control flow for Multitasking.

고 전송을 완료한다. 여기에서의 문제점은 전송 완료 신호인 Done은 요구 신호(request) ①이나 ①'과는 다르게 동시에 발생할 경우가 있기 때문에 이때는 Carbiter에서 우선순위를 제어해야만 한다. 또한 이러한 완료 신호 Done은 마스터와 슬레이브에서 모두 발생 할 수 있다. 제안된 구조에서는 동시에 Done신호가 발생하지 않도록 Carbiter에 done신호의 arbitration을 추가하였다. 이 arbitration은 가장 가까운 마스터와 슬레이브에 우선순위를 두는 방식으로 설계를 하였다.

본 논문에서 제안된 버스구조는 앞 절에서 설명한 것과 같이 세 가지의 arbitration 방식이 필요하게 된다. 버스의 점유율에 따라 TDMA방식의

arbitration을 하는 local arbitration 방식, 시스템 버스에서의 채널의 제어를 위한 central arbitration의 구조를 가지는 방식, 메모리버스에서의 arbitration 방식이 있다. Central arbiter는 단순한 비교방식으로 전체 채널의 승인 여부만을 결정한다. 이 구조는 병목현상이 발생 할 수 있다. 즉, 단일 전송(single transfer)의 경우는 파이프라인의 구조의 특성을 얻기는 어렵다. 다시 말해서, 제안된 버스구조는 일반적으로 많이 쓰이는 시스템(단일 전송이 많이 발생하는 시스템)에는 그 성능향상과 시스템의 효율성을 높일 수는 없다. 하지만, Burst 전송 방식의 특성을 가지는 특정한 시스템에는 적합한 구조이다. 그러므로 제안한 버스구조는 시스템 버스에서 데이터 전송 시 burst의 동작이 빈번하게 일어나는 시스템에 구성하는 것이 좋다. 또한 제안된 버스 구조의 파이프라인은 양방향 구조로 설계 되어있고, 3-state 버퍼를 사용하였기 때문에 배선 효율이 높다. 이때 데이터 버스는 전송의 동기를 맞추는 파이프라인구조로 채택하였다.

라. 시스템 버스와 MBUS의 통합 동작

앞 절에서 설명한 시스템 버스와 메모리 버스의 결합하기 위하여 Mbridge가 필요하다. 이 Mbridge는 시스템 버스에서는 슬레이브의 동작을 하며, 메모리 버스에서는 마스터로 동작한다. 마스터에서 요구를 받은 후, 버퍼를 통하여 데이터를 저장하고 대기하다가 메모리 버스의 동기화된 신호로 출력하게 된다.

이러한 Mbridge를 통하여 메모리 접근을 하기위한 데이터의 흐름과 제어 순서는 그림 8과 같이 이루어진다.

메모리 버스에서 MPEG의 동작이 이루어지는 ①의 요구가 있고, 다른 시간상에서 시스템 버스의 마스터에서 ①'의 요구가 발생하였다면, 메모리 버스의 ②의 승인 신호와 ③의 데이터 쓰기 동작이 이루어진다. 이 MPEG의 동작이 계속 일어나는 동안 시스템 버스의 ②'의 승인 신호와 채널 게이트 제어 신호를 보내고, ③'의 데이터 쓰기 동작이 발생한다. Mbridge는 이러한 요구 신호를 버퍼에 저장하고 다시 메모리 버스의 arbiter로 요구를 하게 되고, 만약 ④의 MPEG동작이 끝나거나, 메모리를 접근할 수 있는 순위를 할당 받게 되면, Mbridge의 버퍼에 저장된 데이터 ⑥ 이 출력되어 메모리를 쓰게 된다.

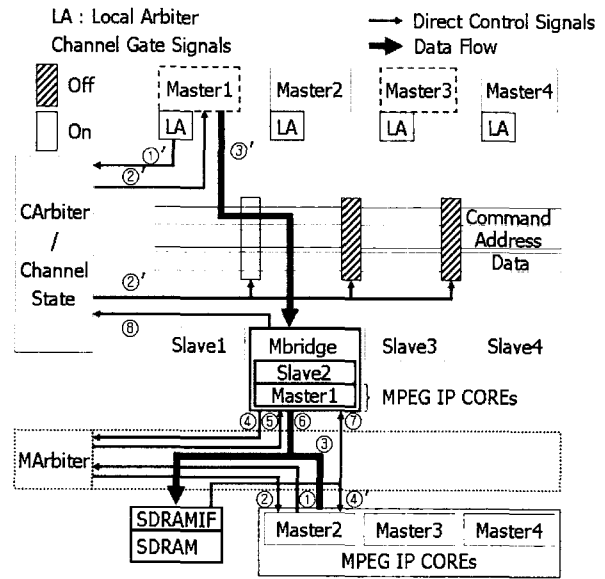


그림 8. 시스템 버스에서 메모리 버스로의 데이터 흐름과 직접 제어 신호의 흐름.
Fig. 8. Data and direct control signals flow of system bus to memory bus.

메모리에 데이터 쓰기가 완료하게 되면, ⑦의 완료 신호를 보내게 되고, 다시 Mbridge를 통하여 ⑧의 완료 신호를 시스템 버스의 Carbiter로 보내면서 전송이 종료된다.

III. 실험

제안된 버스구조는 OCP의 데이터 흐름 신호들 중 기본 신호들을 이용하여 시뮬레이션을 하고 검증하였다. 또한, AMBA와 성능을 비교하기 위하여 AMBA wrapper를 설계하였다. 이러한 비교에서 SDRAM과 같은 외부 메모리를 사용한다고 가정하였다. 전체 시스템은 마스터 4개와 슬레이브 4개로 구성하였다. 마스터의 버스 요구는 inter-communication interval에 의해 정해지며 이는 마스터가 전송 완료 후 몇 버스 사이클 후에 버스사용을 요구 할 것인가를 나타낸다. 시뮬레이션 테스트는 uniform traffic pattern과 non-uniform traffic pattern, random traffic pattern 방식을 사용 하였다. Uniform traffic pattern 방식은 4개의 마스터가 각각의 슬레이브로의 작업량을 동일한 비율로 가지는 방식이다. 설계된 버스 구조는 4개의 채널을 지원하므로 채널의 수를 고려하여 균등하게 25%씩의 작업을 요청하게 설계 하였다. Non-uniform traffic pattern은 마스터가 가장 인근한 슬레이브로 더 많은 작업량을 가지게

표 2. 시뮬레이션 검증을 위한 시스템 및 테스트 환경

Table 2. System and test configuration for simulation verification.

	Addr/Data	Master /Slave	Priority	Burst
AMBA	32/32 bit	4/4	TDMA	16 increments
Proposed Bus	32/32 bit	4/4	TDMA	16 increments

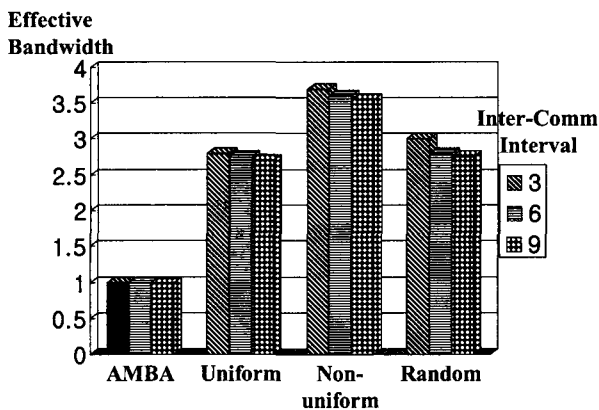


그림 9. 각 패턴에 대한 Effective bandwidth 결과
Fig. 9. Effective bandwidth versus patterns.

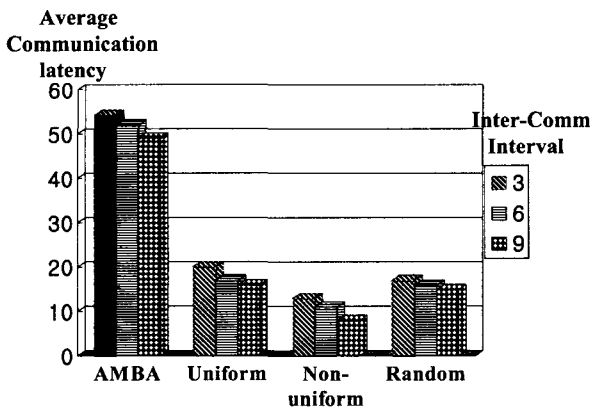


그림 10. 각 패턴에 대한 평균 전송 latency 결과
Fig. 10. Avg. Communication latency versus patterns.

하는 방식이다. 이 방식은 인접한 슬레이브는 85%의 작업량을 주고, 나머지는 균등하게 5%의 작업량을 가지게 함으로써 IP배치가 중요한 버스의 특성을 살리는 방식이다. Random traffic pattern은 마스터가 랜덤하게 슬레이브를 선택하는 방식이다. 표 2는 성능 검증을 위한 시스템 테스트 환경을 보여준다. 성능 비교를 위하여 동일 조건에서 AMBA와 제안된 버스를 테스트 하

였다. 우선순위는 각 버스 모두 TDMA방식을 채택하였고, 마스터 1이 40%, 마스터 2가 30%, 마스터 3가 20%, 그리고 마스터 4가 10%의 점유율을 갖도록 설계하였다. 데이터의 전송은 단일 전송은 고려하지 않고, 정해진 16 increments burst 동작만을 수행한다. 또한 inter-communication Interval을 3, 6, 9로 변화를 주어 실험하였으며, 실험 결과로 effective bandwidth와 latency를 비교한다. Effective bandwidth는 (완료된 전송의 수/사용된 총 버스 사이클)로 정의되고 latency는 마스터가 버스 요구를 보낸 후 승인을 받기까지의 버스 사이클이다.

위의 환경과 조건으로 uniform traffic pattern 과 non-uniform pattern 방식, random traffic pattern으로 4개의 마스터가 총 240번의 16INC burst를 수행 하였을 때의 effective bandwidth와 latency를 AMBA와 비교한 결과를 그림 9와 그림 10에서 보여준다.

AMBA와 비교하여 제안된 버스의 Non-uniform 방식에서 bandwidth는 최고 3.7배 향상하며 latency는 1/4로 감소하는 성능을 보인다. Inter-communication Interval에 따른 차이를 보이는 이유는 Interval이 클수록 더 적은 동시적인 버스 요청이 있기 때문이다. 자주 전송하는 모듈끼리 근접하게 IP를 배치하였기 때문에 Non-uniform 패턴이 가장 뛰어난 성능을 보인다.

IV. 결 론

본 논문은 MPEG 시스템에서 MPEG 동작의 성능 감소 없이 다중 작업이 가능한 새로운 동기 세그먼트 버스 구조를 제안하였다. 제안된 버스 구조는 IP의 재사용 가능한 구조로 이루어져 있다. 직접 제어 신호와 파이프라인 버스 구조는 높은 성능을 얻을 수 있고, 양방향 버스의 구조로 버스 배선을 줄일 수 있다. 이 구조는 3-state 버퍼로 구성되어 있고, 3-state 버퍼의 testability를 해결하고자 단 방향 버스 구조로 대체 할 수 있도록 설계 되었다. 단 방향 구조의 경우 양방향 구조 보다 2배의 버스 배선이 필요하다. 하지만 MPEG 시스템 모바일의 경우 4개 이상의 마스터가 필요 하지 않다고 가정하여 양방향 구조를 채택하여 구현 하였다. 또한, 마스터의 추가가 필요할 때는 Local arbiter의 수정만으로 확장이 가능한 구조이다. 결과는 uniform과 non-uniform, random한 경우를 AMBA와 동일한 조건 환경아래에 시뮬레이션 하였고, non-uniform에서는

약 3.7배의 bandwidth 성능 향상과 1/4배의 latency 감소 특성을 얻었다. AMBA의 Multi-layer 방식은 버스의 배선과 면적을 고려하여 저 전력 구현에 적합하지 않아 비교에서는 제외하였다.

감사의 글

저자들은 시뮬레이션 툴을 지원해준 IDEC에 감사드린다.

참 고 문 헌

- [1] Michael Keating, Pierre Bricaud, "Reuse methodology manual for system-on-a-chip designs", Kluwer Academic Publishers, 1998.
- [2] ARM, "AMBA Specification Overview," [Online document], Available [HTTP://www.arm.com/Pro+Peripherals/AMBA](http://www.arm.com/Pro+Peripherals/AMBA).
- [3] IBM, "CoreConnect Bus Architecture," [Online document], Available http://www.chips.ibm.com/products/coreconnect/docs/cron_wp.pdf.
- [4] Sonics, "Sonics μ Network Technical Overview," [Online document], Available [HTTP: http://www.sonicsinc.com/Documents/Overview.pdf](http://www.sonicsinc.com/Documents/Overview.pdf).
- [5] OCP International Partnership, "Open Core Protocol Specification," [Online document], Available [HTTP://www.ocpip.org](http://www.ocpip.org).
- [6] Kenichiro Anjo, Atsushi Okamura, Masato Motomura, "Wrapper-Based Bus Implementation Techniques for performance Improvement and Cost Reduction," IEEE journal of SOLID-STATE CIRCUITS, Vol.35, Issue:5, pp.804-817, May. 2004.
- [7] Plosila, J, Seceleanu, T. Liljeberg, P, "Implementation of a self-timed segmented bus", IEEE Design & Test of Computers, Vo.20, Issue: 6, pp.44-45. Nov-Dec. 2003.
- [8] Ruiqing Lu, Cheng-Koh Koh, "SAMBA-bus: A high performance bus architecture for system-on-chips," Computer Aided Design, 2003. ICCAD-2003. International Conference on , pp.8-12, Nov. 2003.

저 자 소 개



전 치 훈(학생회원)
 2003년 한림대학교 전자공학과
 학사 졸업.
 2003년~현재 숭실대학교 정보
 통신공학과 석사과정.

<주관심분야 : On-chip 데이터 통신 알고리즘 및
 구조, 저전력 MPEG, SIP/SOC를 위한 IP 집적화
 및 on-chip 데이터 통신>



연 규 성(학생회원)
 2003년 한림대학교 전자공학과
 학사 졸업.
 2003년~현재 숭실대학교 정보
 통신공학과 석사과정.

<주관심분야 : Mobile 시스템 설계, block-wise
 shutdown 기법, 저전력 멀티미디어 시스템, 저전
 력 MPEG 프로세서, motion estimation,
 SIP/SOC를 위한 IP 집적화 및 on-chip 데이터
 통신>



황 태 진(학생회원)
 2003년 한림대학교 전자공학과
 학사 졸업.
 2003년~현재 숭실대학교 정보
 통신공학과 석사과정.

<주관심분야 : 저전력 CMOS 회로 설계, 고속 및
 고성능 DLL/PLL 설계, 고속 clocking 시스템, 아
 날로그 설계 기법>



위 재 경(정회원)
 1988년 연세대학교 물리학과
 학사 졸업.
 1990년 서울대학교 물리학과
 석사 졸업.
 1998년 서울대학교 전자공학과
 박사 졸업.

1990년~2002년 하이닉스전자 메모리연구소
 근무
 2002년~2004년 한림대학교 정보통신공학부
 조교수
 2004년~현재 숭실대학교 정보통신전자공학부
 조교수
 <주관심분야 : 고속 저전력 SoC/SOP 설계 방법
 론, high speed I/O interface, DLL/PLL>