



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0095139
(43) 공개일자 2015년08월20일

(51) 국제특허분류(Int. Cl.)
G06F 13/16 (2006.01) G06F 12/00 (2006.01)
(21) 출원번호 10-2014-0016273
(22) 출원일자 2014년02월12일
심사청구일자 없음

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
권기석
서울특별시 송파구 중대로 24, 108동 702호 (문정동, 웨미리아파트)
김석진
서울특별시 영등포구 도신로29길 28, 104동 602호 (영등포동, 영등포푸르지오)
김도형
경기도 화성시 동탄문화센터로 38, 411동 502호 (반송동, 솔빛마을서해그랑블아파트)
(74) 대리인
리앤목특허법인

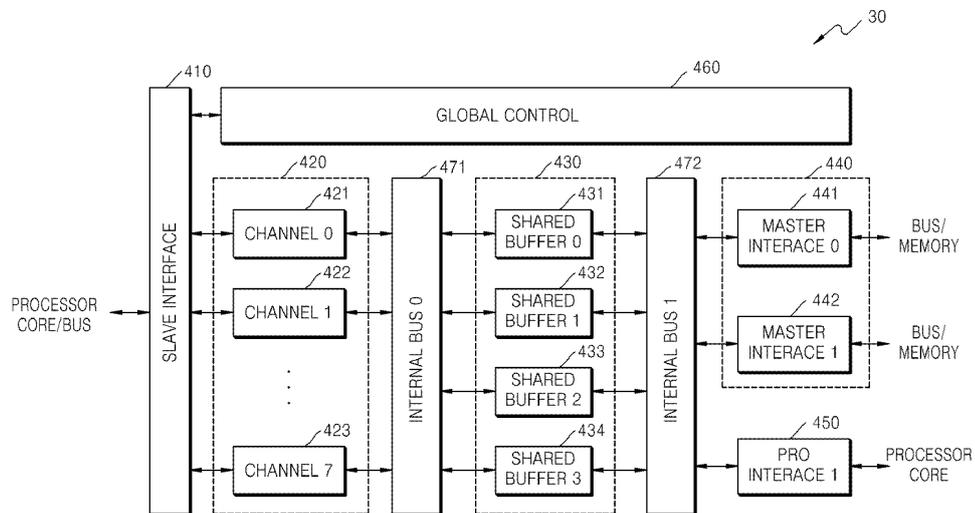
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 채널 버퍼로의 직접 접근이 가능한 직접 메모리 접근 컨트롤러 및 시스템.

(57) 요약

DMA 컨트롤러는, 버퍼에 요청된 작업을 나타내는 정보를 프로세서 코어로부터 직접 수신하고, 상기 버퍼에 저장된 상기 작업에 대응하는 데이터를 상기 프로세서 코어로 전송하거나 상기 작업에 대응하는 데이터를 상기 버퍼에 기록하는 프로세서 인터페이스를 포함한다.

대표도



명세서

청구범위

청구항 1

버퍼에 요청된 제 1 작업을 나타내는 정보를 프로세서 코어로부터 직접 수신하고, 상기 버퍼에 저장된 상기 제 1 작업에 대응하는 데이터를 상기 프로세서 코어로 전송하거나 상기 제 1 작업에 대응하는 데이터를 상기 버퍼에 기록하는 프로세서 인터페이스; 및

상기 프로세서 인터페이스와 연결되고, 상기 버퍼를 적어도 하나 이상 포함하는 버퍼 그룹;을 포함하는 직접 메모리 접근(Direct Memory Access, DMA) 컨트롤러.

청구항 2

제 1 항에 있어서,

복수의 직접 메모리 접근 채널들을 포함하는 채널 그룹;

상기 채널 그룹에 요청된 제 2 작업을 나타내는 정보를 상기 프로세서 코어로부터 수신하고, 상기 채널 그룹에 포함된 상기 채널들 중 적어도 하나 이상에 상기 제 2 작업을 할당하는 슬레이브 인터페이스; 및

상기 버퍼 그룹과 연결되고, 상기 제 2 작업에 기초하여 외부 메모리와 상기 제 2 작업에 대응하는 데이터를 송수신하는 마스터 인터페이스를 적어도 하나 이상 포함하는 마스터 인터페이스 그룹;을 더 포함하는 DMA 컨트롤러.

청구항 3

제 2 항에 있어서,

상기 채널은 제 1 슬레이브 장치에 포함된 상기 데이터를 상기 버퍼에 저장하는 모드 및 상기 버퍼에 저장된 상기 데이터를 제 2 슬레이브 장치로 전송하는 모드 중 적어도 하나 이상을 수행하고,

상기 제 1 슬레이브 장치 및 상기 제 2 슬레이브 장치는 상기 마스터 인터페이스 그룹에 연결되는 DMA 컨트롤러.

청구항 4

제 2 항에 있어서,

상기 제 1 작업을 나타내는 정보는 상기 채널 그룹에 포함된 채널들 중 어느 하나를 지정하는 정보를 포함하고, 상기 프로세서 인터페이스는 상기 버퍼 그룹에 포함된 버퍼들 중 상기 지정된 채널과 기 연결된 버퍼로부터 상기 제 1 작업에 대응하는 데이터를 상기 프로세서 코어로 전송하거나 상기 지정된 채널과 기 연결된 버퍼에 상기 제 1 작업에 대응하는 데이터를 기록하는 DMA 컨트롤러.

청구항 5

제 2 항에 있어서,

상기 프로세서 인터페이스와 상기 슬레이브 인터페이스는 물리적으로 동일한 인터페이스인 DMA 컨트롤러.

청구항 6

제 2 항에 있어서,

상기 버퍼 그룹, 상기 마스터 인터페이스 그룹 및 상기 프로세서 인터페이스와 연결된 내부 버스;를 더 포함하는 DMA 컨트롤러.

청구항 7

제 1 항에 있어서,

상기 버퍼는 선입선출(FIFO) 방식으로 상기 데이터를 입출력하는 DMA 컨트롤러.

청구항 8

제 1 항에 있어서,

상기 버퍼는 임의 접근(random access) 방식으로 상기 데이터를 입출력하는 DMA 컨트롤러.

청구항 9

제 4 항에 있어서,

상기 임의 접근 방식은 기 설정된 패턴에 기초하여 상기 버퍼에 포함된 엔트리로부터 상기 데이터를 입출력하는 제 1 방식, 입력된 스트라이드(stride)에 기초하여 상기 엔트리로부터 상기 데이터를 입출력하는 제 2 방식 및 패턴 레지스터에 정의된 순서 패턴에 기초하여 상기 엔트리로부터 상기 데이터를 입출력하는 제 3 방식 중 어느 하나를 포함하는 DMA 컨트롤러.

청구항 10

제 1 항에 있어서,

상기 버퍼 그룹에 포함된 적어도 2 이상의 버퍼들이 연결되어 더블 버퍼(double buffer)로서 상기 데이터를 입출력하는 DMA 컨트롤러.

청구항 11

제 10 항에 있어서,

상기 더블 버퍼에 포함된 버퍼들 중 일부는 선입선출 방식으로 상기 데이터를 입출력하고, 나머지는 임의 접근 방식으로 상기 데이터를 입출력하는 DMA 컨트롤러.

청구항 12

직접 메모리 접근(Direct Memory Access, DMA) 컨트롤러; 및

상기 직접 메모리 접근 컨트롤러와 연결된 프로세서 코어;를 포함하는 시스템에서,

상기 직접 메모리 접근 컨트롤러는

버퍼에 요청된 제 1 작업을 나타내는 정보를 상기 프로세서 코어로부터 직접 수신하고, 상기 버퍼에 저장된 상기 제 1 작업에 대응하는 데이터를 상기 프로세서 코어로 전송하거나 상기 제 1 작업에 대응하는 데이터를 상기 버퍼에 기록하는 프로세서 인터페이스; 및

상기 프로세서 인터페이스와 연결되고, 상기 버퍼를 적어도 하나 이상 포함하는 버퍼 그룹;을 포함하는 시스템.

청구항 13

제 12 항에 있어서,

상기 직접 메모리 접근 컨트롤러는

복수의 직접 메모리 접근 채널들을 포함하는 채널 그룹;

상기 채널 그룹에 요청된 제 2 작업을 나타내는 정보를 상기 프로세서 코어로부터 수신하고, 상기 채널 그룹에 포함된 상기 채널들 중 적어도 하나 이상에 상기 제 2 작업을 할당하는 슬레이브 인터페이스; 및

상기 버퍼 그룹과 연결되고, 상기 제 2 작업에 기초하여 외부 메모리와 상기 제 2 작업에 대응하는 데이터를 송수신하는 마스터 인터페이스를 적어도 하나 이상 포함하는 마스터 인터페이스 그룹;을 더 포함하는 시스템.

청구항 14

제 13 항에 있어서,

상기 채널은 제 1 슬레이브 장치에 포함된 상기 데이터를 상기 버퍼에 저장하는 모드 및 상기 버퍼에 저장된 상기 데이터를 상기 제 2 슬레이브 장치로 전송하는 모드 중 적어도 하나 이상을 수행하고,

상기 제 1 슬레이브 장치 및 상기 제 2 슬레이브 장치는 상기 마스터 인터페이스 그룹에 연결되는 시스템.

청구항 15

제 13 항에 있어서,

상기 프로세서 인터페이스와 상기 슬레이브 인터페이스는 물리적으로 동일한 인터페이스인 DMA 컨트롤러.

청구항 16

제 12 항에 있어서,

상기 버퍼는 선입선출(FIFO) 방식 또는 임의 접근(random access) 방식으로 상기 데이터를 입출력하는 시스템.

청구항 17

제 12 항에 있어서,

상기 임의 접근 방식은 기 설정된 패턴에 기초하여 상기 버퍼에 포함된 엔트리로부터 상기 데이터를 입출력하는 제 1 방식, 입력된 스트라이드(stride)에 기초하여 상기 엔트리로부터 상기 데이터를 입출력하는 제 2 방식 및 패턴 레지스터에 정의된 순서 패턴에 기초하여 상기 엔트리로부터 상기 데이터를 입출력하는 제 3 방식 중 어느 하나를 포함하는 시스템.

청구항 18

제 12 항에 있어서,

상기 버퍼 그룹에 포함된 적어도 2 이상의 버퍼들이 연결되어 더블 버퍼(double buffer)로서 상기 데이터를 입출력하는 시스템.

청구항 19

제 18 항에 있어서,

상기 더블 버퍼에 포함된 버퍼들 중 일부는 선입선출 방식으로 상기 데이터를 입출력하고, 나머지는 임의 접근 방식으로 상기 데이터를 입출력하는 시스템.

청구항 20

제 12 항에 있어서,

상기 버퍼 그룹에 포함된 2 이상의 버퍼들 각각은 상기 채널로부터 전송된 신호에 기초하여 선입선출(FIFO) 방식으로 상기 데이터를 입출력하거나 서로 연결된 더블 버퍼(double buffer)로서 상기 데이터를 입출력하는 시스템.

발명의 설명

기술 분야

[0001] 채널 버퍼로의 직접 접근이 가능한 직접 메모리 접근 컨트롤러 및 시스템을 개시한다.

배경 기술

[0002] 직접 메모리 접근 방식(Direct Memory Access, 이하 DMA)은 중앙처리장치(Central Processing Unit, 이하 CPU)를 거치지 않고 외부 메모리와 입/출력 장치 사이에서 수행되는 데이터 전송 방법의 하나이다. DMA를 사용하기 위해서는 시스템 버스 상에 DMA 컨트롤러(Direct Memory Access Controller)가 추가되어야 한다.

[0003] DMA 컨트롤러는 CPU로부터 시스템의 제어 신호를 넘겨받아 CPU의 기능 중 일부를 대신 처리한다. 즉, CPU는 데이터 전송 동작을 DMA 컨트롤러에 일임하고, DMA 컨트롤러는 CPU를 대신하여 데이터 전송 동작을 처리한다. DMA

컨트롤러는 CPU를 거치지 않고 데이터를 외부 메모리로 직접 전송하고, 모든 데이터 전송이 완료되면 DMA 컨트롤러는 인터럽트를 생성하고, 생성된 인터럽트를 CPU로 전달한다.

[0004] 최근에는 DMA 컨트롤러를 포함하는 시스템에서 데이터 스트림(이하, '데이터'라고 함)을 전달하는데 소요되는 지연 시간(latency)을 단축시키고, 외부 메모리에 접근함에 따라 소비되는 전력을 줄이기 위한 연구가 진행되고 있다.

발명의 내용

해결하려는 과제

[0005] 채널 버퍼로의 직접 접근이 가능한 직접 메모리 접근 컨트롤러 및 시스템을 제공하는데 있다. 해결하려는 기술적 과제는 상기된 바와 같은 기술적 과제들로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0006] 일 측면에 따른 DMA 컨트롤러는, 버퍼에 요청된 제 1 작업을 나타내는 정보를 프로세서 코어로부터 직접 수신하고, 상기 버퍼에 저장된 상기 제 1 작업에 대응하는 데이터를 상기 프로세서 코어로 전송하거나 상기 제 1 작업에 대응하는 데이터를 상기 버퍼에 기록하는 프로세서 인터페이스; 및 상기 프로세서 인터페이스와 연결되고, 상기 버퍼를 적어도 하나 이상 포함하는 버퍼 그룹;을 포함한다.

[0007] 다른 측면에 따른 시스템은, 직접 메모리 접근(Direct Memory Access, DMA) 컨트롤러; 및 상기 직접 메모리 접근 컨트롤러와 연결된 프로세서 코어;를 포함하는 시스템에서, 상기 직접 메모리 접근 컨트롤러는 버퍼에 요청된 제 1 작업을 나타내는 정보를 상기 프로세서 코어로부터 직접 수신하고, 상기 버퍼에 저장된 상기 제 1 작업에 대응하는 데이터를 상기 프로세서 코어로 전송하거나 상기 제 1 작업에 대응하는 데이터를 상기 버퍼에 기록하는 프로세서 인터페이스; 및 상기 프로세서 인터페이스와 연결되고, 상기 버퍼를 적어도 하나 이상 포함하는 버퍼 그룹;을 포함한다.

발명의 효과

[0008] 상술한 바에 따르면, 프로세서 코어는 DMA 컨트롤러 내부의 버퍼에 직접 접근할 수 있는바, DMA 컨트롤러의 내부 버퍼와 스크래치 패드 메모리 간의 데이터 전송 과정을 생략할 수 있다. 따라서, 프로세서 시스템과 외부 메모리 간에 데이터를 전송하는데 걸리는 지연 시간과 전력 소모를 줄일 수 있다.

[0009] 또한, 프로세서 코어는 선입선출 방식으로 동작하는 버퍼에 직접 접근할 수 있는바, 프로세서 코어가 별도로 버퍼의 엔트리 주소를 계산할 필요가 없다. 따라서, 프로세서 시스템의 성능이 향상될 수 있다.

[0010] 또한, 버퍼가 임의 접근 방식으로 동작하더라도, DMA 컨트롤러는 프로세서 코어로 주소 생성 패턴을 제공할 수 있는바, 프로세서 코어가 별도로 버퍼의 엔트리 주소를 계산할 필요가 없다. 따라서, 프로세서 시스템의 성능이 향상될 수 있다.

도면의 간단한 설명

[0011] 도 1은 일 실시예에 따른 디지털 시스템의 일 예를 도시한 구성도이다.
 도 2는 DMA 컨트롤러에 포함된 구성들을 도시한 도면이다.
 도 3은 DMA 컨트롤러가 데이터를 송수신하는 방법을 나타내는 흐름도이다.
 도 4는 일 실시예에 따른 DMA 컨트롤러의 일 예를 도시한 구성도이다.
 도 5는 일 실시예에 따른 DMA 컨트롤러가 제 2 모드로 설정되어 데이터를 송수신하는 일 예를 나타내는 흐름도이다.
 도 6은 일 실시예에 따른 DMA 컨트롤러의 다른 예를 도시한 구성도이다.
 도 7은 일 실시예에 따른 DMA 컨트롤러의 또 다른 예를 도시한 구성도이다.
 도 8은 일 실시예에 따른 프로세서 코어에서 DMA 컨트롤러에 포함된 버퍼에 접근하기 위한 명령어 포맷의 일 예를 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0012] 이하에서는 도면을 참조하여 본 발명의 실시예들을 상세히 설명한다. 본 발명의 하기 실시예는 본 발명을 구체화하기 위한 것일 뿐 본 발명의 권리 범위를 제한하거나 한정하지 않는다. 또한, 본 발명의 상세한 설명 및 실시예로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 용이하게 유추할 수 있는 것은 본 발명의 권리범위에 속하는 것으로 해석된다.
- [0013] 도 1은 일 실시예에 따른 디지털 시스템의 일 예를 도시한 구성도이다.
- [0014] 도 1을 참조하면, 디지털 시스템(1)은 프로세서 시스템(10), 버스(60), 외부 메모리(70) 및 디스플레이 컨트롤러(80)를 포함한다. 그리고, 프로세서 시스템(10)은 프로세서 코어(20) 및 DMA 컨트롤러(30)를 포함한다.
- [0015] 도 1에 도시된 디지털 시스템(1)에는 본 실시예와 관련된 구성요소들만이 도시되어 있다. 따라서, 도 1에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음을 본 실시예와 관련된 기술분야에서 통상의 지식을 가진 자라면 이해할 수 있다.
- [0016] 프로세서 시스템(10)은 소정의 작업을 처리한다. 여기에서, 소정의 작업은 디지털 시스템(1) 외부로부터 전달된 신호에 기초한 작업을 의미할 수도 있고, 상기 외부로부터 전달된 신호에 기초한 작업을 처리하는 과정에서 발생하는 중간 작업을 의미할 수도 있다.
- [0017] 본 발명의 일 실시예에 따른 프로세서 시스템(10)은 DMA 컨트롤러(30) 및 DMA 컨트롤러(30)와 연결된 프로세서 코어(20)를 포함한다.
- [0018] 도 1에는 도시되지 않았으나, 프로세서 시스템(10)은 스크래치 패드 메모리(미도시) 또는 캐시(미도시)를 더 포함할 수 있다. 스크래치 패드 메모리(미도시) 또는 캐시(미도시)는 외부 메모리(70)와는 별도로 프로세서 시스템(10) 내에 구비되며 데이터를 임시 저장할 수 있는 장치라면 제한 없이 해당 될 수 있다. 다시 말해, 도 1에 도시된 스크래치 패드 메모리(미도시)와 캐시(미도시)는 모두 동종의 스크래치 패드 메모리가 될 수도 있고, 또는 동종의 캐시가 될 수도 있다.
- [0019] 디지털 시스템(1)은 프로세서 시스템(10) 외에도 버스(60), 외부 메모리(70) 및 디스플레이 컨트롤러(80)를 더 포함할 수 있다. 여기에서, 버스(60)는 프로세서 시스템(10)과 외부 장치(예를 들어, 외부 메모리(70) 또는 디스플레이 컨트롤러(80))를 연결함으로써 데이터 또는 신호의 송수신이 가능하도록 한다. 도 1을 참조하면, 디지털 시스템(1)은 프로세서 시스템(10)과 외부 장치를 연결하는 버스(60)를 별도로 구비한 것으로 도시하였으나, 버스(60)가 구비되지 않은 상태에서 프로세서 시스템(10)과 외부 장치가 직접 연결될 수도 있다.
- [0020] 또한, 외부 메모리(70)는 SRAM, DRAM과 같은 특정 장치에 한정되지 않으며, 슬레이브 장치라면 제한 없이 해당 될 수 있다. 또한, 디스플레이 컨트롤러(80)는 디지털 시스템(1)을 포함하는 디바이스에 구비된 디스플레이 장치를 제어하는 기능을 수행한다.
- [0021] 도 1에 도시된 디지털 시스템(1)에서 데이터가 전달되고 처리되는 흐름은 다음과 같다.
- [0022] 먼저, DMA 컨트롤러(30)는 외부 메모리(70)에서 DMA 컨트롤러(30) 내부에 포함된 버퍼로 데이터를 전송한다. 여기에서, 데이터는 프로세서 코어(20)가 처리하는 작업과 관련된 데이터를 의미하고, 외부 메모리(70)는 데이터가 저장되어 있는 저장 장치를 의미한다.
- [0023] 그 후에, DMA 컨트롤러(30)는 버퍼에 저장된 데이터를 프로세서 코어(20) 내부에 포함된 레지스터 파일로 데이터를 전송한다.
- [0024] 그 후에, 프로세서 코어(20)는 레지스터 파일에 기록된 데이터를 이용하여 작업을 수행하고, 그 결과를 다시 레지스터 파일에 기록한다.
- [0025] 그 후에, 프로세서 코어(20)는 메모리 쓰기 명령을 사용하여 DMA 컨트롤러(30) 내부에 포함된 버퍼로 데이터를 전송한다.
- [0026] 그 후에, DMA 컨트롤러(40)는 버퍼에 저장된 데이터를 외부 메모리(70)로 전송한다.
- [0027] 만약, 프로세서 시스템(10)이 스크래치 패드 메모리(미도시) 또는 캐시(미도시)를 더 포함한 경우, 프로세서 코어(20)와 DMA 컨트롤러(30) 사이의 데이터 송수신은 스크래치 패드 메모리(미도시) 또는 캐시(미도시)를 경유하여 수행될 수 있다.

- [0028] 한편, 일반적인 DMA 컨트롤러(30)의 내부에서 데이터가 전송되는 흐름은 도 2를 참조하여 후술한다.
- [0029] 도 2는 DMA 컨트롤러에 포함된 구성들을 도시한 도면이다.
- [0030] 도 2를 참조하면, DMA 컨트롤러(30)는 슬레이브 인터페이스(310), 적어도 하나 이상의 채널(321 내지 323), 적어도 하나 이상의 마스터 인터페이스(331 내지 332), 내부 버스(340) 및 전역 제어부(350)를 포함한다. 여기서, 하나의 채널(321)에는 채널 제어부(3211) 및 버퍼(3212)가 포함된다.
- [0031] 채널 제어부(3211)는 슬레이브 인터페이스(310)로부터 전달된 전송 정보를 저장하고, 채널(321)을 제어한다. 여기에서, 전송 정보는 프로세서 코어(20)가 수행하는 작업에 대응하는 데이터에 관한 정보를 포함한다. 일반적으로, 전송 정보에는 소스(source) 메모리 주소, 대상(destination) 메모리 주소, 데이터의 기본 단위(예를 들어, 32 비트 또는 64 비트), 총 데이터의 개수, 채널의 우선 순위, 버스 설정 등이 포함된다. 버퍼(3212)는 프로세서 코어(20)가 외부 메모리(70)로 전송할 데이터 또는 외부 메모리(70)가 프로세서 코어(20)로 전송할 데이터를 저장한다.
- [0032] 하나의 채널(321)에 대한 전송 설정은 프로세서 코어(20)가 슬레이브 인터페이스(310)를 통하여 채널 제어부(3211)에 포함된 레지스터들에 값을 기록하는 방식으로 이루어진다.
- [0033] DMA 컨트롤러(30)에 포함된 적어도 하나 이상의 채널(321 내지 323)은 내부 버스(340)를 통하여 적어도 하나 이상의 마스터 인터페이스(331 내지 332)를 공유한다. 적어도 하나 이상의 마스터 인터페이스(331 내지 332)는 외부 버스(60)와 연결되고, 버퍼(3212)로부터 외부 메모리(70)로의 데이터 전송 또는 외부 메모리(70)로부터 버퍼(3212)로의 데이터 전송을 수행한다. 한편, DMA 컨트롤러(30)에 포함된 마스터 인터페이스(331 내지 332)는 총 2 개인 것으로 도시되었으나, 개수에 한정되지 않는다.
- [0034] 전역 제어부(350)는 DMA 컨트롤러(30)에 대한 전반적인 제어를 담당한다. 예를 들어, 전역 제어부(350)는 DMA 컨트롤러(30)에 포함된 적어도 하나 이상의 채널(321 내지 323)을 동시에 턴 온-오프(turn on-off)할 수 있다.
- [0035] 도 2에 도시된 DMA 컨트롤러(30)가 데이터를 송수신하는 일 예는 도 3을 참조하여 후술한다.
- [0036] 도 3은 DMA 컨트롤러가 데이터를 송수신하는 방법을 나타내는 흐름도이다.
- [0037] 도 3을 참조하면, DMA 컨트롤러(30)가 데이터를 송수신하는 방법은 도 2에 도시된 DMA 컨트롤러(30)에서 시계열적으로 처리되는 단계들로 구성된다. 따라서, 이하에서 생략된 내용이라 하더라도 도 1 및 도 2에 도시된 디지털 시스템(1) 또는 DMA 컨트롤러(30)에 관하여 이상에서 기술된 내용은 도 3의 데이터를 송수신하는 방법에도 적용됨을 알 수 있다.
- [0038] 이하에서는, DMA 컨트롤러(30)에 포함된 적어도 하나 이상의 채널(321 내지 323) 중에서 제 1 채널(321)을 통하여 데이터가 송수신되는 것으로 가정하여 설명한다.
- [0039] S310에서, 프로세서 코어(20)는 슬레이브 인터페이스(310)를 통하여 제 1 채널(321)에 전송 정보를 전달하고, 제 1 채널(321)을 활성화(enable)한다. 여기에서, 전송 정보에는 소스 메모리 주소, 대상 메모리 주소, 데이터의 기본 단, 총 데이터의 개수, 채널의 우선 순위, 버스 설정 등이 포함된다.
- [0040] S320에서, 제 1 채널(321)은 외부 메모리(70) 중 소스 메모리 주소가 가리키는 외부 메모리와 연결된 마스터 인터페이스 A(331)에 전송 정보를 전달하고, 전송 정보에 대응하는 데이터를 전송할 것을 요청한다.
- [0041] S330에서, 마스터 인터페이스 A(331)는 소스 메모리 주소가 가리키는 외부 메모리에서 데이터를 읽어온 후, 버퍼(3212)에 데이터를 저장한다.
- [0042] S340에서, 제 1 채널(321)의 버퍼(3212)에 데이터가 저장되면, 제 1 채널(321)은 외부 메모리(70) 중 대상 메모리 주소가 가리키는 외부 메모리와 연결된 마스터 인터페이스 B(332)에 전송 정보를 전달하고, 전송 정보에 대응하는 데이터를 전송받을 것을 요청한다.
- [0043] S350에서, 마스터 인터페이스 B(332)는 대상 메모리 주소가 가리키는 외부 메모리로 버퍼(3212)에 저장된 데이터를 전송한다.
- [0044] S360에서, 제 1 채널(321)은 설정된 데이터 전송 과정이 완료되었는지 판단한다. 만약, 데이터 전송이 완료되지 않은 경우에는 S320으로 진행하고, 데이터 전송이 완료된 경우에는 S370으로 진행한다.
- [0045] S370에서, 제 1 채널(321)은 비활성화된다.

- [0046] 상술한 바에 따르면, 마스터 인터페이스 A(331) 및 마스터 인터페이스 B(332)는 서로 다른 마스터 인터페이스인 것으로 설명하였으나, 전송 정보에 포함된 소스 메모리 주소 및 대상 메모리 주소에 따라 동일한 마스터 인터페이스가 될 수도 있다.
- [0047] 버퍼(3212)에 저장된 모든 데이터가 대상 메모리 주소가 가리키는 외부 메모리로 전송되면 제 1 채널(321)은 비활성화(disable)된다.
- [0048] 다시 도 2를 참조하면, 도 1 및 도 2를 참조하여 상술한 각 과정들이 수행되는 동안에 데이터는 다수의 버퍼 또는 레지스터 파일들을 통과한다. 따라서, 외부 메모리(70)에 저장된 데이터가 프로세서 코어(20)에 포함된 레지스터 파일에 도달하는데 까지 걸리는 지연 시간(latency) 및 레지스터 파일에 기록된 데이터가 외부 메모리(70)에 도달하는데 까지 걸리는 지연 시간이 길어지며, 데이터를 전송하는데 소모되는 전력도 커진다.
- [0049] 이에 반하여, 본 발명의 일 실시예에 따른 프로세서 코어(20)는 DMA 컨트롤러(30)에 포함된 버퍼에 직접 접근함으로써, 프로세서 코어(20)와 외부 메모리(70)가 데이터를 송수신하는데 걸리는 지연 시간과 전력 소모를 줄일 수 있다.
- [0050] 본 발명의 일 실시예에 따른 DMA 컨트롤러(30)의 일 예는 도 3 내지 도 5를 참조하여 후술한다.
- [0051] 도 4는 일 실시예에 따른 DMA 컨트롤러의 일 예를 도시한 구성도이다.
- [0052] 도 4를 참조하면, DMA 컨트롤러(30)는 슬레이브 인터페이스(410), 채널 그룹(420), 버퍼 그룹(430), 마스터 인터페이스 그룹(440), 프로세서 인터페이스(450), 전역 제어부(460), 제 1 내부 버스(471) 및 제 2 내부 버스(472)를 포함한다.
- [0053] 도 4에 도시된 DMA 컨트롤러(30)에는 본 실시예와 관련된 구성요소들만이 도시되어 있다. 따라서, 도 4에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음을 본 실시예와 관련된 기술분야에서 통상의 지식을 가진 자라면 이해할 수 있다.
- [0054] 슬레이브 인터페이스(410)는 프로세서 코어(20)로부터 채널 그룹(420)에 요청된 제 1 작업을 나타내는 정보를 수신하고, 채널 그룹(420)에 포함된 채널들(421 내지 423) 중 적어도 하나 이상에 제 1 작업을 할당한다. 여기에서, 제 1 작업은 외부 메모리(70)에 저장된 데이터를 수신할 것을 요청하는 작업 또는 외부 메모리(70)로 데이터를 전송할 것을 요청하는 작업을 의미한다.
- [0055] 다시 말해, 프로세서 코어(20)는 프로세서 코어(20)에서 수행할 작업에 필요한 데이터를 외부 메모리(70)에서 가져오라는 요청 또는 프로세서 코어(20)에서 수행된 작업의 결과로 생성된 데이터를 외부 메모리(70)에 기록하라는 요청을 슬레이브 인터페이스(410)에 전달한다. 슬레이브 인터페이스(410)는 프로세서 코어(20)로부터 전송된 요청에 기초하여, 채널 그룹(420)에 포함된 채널들(421 내지 423) 중 적어도 하나 이상에 프로세서 코어(20)의 요청을 전달한다. 도 4에 도시된 슬레이브 인터페이스(410)는 도 2에 도시된 슬레이브 인터페이스(310)와 동일한 역할을 수행한다. 따라서, 이하에서 슬레이브 인터페이스(410)에 대한 자세한 설명은 생략한다.
- [0056] 채널 그룹(420)은 복수의 채널들을 포함한다. 다시 말해, 채널 그룹(420)은 복수의 직접 메모리 접근 채널들(421 내지 423)을 포함한다. 도 4의 채널 그룹(420)에 포함된 채널들(421 내지 423)은 도 2에 도시된 채널 제어부들과 동일한 역할을 수행한다. 따라서, 이하에서 채널들(321 내지 323)에 대한 자세한 설명은 생략한다. 또한, 채널 그룹(420)은 총 8개의 채널들(421 내지 423)을 포함하는 것으로 도시하였으나, 이에 한정되지 않는다.
- [0057] 버퍼 그룹(430)은 마스터 인터페이스 그룹(440) 및 프로세서 인터페이스(450)와 연결되고, 데이터를 저장하는 적어도 하나 이상의 버퍼들(431 내지 434)을 포함한다. 여기에서, 각각의 버퍼에는 데이터를 저장할 수 있는 복수의 엔트리가 포함되어 있다. 도 4의 버퍼 그룹(430)에 포함된 버퍼들(431 내지 434)은 도 2에 도시된 버퍼들과 동일한 역할을 수행한다. 따라서, 이하에서 버퍼들(431 내지 434)에 대한 자세한 설명은 생략한다. 또한, 버퍼 그룹(430)은 총 4개의 버퍼들(431 내지 434)을 포함하는 것으로 도시하였으나, 이에 한정되지 않는다.
- [0058] 본 발명의 일 실시예에 따른 버퍼 그룹(430)은 채널 그룹(420)과 분리되어 있다. 다시 말해, 도 2를 참조하여 상술한 DMA 컨트롤러(30)는 하나의 채널에 하나의 버퍼가 매핑되어 있으나, 도 4에 도시된 DMA 컨트롤러(30)는 채널 그룹(420)에 포함된 복수의 채널들(421 내지 423)이 버퍼들(431 내지 434)을 공유한다. 따라서, 도 4에 도시된 DMA 컨트롤러(30)는 채널 그룹(420)이 버퍼들(431 내지 434) 공유하기 위한 제 1 내부 버스(471)와 버퍼들(431 내지 434)이 마스터 인터페이스들(441 내지 442)을 공유하기 위한 제 2 내부 버스(472)를 포함한다.

- [0059] DMA 컨트롤러(30)에 포함된 채널들(421 내지 423)이 버퍼들(431 내지 434)을 공유함으로써, 채널들(421 내지 423)은 활성화된 채널의 개수에 따라 효율적으로 버퍼들(431 내지 434)을 분배해서 사용할 수 있다. 또한, 버퍼들(431 내지 434)은 그들 중 일부가 연결되어 더블 버퍼(double buffer), 트리플 버퍼(triple buffer)와 같은 형태의 데이터 저장 구조를 형성할 수 있다.
- [0060] 이하에서, 본 발명의 일 실시예에 따른 버퍼 그룹(430)에 포함된 버퍼들(431 내지 434)이 동작하는 3가지 방식을 설명한다. 이하의 방식들에서, 단일 버퍼가 동작하는 예는 버퍼들(431 내지 434) 중에서 버퍼(431)가 동작하는 것으로 가정하고, 적어도 2 이상의 버퍼들이 더블 버퍼로 동작하는 예는 버퍼들(431 내지 434) 중에서 2개의 버퍼(431, 432)가 더블 버퍼를 형성하는 것으로 가정하여 설명한다. 그러나, 적어도 2 이상의 버퍼들이 논리적인 단일 버퍼로서, 후술할 선입선출(FIFO) 방식으로 데이터를 입출력할 수도 있다.
- [0061] 일 예로, 버퍼(431)는 선입선출(FIFO) 방식으로 데이터를 입출력할 수 있다.
- [0062] 버퍼(431)가 선입선출(FIFO) 방식으로 동작하는 경우에, 외부 메모리(70)에서 프로세서 코어(20)로 데이터가 전달되는 과정은 다음과 같다. 마스터 인터페이스(441)는 버퍼(431)의 엔트리들에 데이터를 차례대로 기록하고, 프로세서 인터페이스(450)는 엔트리들에 기록된 데이터를 차례대로 가져간다.
- [0063] 만약, 상기 동작이 수행되는 과정 중에 버퍼(431)에 빈 엔트리가 충분히 존재하지 않으면, 마스터 인터페이스(441)는 외부 메모리(70)에 대한 데이터 전송 요청을 하지 않고, 충분한 양의 빈 엔트리가 발생될 때까지 대기한다. 만약, 상기 동작이 수행되는 과정 중에 버퍼(431)에 충분한 데이터가 기록되어 있지 않으면, 프로세서 인터페이스(450)는 프로세서 코어(20)에 이를 알리는 신호를 전송하고, 프로세서 코어(20)는 버퍼(431)의 엔트리들에 데이터가 충분히 기록될 때까지 대기한다.
- [0064] 버퍼(431)가 선입선출(FIFO) 방식으로 동작하는 경우에, 프로세서 코어(20)에서 외부 메모리(70)로 데이터가 전달되는 과정은 다음과 같다. 프로세서 인터페이스(450)에서 버퍼(431)의 엔트리에 데이터를 차례대로 기록하고, 마스터 인터페이스(441)는 엔트리에 기록된 데이터를 차례대로 가져간다.
- [0065] 만약, 상기 동작이 수행되는 과정 중에 버퍼(431)에 빈 엔트리가 충분히 존재하지 않으면, 프로세서 인터페이스(450)는 프로세서 코어(20)에 이를 알리는 신호를 전송하고, 프로세서 코어(20)는 충분한 양의 빈 엔트리가 발생될 때까지 대기한다. 만약, 상기 동작이 수행되는 과정 중에 버퍼(431)에 충분한 데이터가 기록되어 있지 않으면, 마스터 인터페이스(441)는 외부 메모리(70)에 대한 데이터 수신 요청을 하지 않고, 버퍼(431)의 엔트리들에 데이터가 충분히 기록될 때까지 대기한다.
- [0066] 다른 예로, 버퍼(431)는 임의 접근(random access) 방식으로 데이터를 입출력할 수 있다. 여기에서, 임의 접근 방식은 기 설정된 패턴에 기초하여 버퍼(431)에 포함된 엔트리로부터 데이터를 입출력하는 제 1 방식, 입력된 스트라이드(stride)에 기초하여 엔트리로부터 데이터를 입출력하는 제 2 방식 및 패턴 레지스터에 정의된 순서 패턴에 기초하여 엔트리로부터 데이터를 입출력하는 제 3 방식을 포함한다.
- [0067] 버퍼(431)가 임의 접근(random access) 방식으로 동작하는 경우에, 버퍼(431)는 레지스터 파일이나 SRAM처럼 접근할 엔트리를 지정하기 위한 주소를 입력으로 받아야 한다. DMA 컨트롤러(30)가 설계된 방식에 따라서, DMA 컨트롤러(30)는 소정의 규칙에 의하여 상기 주소를 계속 생성할 수도 있다.
- [0068] DMA 컨트롤러(30)가 버퍼 엔트리 주소 생성 패턴을 제공하도록 설계되어 있다고 가정하면, 프로세서 코어(20)는 슬레이브 인터페이스(410)를 통하여 채널을 설정할 때 다수의 버퍼 엔트리 주소 생성 패턴들 중에서 어느 하나를 선택할 수 있다. 여기에서, 버퍼 엔트리 주소 생성 패턴은 아래의 3가지 방식들 중 어느 하나로 생성될 수 있다. 이하, 버퍼(431)가 8개의 엔트리들을 포함하고 있고, 각각의 엔트리들에 0부터 오름차순으로 자연수가 부여되었다고 가정하고 상기 3가지 방식들을 구체적으로 설명한다.
- [0069] 제 1 방식은 기 설정된 패턴에 기초하여 버퍼(431)에 포함된 엔트리로부터 데이터를 입출력하는 방식을 의미한다. 예를 들어, 기 설정된 패턴에는 '0→1→2→3→4→5→6→7→0→...'와 같은 순차적인(sequential) 패턴, '0→4→2→6→1→5→3→7→0→...'와 같은 비트 반전(bit reverse) 패턴, '0→2→4→6→1→3→5→7→0→...'와 같은 패턴이 해당될 수 있다.
- [0070] 제 2 방식은 입력된 스트라이드(stride)에 기초하여 버퍼(431)에 포함된 엔트리로부터 데이터를 입출력하는 방식을 의미한다. 예를 들어, 프로그래머가 스트라이드로 K를 설정하였다고 가정하면, '0→K%N→(K*2)%N→(K*3)%N→(K*4)%N→...'와 같은 순서로 엔트리로부터 데이터를 입출력할 수 있다. 여기에서, N은 버퍼의 엔트리 개수(예를 들어, 8)를 의미하고, %는 나눗셈의 나머지를 구하는 연산을 의미한다.

- [0071] 제 3 방식은 패턴 레지스터에 정의된 순서 패턴에 기초하여 버퍼(431)에 포함된 엔트리로부터 데이터를 입출력하는 방식을 의미한다. 예를 들어, 프로그래머가 패턴 레지스터에 순서 패턴을 '0x75316420'로 설정하였다고 가정하면, '7→5→3→1→6→4→2→0→7→...'와 같은 순서로 엔트리로부터 데이터를 입출력할 수 있다.
- [0072] 또 다른 예로, 버퍼들(431 내지 434) 중 적어도 2 이상의 버퍼들이 연결되어 더블 버퍼로 데이터를 입출력할 수 있다. 예를 들어, 2개의 버퍼들(431, 432)이 연결되어 더블 버퍼로 데이터를 입출력할 수 있다.
- [0073] 이때, 더블 버퍼를 구성하는 각각의 버퍼는 서로 다른 방식으로 데이터를 입출력할 수 있다. 예를 들어, 2개의 버퍼들(431, 432)이 연결되어 하나의 더블 버퍼를 구성한다고 가정하면, 제 1 버퍼(431)는 선입선출 방식으로 데이터를 입출력하고, 제 2 버퍼(432)는 임의 접근 방식으로 데이터를 입출력할 수 있다. 그리고, 더블 버퍼는 제 1 버퍼(431) 및 제 2 버퍼(432)를 엔트리로 갖는 단일 버퍼로서 선입선출 방식으로 동작할 수 있다.
- [0074] 버퍼들(431, 432)이 더블 버퍼로 동작하는 경우에, 외부 메모리(70)에서 프로세서 코어(20)로 데이터가 전달되는 과정은 다음과 같다. 마스터 인터페이스(441)는 제 1 버퍼(431) 및 제 2 버퍼(432) 중 어느 하나의 빈 버퍼를 할당받고, 할당된 버퍼에 데이터를 기록한다. 이때, 할당된 버퍼는 선입선출 방식 또는 임의 접근 방식으로 데이터를 입출력할 수 있다. 할당된 버퍼(예를 들어, 제 1 버퍼(431))의 모든 엔트리들에 데이터가 기록되면, 마스터 인터페이스(441)는 데이터가 기록된 버퍼(예를 들어, 제 1 버퍼(431))를 반납하고, 다른 버퍼(예를 들어, 제 2 버퍼(432))를 할당받는다. 만약, 상기 동작이 수행되는 과정 중에 빈 버퍼가 존재하지 않으면, 마스터 인터페이스(441)는 외부 메모리(70)에 대한 데이터 전송 요청을 하지 않고, 빈 버퍼가 생길 때까지 대기한다.
- [0075] 프로세서 인터페이스(450)는 데이터가 기록된 버퍼를 할당받고, 할당된 버퍼의 엔트리에 기록된 데이터를 출력한다. 이때, 할당된 버퍼는 선입선출 방식 또는 임의 접근 방식으로 데이터를 입출력할 수 있다. 할당된 버퍼(예를 들어, 제 1 버퍼(431))의 모든 데이터가 출력되면, 프로세서 인터페이스(450)는 데이터가 기록된 버퍼(예를 들어, 제 1 버퍼(431))를 반납하고, 다른 버퍼(예를 들어, 제 2 버퍼(432))를 할당받는다. 만약, 상기 동작이 수행되는 과정 중에 데이터가 기록된 버퍼가 존재하지 않으면, 프로세서 인터페이스(450)는 프로세서 코어(20)에 이를 알리는 신호를 전송하고, 프로세서 코어(20)는 버퍼들(431, 432)에 데이터가 기록될 때까지 대기한다.
- [0076] 버퍼들(431, 432)이 더블 버퍼로 동작하는 경우에, 외부 메모리(70)에서 프로세서 코어(20)로 데이터가 전달되는 과정도 상술한 바와 유사하다. 따라서, 이하에서 자세한 설명은 생략한다.
- [0077] 상술한 바와 같이, 버퍼 그룹(430)에 포함된 버퍼들(431 내지 434)이 데이터를 입출력하는 방식은 선입선출 방식, 임의 접근 방식 또는 더블 버퍼가 있다. 이때, 버퍼들(431 내지 434) 각각은 채널로부터 전송된 신호에 기초하여 선입선출(FIFO) 방식으로 데이터를 입출력하거나 더블 버퍼(double buffer)로서 데이터를 입출력할 수 있다. 다시 말해, 버퍼들(431 내지 434) 각각이 데이터를 입출력하는 방식은 고정되어 있지 않고, 채널로부터 전송된 신호에 기초하여 변경될 수 있다.
- [0078] 마스터 인터페이스 그룹(440)은 제 1 작업에 기초하여 외부 메모리와 제 1 작업에 대응하는 데이터를 송수신하는 적어도 하나 이상의 마스터 인터페이스(441 내지 442)를 포함한다. 여기에서, 제 1 작업은 외부 메모리(70)에 저장된 데이터를 수신할 것을 요청하는 작업 또는 외부 메모리(70)로 데이터를 전송할 것을 요청하는 작업을 의미한다. 도 4의 마스터 인터페이스 그룹(440)에 포함된 마스터 인터페이스들(441 내지 442)은 도 2에 도시된 마스터 인터페이스들(331 내지 332)과 동일한 역할을 수행한다. 따라서, 이하에서 마스터 인터페이스들(441 내지 442)에 대한 자세한 설명은 생략한다. 또한, 마스터 인터페이스 그룹(440)은 총 2개의 마스터 인터페이스들(331 내지 332)을 포함하는 것으로 도시하였으나, 이에 한정되지 않는다.
- [0079] 전역 제어부(460)는 DMA 컨트롤러(30)에 대한 전반적인 제어를 담당한다. 도 4의 전역 제어부(460)는 도 2에 도시된 전역 제어부(350)와 동일한 역할을 수행한다. 따라서, 이하에서 전역 제어부(460)에 대한 자세한 설명은 생략한다.
- [0080] 프로세서 인터페이스(450)는 프로세서 코어(20)로부터 버퍼들(431 내지 434)에 요청된 제 2 작업을 나타내는 정보를 수신하고, 버퍼들(431 내지 434)에 저장된 데이터를 프로세서 코어(20)로 전송하거나 버퍼들(431 내지 434)에 데이터를 기록한다. 여기에서, 제 2 작업은 버퍼들(431 내지 434)에 저장된 데이터를 전송할 것을 요청하는 작업 또는 버퍼들(431 내지 434)에 데이터를 기록할 것을 요청하는 작업을 의미한다.
- [0081] 다시 말해, 프로세서 코어(20)는 프로세서 코어(20)에서 수행할 작업에 필요한 데이터를 버퍼들(431 내지 434)

에서 가져오라는 요청 또는 프로세서 코어(20)에서 수행된 작업의 결과로 생성된 데이터를 버퍼들(431 내지 434)에 기록하라는 요청을 프로세서 인터페이스(450)에 전달한다. 프로세서 인터페이스(450)는 프로세서 코어(20)로부터 전송된 요청에 기초하여, 버퍼 그룹(430)에 포함된 버퍼들(431 내지 434) 중 적어도 하나 이상에 프로세서 코어(20)의 요청을 전달한다.

- [0082] 본 발명의 일 실시예에 따른 DMA 컨트롤러(30)는 프로세서 코어(20)와 연결된 2개의 슬레이브 인터페이스들을 포함한다. 슬레이브 인터페이스(410)은 채널들(421 내지 423)과 연결되어 DMA 컨트롤러(30)에 포함된 일반적인 슬레이브 인터페이스의 역할을 수행한다. 그러나, 본 발명의 일 실시예에 따른 프로세서 인터페이스(450)는 버퍼들(431 내지 434)과 직접적으로 연결되어, 프로세서 코어(20)로부터의 버퍼에 대한 접근 요청을 받아들인다.
- [0083] DMA 컨트롤러(30)가 프로세서 코어(20)와 버퍼들(431 내지 434)을 직접적으로 연결하는 프로세서 인터페이스(450)를 구비함으로써, DMA 컨트롤러(30)에 포함된 버퍼들(431 내지 434)과 스크래치 패드 메모리(40) 간의 데이터 전송 과정을 생략할 수 있다. 따라서, 프로세서 시스템(10)과 외부 메모리(70) 간에 데이터를 송수신하는 데 걸리는 지연 시간과 전력 소모를 줄일 수 있다.
- [0084] 본 발명의 일 실시예에 따른 DMA 컨트롤러(30)는 3가지의 전송 모드를 제공한다. 다시 말해, DMA 컨트롤러(30)에 포함된 채널들(421 내지 423)은 각각 3가지의 전송 모드에 포함된 제 1 모드 내지 제 3 모드 중 적어도 하나 이상을 수행할 수 있다. 여기에서, 제 1 모드는 제 1 슬레이브 장치에 포함된 데이터를 버퍼에 저장하고, 버퍼에 저장된 데이터를 제 2 슬레이브 장치로 전송하는 모드를 의미한다. 또한, 제 2 모드는 제 1 슬레이브 장치에 포함된 데이터를 버퍼에 저장하는 모드를 의미한다. 또한, 제 3 모드는 버퍼에 저장된 데이터를 제 2 슬레이브 장치로 전송하는 모드를 의미한다. 또한, 제 1 슬레이브 장치 및 제 2 슬레이브 장치는 제 1 마스터 인터페이스(441) 및 제 2 마스터 인터페이스(442)에 각각 연결된 외부 장치를 의미한다.
- [0085] 이하, 채널들(421 내지 423) 중 제 1 채널(421)이 활성화된 것을 예로 하여 제 1 모드 내지 제 3 모드에 대하여 구체적으로 설명한다.
- [0086] 제 1 모드는 일반적인 DMA 컨트롤러(30)가 수행하는 전송 모드를 의미한다. 구체적으로, 제 1 채널(421)은 제 1 마스터 인터페이스(441)와 연결된 제 1 슬레이브 장치로부터 데이터를 읽어와서 버퍼 그룹(430)에 저장한다. 그리고, 제 1 채널(421)은 버퍼 그룹(430)에 저장된 데이터를 제 2 마스터 인터페이스(442)와 연결된 제 2 슬레이브 장치로 전송한다. 여기에서, 제 1 마스터 인터페이스(441)와 제 2 마스터 인터페이스(442)는 동일한 인터페이스일 수 있다. 다시 말해, 데이터의 종류에 따라서 제 1 마스터 인터페이스(441)와 제 2 마스터 인터페이스(442)는 동일한 마스터 인터페이스가 될 수 있다.
- [0087] 제 2 모드는 본 발명의 일 실시예에 따른 DMA 컨트롤러(30)가 수행하는 전송 모드를 의미한다. 구체적으로, 제 1 채널(421)은 제 1 마스터 인터페이스(441)와 연결된 제 1 슬레이브 장치로부터 데이터를 읽어와서 버퍼 그룹(430)에 저장한다.
- [0088] 제 3 모드도 본 발명의 일 실시예에 따른 DMA 컨트롤러(30)가 수행하는 전송 모드를 의미한다. 구체적으로, 제 1 채널(421)은 버퍼 그룹(430)에 저장된 데이터를 제 2 마스터 인터페이스(442)와 연결된 제 2 슬레이브 장치로 전송한다.
- [0089] 제 2 모드 및 제 3 모드는 제 1 모드에 포함된 데이터의 흐름에서 각각 절반씩을 나타내는 전송 모드이다. 채널들(421 내지 423) 각각은 전송 정보에 포함된 설정에 따라서 제 1 모드 내지 제 3 모드 중 어느 한 모드로 동작할 수 있다.
- [0090] 예를 들어, DMA 컨트롤러(30)가 제 2 모드로 설정되어 데이터를 송수신하는 과정은 도 5를 참조하여 후술한다.
- [0091] 도 5는 일 실시예에 따른 DMA 컨트롤러가 제 2 모드로 설정되어 데이터를 송수신하는 일 예를 나타내는 흐름도이다.
- [0092] 도 5를 참조하면, DMA 컨트롤러(30)가 제 2 모드로 설정되어 데이터를 송수신하는 방법은 도 4에 도시된 DMA 컨트롤러(30)에서 시계열적으로 처리되는 단계들로 구성된다. 따라서, 이하에서 생략된 내용이라 하더라도 도 1 및 도 4에 도시된 디지털 시스템(1) 또는 DMA 컨트롤러(30)에 관하여 이상에서 기술된 내용은 도 5의 데이터를 송수신하는 방법에도 적용됨을 알 수 있다.
- [0093] 이하에서는, DMA 컨트롤러(30)에 포함된 채널들(421 내지 423) 중에서 제 1 채널(421)을 통하여 데이터가 송수신되는 것으로 가정하여 설명한다.

- [0094] S510에서, 프로세서 코어(20)는 슬레이브 인터페이스(410)를 통하여 제 1 채널(421)에 전송 정보를 전달하고, 제 1 채널(421)을 활성화(enable)한다. 이때, 프로세서 코어(20)는 제 1 채널(421)의 전송 모드를 제 2 모드로 설정한다. 여기에서, 전송 정보에는 소스 메모리 주소, 대상 메모리 주소, 데이터의 기본 단, 총 데이터의 개수, 채널의 우선 순위, 버스 설정 등이 포함된다.
- [0095] S520에서, 제 1 채널(421)은 버퍼들(431 내지 434) 중에서 적어도 하나 이상의 버퍼를 할당받는다.
- [0096] S530에서, 제 1 채널(421)은 할당된 버퍼에 빈 엔트리가 충분히 존재하는지 판단한다. 만약, 할당된 버퍼에 빈 엔트리가 충분히 존재하는 경우에는 S540으로 진행하고, 그렇지 않은 경우에는 S531로 진행한다.
- [0097] S531에서, 제 1 채널(421)은 할당된 버퍼에 빈 엔트리가 충분히 발생될 때까지 대기한다.
- [0098] S540에서, 제 1 채널은 외부 메모리(70) 중 소스 메모리 주소가 가리키는 외부 메모리와 연결된 마스터 인터페이스(441)에 전송 정보를 전달하고, 전송 정보에 대응하는 데이터를 전송할 것을 요청한다.
- [0099] S550에서, 마스터 인터페이스(441)는 소스 메모리 주소가 가리키는 외부 메모리에서 데이터를 읽어온 후, 할당된 버퍼에 데이터를 저장한다.
- [0100] S560에서, 제 1 채널(321)은 설정된 데이터 전송 과정이 완료되었는지 판단한다. 만약, 데이터 전송이 완료되지 않은 경우에는 S540으로 진행하고, 데이터 전송이 완료된 경우에는 S570으로 진행한다.
- [0101] S570에서, 할당된 버퍼가 완전히 빈 경우 제 1 채널(421)은 할당된 버퍼를 반환한 후 비활성화하고, 그렇지 않은 경우 제 1 채널(421)은 버퍼가 빌 때까지 대기한다. 이때, 프로세서 코어(20)가 프로세서 인터페이스(450)를 통하여 수시로 할당된 버퍼에서 데이터를 가져간다.
- [0102] 다시 도 4를 참조하면, 본 발명의 일 실시예에 따른 DMA 컨트롤러(30)는 프로세서 인터페이스(450)를 포함할 뿐만 아니라, 제 1 모드 내지 제 3 모드 중 어느 한 모드에 따라 데이터를 송수신할 수 있다. 따라서, 도 1을 참조하여 상술한 바와 같이 일반적인 디지털 시스템(1)의 데이터를 송수신 흐름과 비교하면, 본 발명의 일 실시예에 따른 디지털 시스템(1)은 보다 적은 단계를 거쳐서 데이터를 송수신할 수 있음을 알 수 있다.
- [0103] 구체적으로, 본 발명의 일 실시예에 따른 디지털 시스템(1)에서 데이터가 전달되고 처리되는 흐름은 아래와 같다.
- [0104] 먼저, DMA 컨트롤러(30)는 외부 메모리(70)에서 DMA 컨트롤러(40) 내부에 포함된 버퍼로 데이터를 전송한다. 여기에서, 데이터는 프로세서 코어(20)가 처리하는 작업과 관련된 데이터를 의미하고, 외부 메모리(70)는 데이터가 저장되어 있는 저장 장치를 의미한다.
- [0105] 그 후에, 프로세서 코어(20)는 메모리 읽기 명령을 사용하여 버퍼에서 프로세서 코어(20) 내부에 포함된 레지스터 파일로 데이터를 전송한다. 이 단계는 DMA 컨트롤러(30)에 포함된 프로세서 인터페이스(450)를 통하여 수행된다.
- [0106] 그 후에, 프로세서 코어(20)는 레지스터 파일에 기록된 데이터를 이용하여 작업을 수행하고, 그 결과를 다시 레지스터 파일에 기록한다.
- [0107] 그 후에, 프로세서 코어(20)는 메모리 쓰기 명령을 사용하여 레지스터 파일에서 DMA 컨트롤러(40) 내부에 포함된 버퍼로 데이터를 전송한다. 이 단계는 DMA 컨트롤러(30)에 포함된 프로세서 인터페이스(450)를 통하여 수행된다.
- [0108] 그 후에, DMA 컨트롤러(40)는 버퍼에 저장된 데이터를 외부 메모리(70)로 전송한다.
- [0109] 따라서, 본 발명의 일 실시예에 따른 디지털 시스템(1)은 DMA 컨트롤러(40)와 스크래치 패드 메모리(40) 사이의 데이터 전송 단계를 생략할 수 있으며, 이에 따라 데이터를 전송하는데 드는 지연 시간 및 전력 소모를 줄일 수 있다. 다시 말해, 프로세서 코어(20)는 제 2 슬레이브 인터페이스(410)를 통하여 버퍼들(431 내지 434)에 직접 접근할 수 있는바, 데이터를 전송하는데 드는 지연 시간 및 전력 소모를 줄일 수 있다. 이하, 프로세서 코어(20)가 버퍼들(431 내지 434)에 접근하는 3가지 예들을 설명한다.
- [0110] 일 예로서, 프로세서 코어(20)는 일반적인 메모리 '읽기/쓰기' 명령을 사용하여 버퍼들(431 내지 434)에 접근할 수 있다. 구체적으로, 프로세서 인터페이스(450)는 프로세서 코어(20)의 데이터 메모리 시스템과 연결되며 고유 메모리 주소 영역을 갖는다. 따라서, 프로세서 코어(20)는 상기 주소 영역에 대한 메모리 '읽기' 명령을 사용함으로써 버퍼들(431 내지 434)에 기록된 데이터를 프로세서 코어(20) 내의 레지스터 파일에 저장할 수 있다.

또는, 프로세서 코어(20)는 상기 주소 영역에 대한 메모리 '쓰기' 명령을 사용함으로써 프로세서 코어(20) 내의 레지스터 파일에 저장된 데이터를 버퍼들(431 내지 434)에 기록할 수 있다.

[0111] 다른 예로서, 프로세서 코어(20)의 명령들이 버퍼들(431 내지 434)을 일반 레지스터 파일처럼 사용함으로써, 프로세서 코어(20)가 버퍼들(431 내지 434)에 접근할 수 있다. 구체적으로, 프로세서 인터페이스(450)는 독자적인 연결을 통하여 프로세서 코어(20)와 연결되며, 프로세서 인터페이스(450)를 통하여 프로세서 코어(20)는 버퍼들(431 내지 434)에 접근할 수 있는 명령을 제공한다. 이때, 프로세서 코어(20)가 전송한 명령들이 피연산자(operand)로서 버퍼들(431 내지 434) 중 적어도 하나 이상을 지정할 수 있다. 이 방식은 프로세서 코어(20)가 수행하는 연산 과정에서 버퍼에 바로 접근할 수 있으므로, 데이터가 프로세서 코어(20)의 레지스터 파일을 거칠 필요가 없다는 장점이 있다.

[0112] 또 다른 예로서, 프로세서 코어(20)는 특수 명령어를 사용하여 버퍼들(431 내지 434)에 접근할 수 있다. 상술한 바와 같이, 프로세서 인터페이스(450)는 독자적인 연결을 통하여 프로세서 코어(20)와 연결되며, 프로세서 인터페이스(450)를 통하여 프로세서 코어(20)는 버퍼들(431 내지 434)에 접근할 수 있는 명령을 제공한다. 이때, 프로세서 코어(20)는 특수 명령어를 생성함으로써 버퍼들(431 내지 434)과 프로세서 코어(20)의 레지스터 파일 간에 데이터를 전달하거나, 버퍼들(431 내지 434)을 제어할 수 있다. 여기에서, 프로세서 코어(20)가 생성하는 특수 명령어의 일 예는 도 8을 참조하여 후술한다.

[0113] 도 8은 일 실시예에 따른 프로세서 코어에서 DMA 컨트롤러에 포함된 버퍼에 접근하기 위한 명령어 포맷의 일 예를 도시한 도면이다.

[0114] 도 8을 참조하면, 프로세서 코어(20)가 생성하는 특수 명령어 인코딩의 일 예를 도시되어 있다. 'LOAD_BUF'는 버퍼에서 값을 읽어 프로세서 코어(20)의 레지스터 파일에 저장하는 명령을 의미하고, 'STORE_BUF'는 프로세서 코어(20)의 레지스터 파일에서 값을 읽어 버퍼에 저장하는 명령을 의미한다. 'CH'는 DMA 컨트롤러(30)의 채널들(421 내지 423) 중 적어도 어느 하나를 지정하는 필드를 의미한다. 이는, 프로세서 코어(20)는 DMA 컨트롤러(30)가 어떤 버퍼를 할당해서 사용할 것인지를 알 수 없으므로, 버퍼 번호 대신에 채널 번호를 지정하는 것이다. 'DST_REG'와 'SRC_REG'는 각각 명령의 결과를 저장할 레지스터 파일의 엔트리와 명령의 입력으로 사용할 레지스터 파일의 엔트리를 지정하는 필드를 의미한다.

[0115] 한편, 버퍼의 엔트리를 선입선출이 아닌 임의 접근이 가능하도록 설정할 경우, 프로세서 인터페이스(450)를 통하여 프로세서 코어(20)가 접근할 엔트리를 지정해주어야 한다. 'ADDR' 필드는 이러한 목적으로 사용된다.

[0116] 'CTRL_BUF'는 버퍼를 제어하거나 점검하는 명령으로서, 'CMD' 필드에 의하여 구체적인 명령의 종류가 결정된다. 여기에서, 구체적인 명령은 버퍼 할당 받기, 버퍼 반납하기, 어드레싱 모드 설정하기, 버퍼가 찼는지 확인하기, 버퍼가 비었는지 확인하기 등을 포함한다.

[0117] 도 6은 일 실시예에 따른 DMA 컨트롤러의 다른 예를 도시한 구성도이다.

[0118] 도 6을 참조하면, DMA 컨트롤러(30)는 슬레이브 인터페이스(610), 적어도 하나 이상의 채널(621 내지 623), 적어도 하나 이상의 마스터 인터페이스(631 내지 632), 내부 버스(640), 전역 제어부(650) 및 제 2 슬레이브 인터페이스(660)를 포함한다. 여기에서, 하나의 채널(621)에는 채널 제어부(6211) 및 버퍼(6212)가 포함된다.

[0119] 도 6에 도시된 DMA 컨트롤러(30)의 슬레이브 인터페이스(610), 적어도 하나 이상의 채널(621 내지 623), 적어도 하나 이상의 마스터 인터페이스(631 내지 632), 내부 버스(640) 및 전역 제어부(650)의 기능은 도 2 내지 도 3을 참조하여 설명한 바와 동일하다. 따라서, 이하에서는 자세한 설명을 생략한다.

[0120] 도 6에 도시된 DMA 컨트롤러(30)와 도 2에 도시된 DMA 컨트롤러(30)를 비교하면, 도 6에 도시된 DMA 컨트롤러(30)는 제 2 슬레이브 인터페이스(660)를 더 포함한다. 또한, 도 6에 도시된 DMA 컨트롤러(30)와 도 4에 도시된 DMA 컨트롤러(30)를 비교하면, 각각의 채널마다 버퍼가 연결되어 있는지 여부의 차이만 존재한다.

[0121] 다시 말해, 도 6에 도시된 DMA 컨트롤러(30)는 제 2 슬레이브 인터페이스(660)를 포함하므로, 프로세서 코어(20)가 DMA 컨트롤러(30)에 포함된 버퍼들(621 내지 623)에 직접 접근할 수 있다. 따라서, 도 4 내지 도 5를 참조하여 상술한 DMA 컨트롤러(30)에 관한 내용이 도 6에 도시된 DMA 컨트롤러(30)에도 모두 적용될 수 있다.

[0122] 도 7은 일 실시예에 따른 DMA 컨트롤러의 또 다른 예를 도시한 구성도이다.

[0123] 도 7을 참조하면, DMA 컨트롤러(30)는 슬레이브 인터페이스(710), 채널 그룹(720), 버퍼 그룹(730), 마스터 인터페이스 그룹(740), 전역 제어부(750), 제 1 내부 버스(761) 및 제 2 내부 버스(762)를 포함한다. 여기에서,

채널 그룹(720)는 적어도 하나 이상의 채널들(721 내지 723)을 포함하고, 버퍼 그룹(730)은 적어도 하나 이상의 버퍼들(731 내지 734)을 포함한다. 또한, 마스터 인터페이스 그룹(740)은 적어도 하나 이상의 마스터 인터페이스들(741 내지 742)을 포함한다.

[0124] 도 7에 도시된 DMA 컨트롤러(30)의 슬레이브 인터페이스(710), 채널 그룹(720), 버퍼 그룹(730), 마스터 인터페이스 그룹(740), 전역 제어부(750), 제 1 내부 버스(761) 및 제 2 내부 버스(762)의 기능은 도 4 내지 도 5를 참조하여 설명한 바와 동일하다. 따라서, 이하에서는 자세한 설명을 생략한다.

[0125] 도 7에 도시된 DMA 컨트롤러(30)와 도 4에 도시된 DMA 컨트롤러(30)를 비교하면, 도 7에 도시된 DMA 컨트롤러(30)의 슬레이브 인터페이스(710)는 도 4에 도시된 DMA 컨트롤러(30)의 슬레이브 인터페이스(410) 및 프로세서 인터페이스(450)의 역할을 모두 수행한다.

[0126] 다시 말해, 슬레이브 인터페이스(710)는 슬레이브 인터페이스(410) 및 프로세서 인터페이스(450)로 구분되지 않고, 물리적으로 동일한 인터페이스를 의미한다. 따라서, 도 4 내지 도 5를 참조하여 상술한 DMA 컨트롤러(30)에 관한 내용이 도 7에 도시된 DMA 컨트롤러(30)에도 모두 적용될 수 있다.

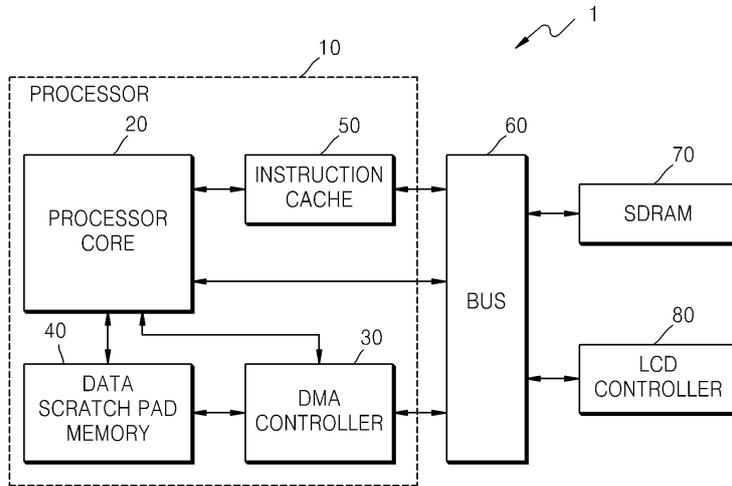
[0127] 본 실시예와 관련된 기술 분야에서 통상의 지식을 가진 자는 상기된 기재의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 방법들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

부호의 설명

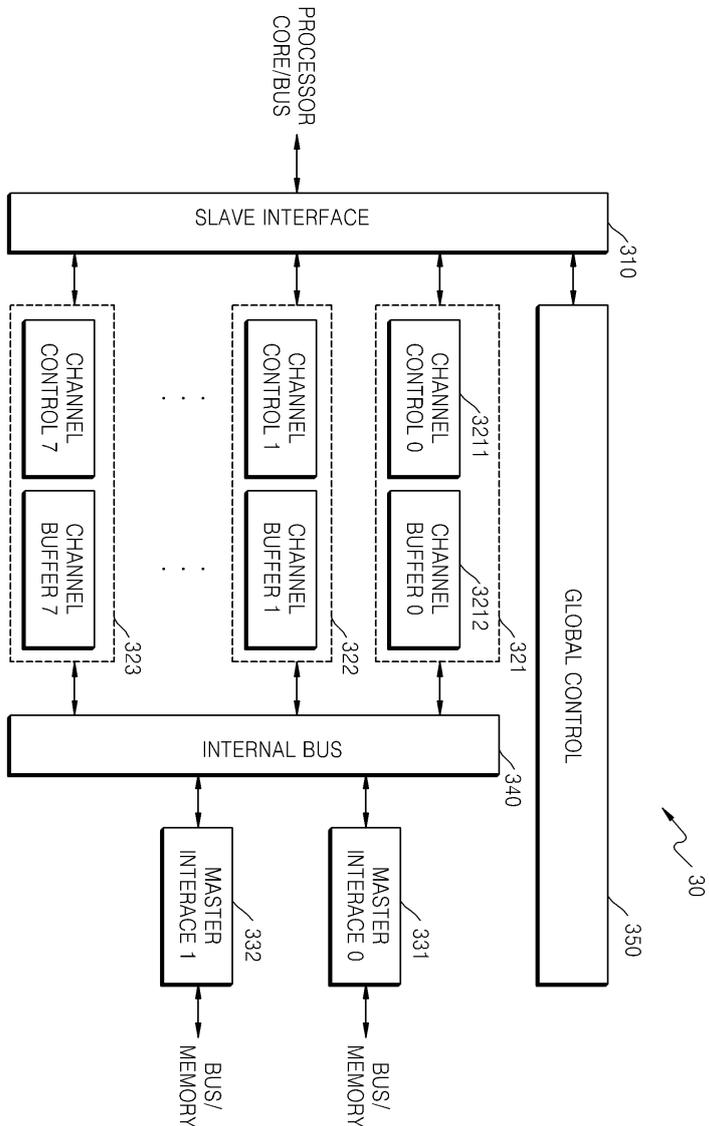
- [0128] 30: DMA 컨트롤러
- 410: 슬레이브 인터페이스
- 420: 채널 그룹
- 421, 422, 423: 채널
- 430: 버퍼 그룹
- 431, 432, 433, 434: 버퍼
- 440: 마스터 인터페이스 그룹
- 441, 442: 마스터 인터페이스
- 450: 프로세서 인터페이스
- 460: 전역 제어부
- 471, 472: 내부 버스

도면

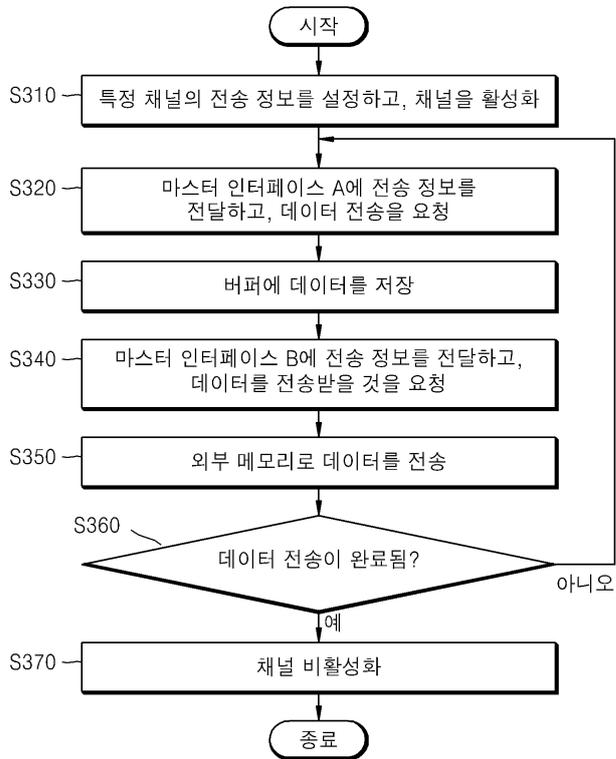
도면1



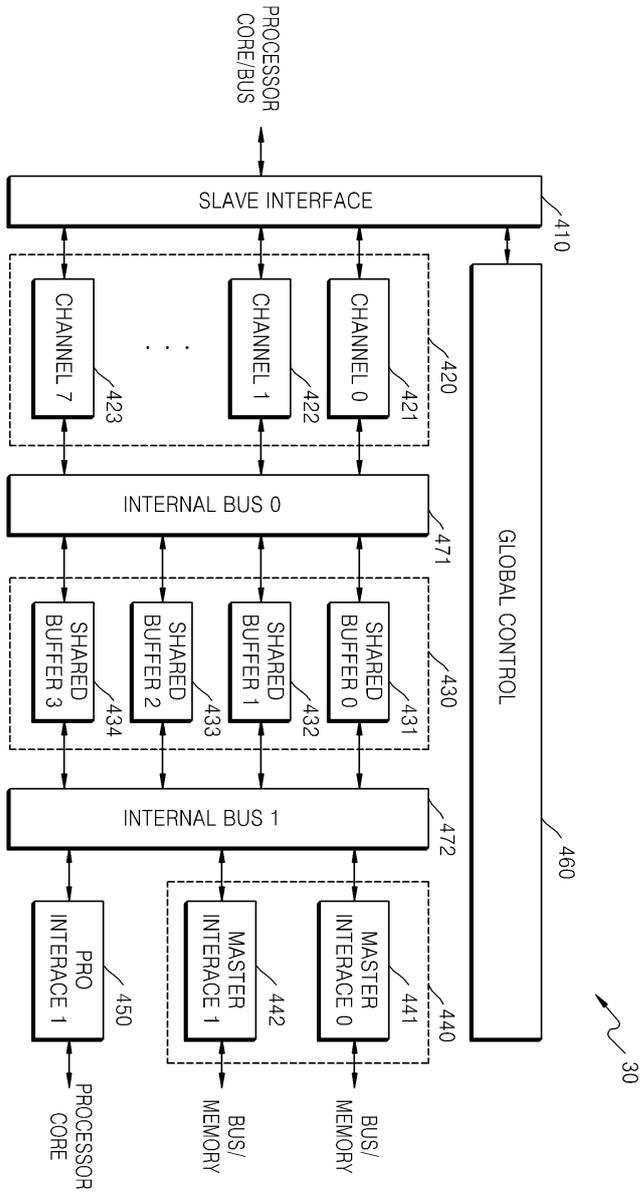
도면2



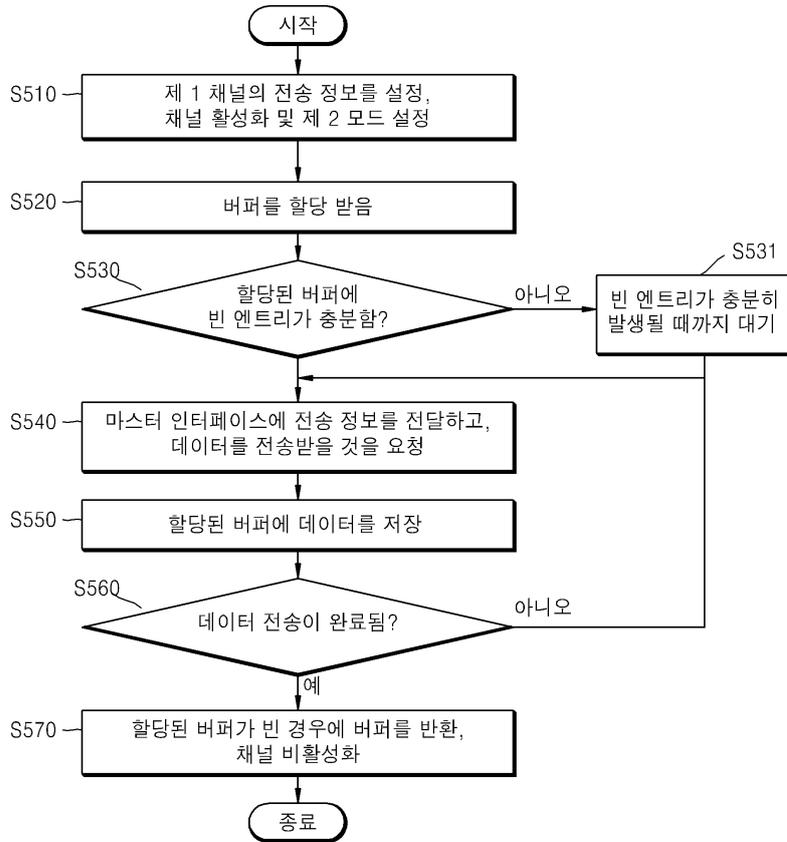
도면3



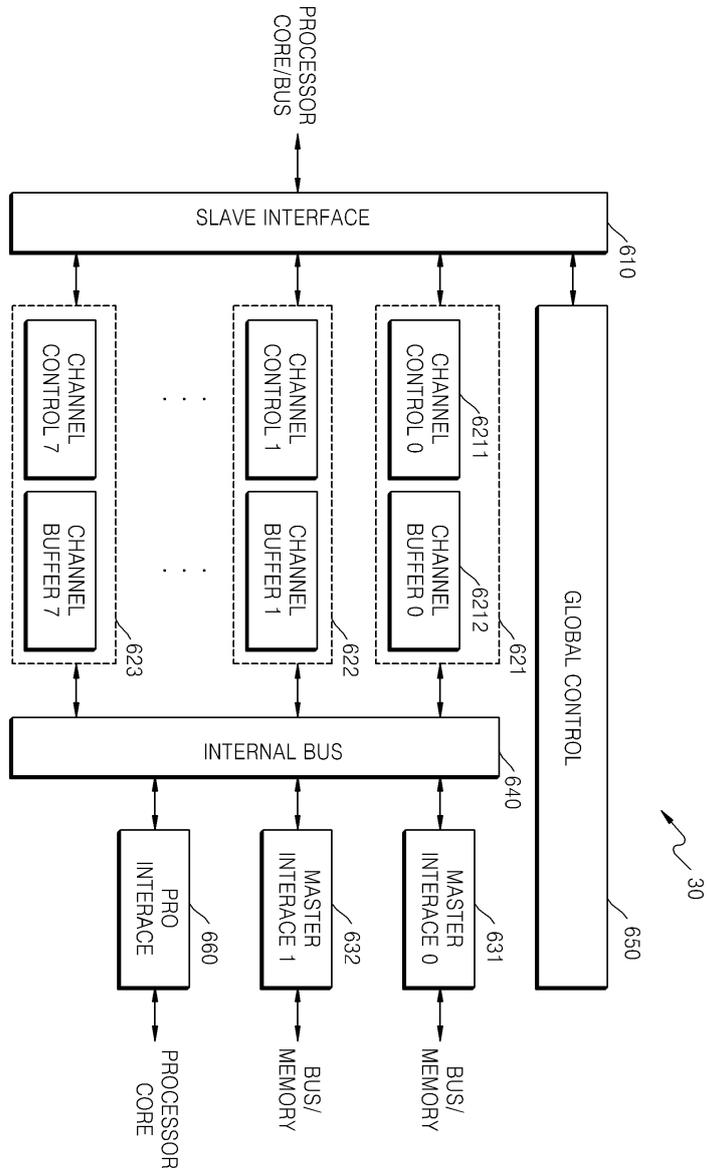
도면4



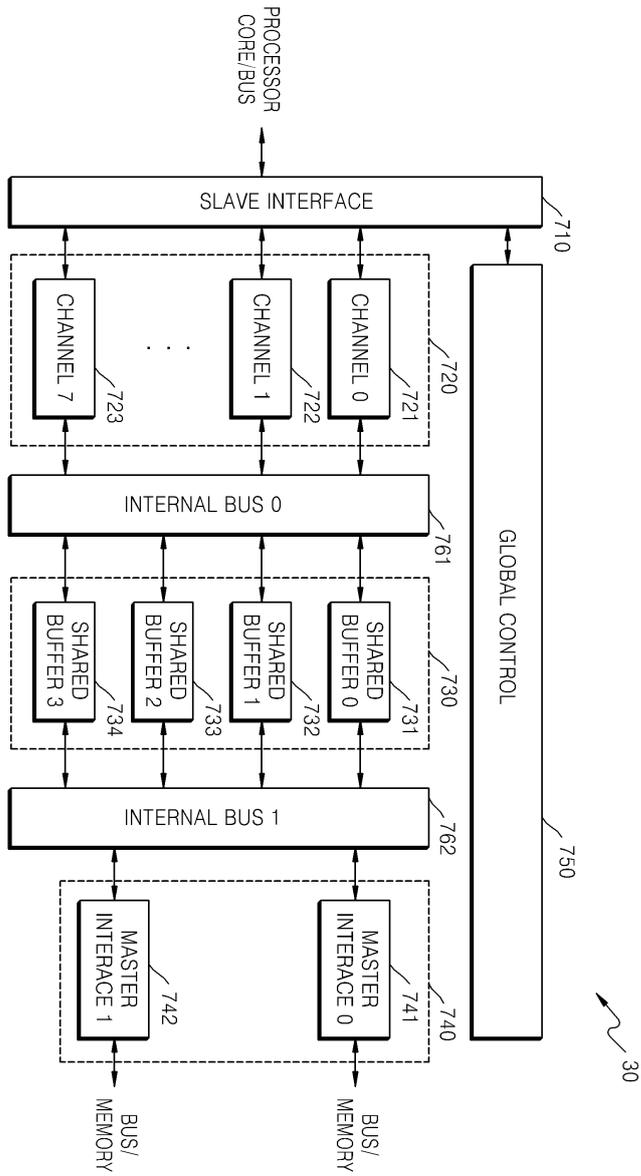
도면5



도면6



도면7



도면8

