(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.[7]
G06F 13/28

(11) 공개번호 10-2004-0040458
(43) 공개일자 2004년05월12일

(21) 출원번호 10-2004-7003435
(22) 출원일자 2004년03월08일
번역문 제출일자 2004년03월08일
(86) 국제출원번호 PCT/US2002/027642
(87) 국제공개번호 WO 2003/023625
(86) 국제출원출원일자 2002년08월30일
(87) 국제공개일자 2003년03월20일

(30) 우선권주장 09/949,461 2001년09월07일 미국(US)

(71) 출원인 인텔 코오퍼레이션
미합중국 캘리포니아 산타클라라 미션 칼리지 블러바드 2200

(72) 발명자 가나패씨,구마
미국94043캘리포니아주마운틴뷰플라워레인365

가나패씨필라이,루반
미국94568캘리포니아주더블린암버글렌스트리트5602

샤,샤린
미국95086캘리포니아주선니밸암스트롱플레이스2455

모우싸,조지
미국94086캘리포니아주서니밸알핀테라스984-2

필호워얼,에프,3세
미국94587캘리포니아주유니온시티데코따로드2175

샤,루치어
미국94040캘리포니아주마운틴뷰넘버180이스트엘카미노리얼870

(74) 대리인 주성민
백만기
이중희

심사청구 : 있음

(54) 시스템 온 칩에 대한 분산형 직접 메모리 액세스 방법 및장치

요약

분산형 직접 메모리 액세스(DMA; Direct Memory Access) 방법, 장치 및 시스템이 시스템 온 칩(SOC) 내에서 제공된다. DMA 제어기 유닛은 글로벌 버퍼 메모리로 직접 메모리 액세스를 원하는 기능 모듈에 분산된다. DMA 제어기 유닛은 어느 기능 블록이 글로벌 버퍼 메모리에 대한 액세스를 가질지를 중재한다. 기능 모듈은 직접 메모리 액세스가 일어나는 시스템 버스에 대하여 인터페이스(interface)한다. 직접 메모리 액세스의 대상이 되는 글로벌 버퍼 메모리는 시스템 버스에 연결된다. 버스 중재기(bus arbitrator)는 어느 기능 모듈이 직접 메모리 액세스를 수행하기 위한 시스템 버스에 대한 액세스를 가질지를 중재하는 데 이용된다. 일단 버스 중재기에 의해 시스템 버스에 대한 액세스를 가질 기능 모듈이 선택되면, 그 기능 모듈은 글로벌 버퍼 메모리에 대한 DMA 루틴(routine)을 설정할 수 있다.

대표도

도 2

명세서

기술분야

본 발명은 일반적으로 직접 메모리 액세스(direct memory access) 제어기에 관한 것이다. 더욱 구체적으로, 본 발명은 시스템 온 칩(system on chip)에 대한 직접 메모리 액세스에 관한 것이다.

배경기술

컴퓨터 시스템 내에서, 시스템 버스(system bus)를 통해 주 메모리 또는 중앙 처리 유닛(CPU; Central Processing Unit)의 레지스터(register)와 입/출력(I/O) 장치들 사이에 데이터를 전송하는 방법에는 일반적으로 세 가지가 있다. 이들 세 가지 방법은 프로그램에 의한 I/O(programmed I/O), 인터럽트 구동 I/O(interrupt driven I/O) 및 직접 메모리 액세스(DMA; Direct Memory Access)이다. 프로그램에 의한 I/O 및 인터럽트 구동 I/O의 경우, CPU는 시스템 버스를 통해 I/O 장치로 데이터를 전송하기 위해 주 메모리로부터 데이터를 획득하고, 주 메모리로 데이터를 전송하기 위해 I/O 장치로부터 데이터를 획득하는 역할을 한다. 따라서 프로그램에 의한 I/O 및 인터럽트 구동 I/O는 CPU가 직접 관련될 것을 필요로 하며, 이는 CPU가 데이터 전송 관리에 구속되어 다른 더욱 중요한 기능들을 수행할 수 없도록 만든다.

이에 직접 메모리 액세스(DMA)가 도입되었으며, 이는 데이터 전송에 있어 CPU의 관여 정도를 경감시키기 위한 집중형 DMA 제어기(centralized DMA controller)를 필요로 한다. 집중형 DMA 제어는 잘 알려져 있다. 집중형 DMA 제어기는 본질적으로 시스템 버스를 통해 주 메모리와 I/O 장치간에 데이터를 전송하는 역할을 담당한다. DMA 액세스에 있어서, CPU의 관여 정도는 DMA 전송을 처음으로 셋업(set up)하고 DMA 전송의 완료시에 이를 정리하는 정도로 줄어들게 된다. 실제 DMA 데이터 전송이 DMA 제어기에 의해 제어되는 동안, CPU는 다른 계산 또는 기능을 수행할 수 있다. 결과적으로, DMA 데이터 전송은 프로그램에 의한 I/O 또는 인터럽트 구동 I/O 방법보다 더욱 효율적이다. 시스템 버스 상에서의 충돌을 방지하기 위해, DMA 제어는 집중형 DMA 제어기로 집중화되었다. 집중형 DMA 제어기는 부착된 I/O 장치들 중 한 번에 하나에 대해서만 그 I/O 장치와 주 메모리 사이에서의 데이터 전송을 허용함으로써, 부착된 I/O 장치들에 대하여 시스템 버스의 중재(arbitration)를 수행한다. 또한 복수의 I/O 장치가 DMA 액세스를 동시에 수행하고자 하는 경우, 집중형 DMA 제어기는 소정의 우선 순위 부여 방법을 수행한다. 결과적으로, 집중형 DMA 제어기의 제어 로직(control logic)은 다소 복잡할 수 있으며, 이로 인해 집중형 DMA 제어기의 회로에 있어서 블록(block) 크기가 다소 커질 수 있다.

개선된 반도체 제조 기술로 인해 집적 회로의 예컨대 트랜지스터 채널 길이와 같은 피처(feature) 크기가 줄어들게 되었다. 작아진 트랜지스터로 인해 더욱 복잡한 집적 회로 칩을 제조할 수 있게 됨에 따라, 기능 시스템(functional system)이 단일 집적 회로 칩 내에 형성될 수 있고, 만족스러운 수율(yield)을 낳는다. 실질적으로 완성된 시스템인 이러한 기능 시스템은 종종 시스템 온 칩(SOC; System On Chip)으로 일컬어진다. SOC에는 메모리로의 또는 메모리로부터의 데이터 전송을 필요로 하는 기능 블록이 다수 존재할 수 있다. 또한 SOC를 제조하는 경우에 이를 유연성 있게 만들어, 더욱 많은 데이터의 취급 및/또는 다른 유형의 기능 블록의 추가를 위해 기능 블록의 수가 확장될 수 있게끔 하는 것이 종종 바람직하다. 기능 블록의 수가 증가함에 따라, SOC에서의 잠재적인 중재 문제가 증가한다. 이러한 경우 집중형 DMA 제어기를 사용하는 것의 단점은, 확장을 위해 제어 로직이 다시 설계될 필요가 있을 수 있다는 점이다.

발명의 상세한 설명

도면에서 동일한 참조 번호 및 명칭은 마찬가지의 기능을 제공하는 동일한 요소이다. 참조 지정 번호 뒤의 문자는 그 참조 지정 번호를 갖는 요소의 예를 나타낸다.

본 발명의 이하의 상세한 설명에서는, 본 발명의 완전한 이해를 제공하기 위해 다수의 특정한 세부 사항에 관하여 설명하고 있다. 그러나 본 발명이 이러한 특정한 세부 사항 없이도 실시될 수 있음은 본 기술 분야의 당업자에게 있어 자명하다. 다른 경우에 있어서, 본 발명의 태양을 불필요하게 불명료하게 하지 않도록 하기 위해 잘 알려진 방법, 절차(procedure), 콤포넌트(component) 및 회로에 대해서는 상세히 기술하지 않았다.

분산형 직접 메모리 액세스(DMA; Direct Memory Access) 방법, 장치 및 시스 템이 시스템 온 칩(SOC) 내에서 제공된다. DMA 제어기 유닛은 글로벌 버퍼 메모리(global buffer memory)에 대한 직접 메모리 액세스를 원하는 기능 모듈들에게 분산된다. DMA 제어기 유닛은 어느 기능 블록이 글로벌 버퍼 메모리에 대한 액세스를 가질지를 중재한다. 기능 모듈은 직접 메모리 액세스가 일어나는 시스템 버스에 대하여 인터페이스(interface)한다. 직접 메모리 액세스의 대상이 되는 글로벌 버퍼 메모리는 시스템 버스에 연결된다. 버스 중재기(bus arbitrator)는 직접 메모리 액세스를 수행하기 위하여, 시스템 버스에 대한 액세스를 어느 기능 모듈이 가질지를 중재하는 데 이용된다. 일단 버스 중재기에 의해 시스템 버스에 대한 액세스를 가질 기능 모듈이 선택되면, 그 기능 모듈은 글로벌 버퍼 메모리에 대한 DMA 루틴(routine)을 설정할 수 있다.

도면의 간단한 설명

도 1은 본 발명의 시스템 온 칩의 블록도.

도 2는 본 발명의 온칩 VOP(Voice Over Packet) 시스템의 상세 블록도.

도 3은 도 2에 도시된 코어(core) DMA 제어기를 포함하는 코어 프로세서의 예를 나타낸 블록도.

도 4는 시작 및 끝 DMA 어드레스 위치를 갖는 DMA 데이터를 포함하는 글로벌 버퍼 메모리의 예시 메모리 맵 및 글로벌 버퍼 메모리에 저장된 DMA 데이터를 기술하는 DMA 기술자(descriptor) 테이블.

도 5는 도 2의 직렬 DMA 제어기의 블록도.

도 6a는 도 2의 멀티채널(multichannel) 직렬 포트를 통해 송신 또는 수신되 는 데이터 프레임의 직렬 데이터 스트림(stream)의 예를 나타낸 도면.

도 6b는 도 5의 TDM 리매퍼(remapper) 메모리에 의해 수행되는 멀티채널 직렬 포트에 대한 데이터 프레임의 리매핑(remapping)의 예를 나타낸 도면.

도 7은 도 2의 코어 DMA 제어기의 블록도.

도 8은 도 2의 코어 DMA 제어기의 블록도.

실시예

이제 도 1을 참조하면, 시스템 온 칩(SOC)의 블록도가 도시되어 있다. 일 실시예의 시스템 온 칩(SOC; 100)은 VOP(Voice Over Packet) 시스템으로서, 전화 네트워크(telephone network)에 대한 인터페이스를 위한 멀티채널 전이중 직렬 포트(multichannel full duplex serial port) 및 패킷 네트워크(packet network)에 대한 다른 접속을 통한 인터페이스를 위한 다른 포트를 포함한다. 일 실시예에서 전화 네트워크는 PSTN 전화 네트워크이며, TDM 직렬 데이터 스트림을 갖는다.

이제 도 2를 참조하면, VOP 시스템 온 칩(VOP SOC; 100)의 상세한 블록도가 도시되어 있다. VOP 시스템 온 칩(100)은 글로벌 버퍼 메모리(210)에 연결된 중앙 시스템 버스(200)를 갖는다. 시스템 버스(200)는 표준화된 버스 시스템으로서, 어드레스 신호, 데이터 신호 및 제어 신호를 위한 전도성 트레이스(trace) 또는 와이어(wire)를 포함한다. 버스는 어드레스 라인(address line)들을 이용하여 DMA 전송을 수행한다. 버스는 다수의 DMA 트랜잭션(transaction)을 동시에 멀티플렉싱(multiplexing)하며, 한 번에 하나의 트랜잭션만을 처리한다. 따라서 버스는 이러한 전송(판독 또는 기록)을 위해 일단의 어드레스 라인들을 필요로 한 다. 어드레스 라인 및 데이터 라인은 별개이다. 데이터 신호에 대한 시스템 버스(200)의 버스 폭은 일 실시예에서 64 비트의 데이터 시스템 버스 폭(DSBW; Data System Bus Width)을 갖는다. 64 비트의 데이터 시스템 폭은 고대역(high bandwidth) 데이터 전송을 용이하게 한다. 시스템 온 칩(100)의 기능 모듈에 의한 중앙 시스템 버스(200)에 대한 액세스는, 시스템 버스(200)에 연결된 버스 중재기들(201A 내지 201N)에 의해, 일정한 프로그램 가능한 중재 프로토콜(programmable arbitration protocol)을 이용하여 중재된다. 버스 중재기들(201A 내지 201N)에 의해 제공되는 일정한 중재 프로토콜 및 표준화된 시스템 버스(200)는 시스템 버스(200)의 중재를 간단하게 해 준다. 글로벌 버퍼 메모리(210)에 대한 액세스를 취득하기 위해서, 시스템 온 칩(100)의 기능 모듈은 우선 버스 중재기들(201A 내지 201N)을 통하여 시스템 버스(200)에 대한 액세스를 취득해야 한다. 시스템 버스(200)에 대한 순환형(round-robin) 중재 방식은, 각각의 분산형 DMA 마스터 제어기(203A 내지 203N, 207 및 215)가 시스템 버스(200)에 자주 액세스하고, 그 시점에 글로벌 메모리(210)를 액세스할 수 있도록 보장해 준다.

VOP SOC(100)에 대한 모든 코드, 데이터 및 상태 정보는 중앙 저장소(central depository)로서 작용하는 글로벌 버퍼 메모리(210) 내에 저장되고 그로부터 판독된다. 따라서 데이터 대역폭이 중요하며, 본 발명의 분산형 DMA 시스템을 이용하여 시스템 버스(200)를 통해 글로벌 버퍼 메모리(210) 안팎으로 정보를 효율적으로 전송하는 것이 바람직하다. 버스 중재기는 시스템 버스에 대한 액세스를 중재하지만, 분산형 DMA 시스템은 글로벌 버퍼 메모리(210)에 대한 액세스 를 중재한다. 글로벌 메모리(210)는 시스템 온 칩(100)의 분산형 DMA 마스터 제어기에 대하여 슬레이브(slave)이며, 슬레이브 DMA 인터페이스를 포함한다. 글로벌 버퍼 메모리(210)는 분산형 DMA 제어기가 개시한, 시스템 버스(200)를 통한 DMA 전송 요청에 대하여 응답한다. 분산형 DMA 시스템은 기능 모듈이 글로벌 버퍼 메모리에 대한 그들 자신의 어드레스를 생성할 수 있도록 해 주며, 큰 데이터 블록들을 액세스하기 위한 명령어가 매 주기마다 지속적으로 전달되는 것을 막는다. DMA에 의해 어드레스가 자동으로 생성되고 복수의 데이터 블록들 또는 바이트들이 차례로 액세스되는 동안에 레지스터가 셋업된다. 글로벌 버퍼 메모리(210)에 대한 직접 메모리 액세스를 원하는 시스템 온 칩(100)의 기능 모듈은, 하나 이상의 신호 프로세서(202A 내지 202N), 전화 네트워크에 대한 인터페이스를 위한 멀티채널 전이중 직렬 포트(206) 및 호스트 및 패킷 네트워크에 대한 인터페이스를 위한 호스트 포트(214)를 포함한다. SOC(100)의 글로벌 버퍼 메모리(210)는 중앙 시스템 버스(200)에 직접 연결된다.

글로벌 버퍼 메모리(210)에 대한 직접 메모리 액세스를 원하는 시스템 온 칩(100)의 모듈은, 마이크로 제어기(micro controller; 223)가 없는 경우 자신만의 DMA 제어기를 갖는다. 마이크로 제어기(223)는 버스 인터페이스(224) 및 버스 중재기(201C)를 통해 시스템 버스(200)에 연결된다. 마이크로 제어기(223)는 작업(task)의 스케줄링(scheduling)과 직렬 포트(206) 및 호스트 포트(214)로부터의 데이터 흐름(flow)을 조정하는 시스템 또는 실행 관리자(executive manager)로서 작용한다. 이는 DMA 제어기를 필요로 하지 않으나, 글로벌 메모리(210)로 라우 팅(routing)되는 명령어 및 데이터에 대한 캐시 라인 미스(cache line miss)들을 생성한다. 마이크로 제어기는 캐시 라인 미스의 생성을 야기하는 데이터 캐시(227) 및 이와 관련된 명령어 캐시(228)를 갖는다. 마이크로 제어기는 DMA 기술자(descriptor)를 셋업하고, SOC(100)의 개개의 기능 유닛에게 DMA 전송을 수행하도록 명령한다. 버스 중재기(201C) 및 버스 인터페이스(224)는 서로 통신함으로써 마이크로 제어기가 시스템 버스에 대한 액세스를 취득하도록 한다. 마이크로 제어기(223)는 작업 제어를 제공하고, 데이터의 교통 경찰과 같이 작용하여 글로벌 버퍼 메모리(210) 안팎으로의 데이터 전송을 조정한다. 마이크로 제어기(223)는 DMA 데이터 전송에 관하여 글로벌 버퍼 메모리 내에 DMA 기술자 리스트를 기록함으로써 DMA를 셋업한다. 이는 그 후 기능 유닛과 통신하여, DMA 기술자 리스트의 시작 어드레스를 가리킴으로써 데이터의 DMA를 수행한다. 이어서 기능 유닛은 DMA 기술자 리스트를 판독하여 데이터의 DMA를 수행한다. DMA 기술자 리스트는 데이터에 대한 시작 어드레스 및 전송될 바이트의 수를 포함한다. DMA 기술자 리스트는 이하 상세히 설명할 다른 정보를 포함한다. 마이크로 제어기(223) 외에도, 하나 이상의 코어 프로세서(202A 내지 202N)는 또한 글로벌 메모리(210)내에 DMA 기술자를 형성할 수 있으며, 특히 코어 DMA 전송의 경우에 그러하다. 예컨대 마이크로 제어기(223)는 DMA를 하나 이상의 프로세서(202A 내지 202N)에 대하여 셋업함으로써, 소정의 통신 채널에 대한 데이터 프레임 또는 블록을 처리한다. 이는 글로벌 버퍼 메모리에 있는 기술자 리스트의 시작 어드레스를 하나의 코어 프로세서에게 통보한다. 상기 하나의 코어 프로세서는 기술자 리스트에 있는 각 라인을 통해 판독하고, 글로벌 버퍼 메모리로부터 코어 프로세서의 로컬 메모리(local memory)로의 데이터의 DMA를 수행한다.

멀티채널 직렬 포트(206)는 복수의 전이중 TDM(Time Division Multiplexed) 원격통신(telecommunication) 채널에 대하여 인터페이스함으로써, VOP SOC(100)가 복수의 통신 채널을 지원하도록 한다. 멀티채널 직렬 포트(206)는 약 512 채널의 복수의 통신 채널 각각에 대하여 수신기 및 송신기로서 작용한다. 전화 네트워크로부터 수신된 데이터는 코어 프로세서들(202A 내지 202N)에 의해 처리되도록 멀티채널 직렬 포트(206)에 의해 글로벌 버퍼 메모리(210)에 저장된다. 코어 프로세서들(202A 내지 202N)에 의해 처리된 후 전화 네트워크 상으로 전송될 데이터는, 멀티채널 직렬 포트(206)에 의해 글로벌 버퍼 메모리(210)로부터 검색(retrieve)된다. 글로벌 버퍼 메모리를 직접 액세스하기 위해, 멀티채널 직렬 포트(206)는 버스 중재기(201A)에 연결된 직렬 DMA 제어기(207)를 포함한다.

호스트 포트(214)는 호스트에 연결되어, 파워업(power up) 시에 외부 데이터를 수신하고, 자신이 처리할 통신 채널에 관한 정보를 셋업한다. 또한 호스트 포트(214)는 패킷 네트워크에 연결되어 복수의 통신 채널에 대하여 데이터의 패킷 페이로드(payload)를 전이중(full duplex) 방식으로 송수신(transceive)한다. 패킷 네트워크로부터 수신된 데이터는 코어 프로세서들(202A 내지 202N)에 의해 처리되도록 호스트 포트(214)에 의해 글로벌 버퍼 메모리(210)에 저장된다. 코어 프로세서들(202A 내지 202N)에 의해 처리된 후 패킷 네트워크 상으로 전송될 데이터는, 호스트 포트(214)에 의해 글로벌 버퍼 메모리(210)로부터 검색된다. 호스트 포트(214)에 의해 송수신되는 패킷 페이로드는 패킷 네트워크상에서의 전송에 앞서 패킷화(packetize)되며, 패킷 네트워크로부터 수신된 후에는 호스트 포트를 통해 수신된 패킷 페이로드 데이터로 역패킷화(depacketize)된다. 그 대신, 패킷 네트워크로 직접 인터페이스하기 위해 패킷화기(packetizer) 및 역패킷화기(depacketizer)가 VOP SOC(100) 내에 포함될 수 있다.

일 실시예에서, 호스트 포트(214)는 또한 Intel I960 호환 가능 호스트 포트의 프로토콜을 구현한다. 호스트 포트(214)는 외부 정보를 수신하기 위해 Intel I960 호스트 프로세서와 같은 호스트에 결합하는데, 이는 글로벌 버퍼 메모리(210)가 시스템 온 칩(100)의 내부에 있기 때문이다. 호스트는 SOC(100)의 부트업(boot up) 중에 부트업 시퀀스(boot up sequence)에 관한 정보를 호스트 포트(214)를 통해 제공할 뿐만 아니라, SOC(100)이 지원할 통신 채널에 관한 채널 셋업 정보 및 사용자가 SOC(100)에 적재(load)할 필요가 있는 그 밖의 임의의 외부 정보를 제공한다. 일 실시예의 호스트 포트(214)는 32 비트 폭의 병렬 데이터 포트를 포함한다. 즉, 패킷 페이로드 데이터 및 다른 정보는 32 비

트 데이터 워드(word) 폭을 이용하여 외부 소스(source)로부터 호스트 포트 안팎으로 흐를 수 있다. 호스트 포트(214)는 외부 버스상의 32 비트 데이터를 시스템 버스(200)상에서의 전송을 위해 64 비트 데이터로 변환시키고, 시스템 버스(200)로부터의 64 비트 데이터를 호스트 포트 밖으로의 전송을 위해 32 비트 데이터로 변환시킨다.

코어 프로세서(202A 내지 202N)는 전화 네트워크로부터 수신된 TDM 신호를 처리하며, 그에 응답하여 전화 네트워크상에서의 통신을 위한 TDM 신호를 생성하거 나, 또는 패킷 네트워크 상에서의 통신을 위한 패킷 페이로드를 생성할 수 있다. 패킷 네트워크로부터 수신된 패킷 페이로드는 코어 프로세서(202A 내지 202N)에 의해 처리되어, 패킷 네트워크 상에서의 전송을 위한 패킷 페이로드로, 또는 전화 네트워크상에서의 통신을 위한 TDM 데이터로 된다. 어느 경우라도, 처리를 필요로 하는 데이터는 우선 글로벌 버퍼 메모리(210)에 저장된다. 그 후 하나 이상의 코어 프로세서(202A 내지 202N)는 처리를 위해 소정의 채널에 대한 데이터를 검색한다. 상기 하나 이상의 코어 프로세서(202A 내지 202N)에 의한 데이터의 처리가 종료되면, 이는 글로벌 버퍼 메모리(210)에 저장된다. 글로벌 버퍼 메모리를 직접 액세스하기 위해, 상기 하나 이상의 코어 프로세서(202A 내지 202N)는 이들의 로컬 메모리 내로 데이터를 DMA하기 위한 하나 이상의 코어 DMA 제어기(203A 내지 203N)를 포함한다. 채널의 기능적 처리와 관련된 코드 및 채널과 그 데이터 샘플들의 사전 처리를 위한 상태 정보가 또한 글로벌 버퍼 메모리(210)에 저장되며, 새로운 데이터 샘플들을 처리할 상기 하나 이상의 코어 프로세서들 중 하나의 로컬 메모리(프로그램 메모리 및 데이터 메모리) 내로 DMA 전송된다.

이제 도 3을 참조하면, 코어 프로세서(202)의 블록도가 도시되어 있다. 도 3에 도시된 코어 프로세서(202)는 도 2의 하나 이상의 코어 프로세서(202A 내지 202N) 중 하나의 예를 나타낸다. 코어 프로세서(202)는 도 3에 도시된 바처럼 함께 연결된 코어 DMA 제어기(203), 디지털 신호 프로세서(DSP; 300), 데이터 메모리(302) 및 프로그램 메모리(304)를 포함한다. 데이터 메모리(302) 및 프로그램 메모리(304) 내의 메모리 위치(memory location)들은 두 개의 통신 채널을 처리 하기 위하여 이중 채널(dual channel)로 할당될 수 있다. 메모리 위치를 이러한 방식으로 두 채널로 할당하는 것은 종종 이중 버퍼화 메모리(double buffered memory) 방식이라 일컬어진다. 즉, 데이터 메모리(302)는 채널 1 데이터 메모리(302A) 및 채널 2 데이터 메모리(302B)와 같이 두 채널 상에 분리된 것으로 간주될 수 있다. 마찬가지로 프로그램 메모리(304)는 채널 1 프로그램 메모리(304A) 및 채널 2 프로그램 메모리(304B)와 같이 두 채널로 분리된 것으로 간주될 수 있다. 채널 1 프로그램 메모리(304A)는 채널 1 데이터 메모리(302A)에 대응된다. 채널 2 프로그램 메모리(304B)는 채널 2 데이터 메모리(302B)에 대응된다.

코어 DMA 제어기(203) 및 DSP(300)은 메모리 채널들 중 어느 것이 각각 액세스 중인지에 따라 전환된다. 코어 DMA 제어기(203)는 처리되고 있지 않은 채널에 대응하는 데이터 메모리 및 프로그램 메모리의 메모리 위치에 데이터 또는 코드를 기록한다. 메모리 위치의 한 채널이 DSP(300)에 의해 처리되는 동안, 메모리 위치의 다른 채널은 DMA에 의해 판독되거나 또는 업데이트(update)된다. 예컨대, DSP(300)은 프로그램 메모리(304)의 채널 1(304A)에 저장된 코드를 이용하여 데이터 메모리(302)의 채널 1(302A)에 있는 데이터를 처리한다. DSP(300)이 채널 1을 처리하는 동안, 코어 DMA 제어기(203)는 다음 채널을 처리하기 위해 프로그램 메모리(304)의 채널 2(304B)를 업데이트한다. 또한 코어 DMA 제어기(203)는 처리가 종료된 후 채널 2로부터 데이터를 판독할 수 있고, 그 후 다음 채널을 처리하기 위해 새로운 데이터로 채널 2(302B)를 업데이트할 수 있다. 코어 DMA 제어기(203)는 직 접 메모리 액세스를 이용하여 글로벌 버퍼 메모리(210)와 데이터 메모리(302) 및 프로그램 메모리(304) 사이에서 코드 및 데이터를 전송한다.

데이터 메모리(302)와 코어 DMA 제어기(203) 사이에서의 데이터 전송은 기회가 있을 때마다 판독 및 기록을 함으로써 이루어진다. 즉 매번의 기회마다, 데이터는 하나 이상의 코어 프로세서(200A 내지 200N)의 데이터 메모리(302)와 코어 DMA 제어기(203) 사이에서 전송된다. 프로세서가 이중 포트(dual-ported) 데이터 메모리(302)의 어느 한 쪽 포트를 이용하고 있지 않는 경우, 데이터는 한 쪽 포트로 기록되거나 또는 한 쪽 포트로부터 판독될 수 있다. 처리중인 뱅크에 데이터를 기록하기 위한 빈 메모리 위치(들)를 이용할 수 있을 때마다, DMA 프로세스는 빈 메모리 위치(들)에 기록을 할 수 있다. 메모리 위치가 처리를 완료한 데이터를 포함하게 된 후, DMA 프로세스는 가능한 경우 데이터 메모리(302)내의 메모리 위치(들)로부터 판독할 수 있다. 코어 프로세서들(200A 내지 200N)은 메모리 뱅크를 액세스하는 데 있어 더 높은 우선 순위를 가지며, DMA 제어기 사이클(cycle)은 이를 빼앗아 데이터 메모리에 대한 액세스를 취득하여 기회가 있을 때마다 판독 및 기록을 한다. 마찬가지로, 프로그램 메모리(304)와 글로벌 버퍼 메모리(210) 사이에서의 코드 전송은 기회가 있을 때마다 판독 및 기록을 함으로써 이루어진다.

데이터 메모리(302)와 글로벌 버퍼 메모리(210) 사이의 DMA 전송을 위한 명령어는 시작(start), 중단(stop), 계속(continue), 중지(suspend) 및 재개(resume)를 포함한다. DMA 제어기(203)에는 DMA 기술자 레지스터(DMA descriptor register)가 존재한다. 이들 명령어 중 하나 이상은 DMA가 진행중인 동안 DMA 기 술자 레지스터에 기록될 수 있으며, 적합한 동작이 수행되도록 할 수 있다. DMA 전송은 또한 중단 명령 또는 중지 명령을 이용하여 사용자에 의해 중단될 수 있다. DMA 전송이 중단되면, 이는 계속 명령을 이용하여 계속될 수 있다. 중지되는 경우, 중지된 DMA 전송은 재개 명령을 이용하여 재개될 수 있다. DMA 프로세스의 완료에 도달하면, 인터럽트 신호가 호스트, 마이크로 제어기, 또는 코어 프로세서(DSP)에 다시 결합되어 DMA 제어기의 상태를 체크한다.

데이터 메모리(302)의 한 쪽 채널에 저장된 데이터의 워드의 비트 폭은 시스템 버스(200)의 데이터 시스템 버스 폭(DSBW)과 일치한다. 프로그램 메모리(304)의 한 쪽 채널에 저장된 코드의 워드의 비트 폭은 DSP(300)의 명령어 폭과 일치하는 것이 바람직하다. 일 실시예에서, 데이터 메모리(302)의 어느 한 쪽에 저장된 데이터의 워드의 폭은 64 비트이고, 프로그램 메모리(304)의 어느 한 쪽 채널에 저장된 프로그램 코드의 워드의 폭은 40 비트이다. 따라서 코어 DMA 제어기(203)는 또한 글로벌 버퍼 메모리(210)로부터 수신한 64 비트의 프로그램 코드를 40 비트 워드 폭으로 변환(즉, 선택적으로 패킹)한다. DMA 제어기는 한 번에 40 비트를 DSP에 제공한다. 코어 DMA 제어기(203) 내에 포함된 FIFO는 64 비트에서 40 비트로의 변환을 수행한다. 글로벌 버퍼 메모리(210)로부터 판독되는 코드의 64 비트의 워드 5개(전체 320 비트)마다 코드의 4 비트의 워드 8개(전체 320 비트)로 변환된다. 즉, 비트들은 데이터 시스템 버스 폭과 프로그램 메모리(304)의 워드 폭의 최소 공배수인 워드로 패킹(packing)된다.

분산형 DMA 제어기 각각은 판독시에 소스 어드레스를 글로벌 버퍼 메모리에 제공하며, 기록시에 목적지(destination) 어드레스를 글로벌 버퍼 메모리에 제공한다. 코어 DMA 제어기(203)는 또한 데이터 메모리(302) 및 프로그램 메모리(304)에 대한 목적지 어드레스를 이들 메모리에 대한 기록시에 생성하거나, 또는 소스 어드레스를 이들 메모리로부터의 판독시에 생성한다. 프로그램 메모리(304)는 40 비트의 폭일 수 있는 어드레스 워드를 갖는 40 비트 어드레싱(addressing)이 가능하다. 데이터 메모리(302)는 64 비트 폭일 수 있는 어드레스 워드를 메모리 내에 갖는 64 비트 어드레싱이 가능하다. 따라서 코어 DMA 제어기(203)는 글로벌 버퍼 메모리(210)와 데이터 메모리(302) 또는 프로그램 메모리(304) 사이에서 DMA 전송을 수행하는 경우 소스 및 목적지 어드레스를 제공한다.

이제 도 4를 참조하면, 글로벌 버퍼 메모리(210)에 대한 예시 메모리 맵의 블록도가 도시되어 있다. 데이터의 워드 또는 블록이 데이터 시스템 버스 폭(DSBW)과 일치하는 크기를 갖는 글로벌 버퍼 메모리에 저장되고 그로부터 판독된다. 일 실시예에서, 데이터 시스템 버스 폭 및 글로벌 버퍼 메모리 내의 어드레싱 가능한 메모리 위치로의 데이터 워드의 폭은 64 비트이다. 즉, 어드레스가 글로벌 버퍼 메모리에 제공되는 경우, 64 비트가 한꺼번에 판독되거나 한꺼번에 기록될 수 있다. 글로벌 버퍼 메모리는 23 비트 폭의 어드레스 워드로 64 비트 경계(boundary) 상에서 어드레싱될 수 있다.

도 4의 예시 메모리 맵에서, 글로벌 버퍼 메모리(210)는 제1 시작 어드레스(B1) 및 제1 끝 어드레스(E1) 사이의 데이터를 한 세트의 직접 메모리 액세스 가능한 데이터(DMAD 1로 칭함)로서 포함한다. 예컨대 DMAD 1은 호스트 포트 또는 멀티채널 직렬 포트로부터 수신된 데이터, 또는 코어 프로세서들(200A 내지 200N) 중 하나에 의해 기록된 데이터를 나타낸다. 도 4의 예시 메모리 맵은 또한 제2 시작 어드레스(B2) 및 제2 끝 어드레스(E2) 사이의 데이터를 제2 세트의 직접 메모리 액세스 가능한 데이터(DMAD 2로 칭함)로서 포함한다. 도 4의 예시 메모리 맵은 또한 글로벌 버퍼 메모리(210)가 제3 시작 어드레스(E3) 및 제3 끝 어드레스(E3) 사이의 데이터를 제3 세트의 직접 메모리 액세스 가능한 데이터(DMAD 3으로 칭함)로서 포함하는 것으로 도시하고 있다.

글로벌 버퍼 메모리는 또한 DMA 기술자 테이블(401)을 포함한다. DMA 기술자 테이블(401)은 정보에 관한 하나 이상의 DMA 기술 리스트(DMA descriptive list; 402A 내지 402N)들을 포함하며, 이는 글로벌 버퍼 메모리(210)에 저장된 직접 메모리 액세스 가능한 데이터를 지칭한다. 하나 이상의 DMA 기술 리스트는 글로벌 버퍼 메모리(210)에 대해 정보를 DMA 전송하는 경우에 분산형 DMA 제어기 각각에 의해 사용된다. DMA 기술 리스트들(402A 내지 402N) 각각은 글로벌 버퍼 메모리 내의 데이터, 코드, 상태, 또는 그 밖의 정보에 대한 시작 포인터(starting pointer)들을 포함한다. DMA 기술 테이블(401) 내의 하나 이상의 DMA 기술 리스트들(402A 내지 402N)은 DMAD 1, DMAD 2 및 DMAD 3의 직접 메모리 액세스 가능한 데이터를 가리킨다. 직접 메모리 액세스 가능한 데이터가 더 이상 유효하지 않거나 사용 가능하지 않게 되면, DMA 기술 리스트는 무효화(invalidate) 또는 소거(clear)된다.

도 4에서, DMA 기술자 테이블(401)은 주어진 채널을 처리하기 위하여 글로벌 버퍼 메모리(210) 내에 저장된 예시적인 DMA 기술 리스트(402A)를 포함한다. DMA 기술 리스트(402N)는 이미 직렬 DMA 제어기 또는 호스트 DMA 제어기에 의하여 이들 각각의 포트로부터 글로벌 버퍼 메모리로 정보를 DMA하는 데 사용되었을 수 있다. 동일한 DMA 기술 리스트는 글로벌 버퍼 메모리로부터 하나 이상의 코어 프로세서(200A 내지 200N)의 로컬 메모리, 데이터 메모리 및 프로그램 메모리로 정보를 DMA하는 데 사용될 수 있다. 이러한 경우에, DMA 기술 리스트는 DMA 기술 리스트(402N)에 대한 링크를 포함할 수 있다.

도 4는 채널을 처리하기 위한 예시적인 DMA 기술 리스트(402A)를 도시한다. 예시적인 DMA 기술 리스트(402A)는 송신 프로그램 포인터(transmit program pointer; 410), 샘플 유형 기술(sample type description; 412), 상태 정보 포인터(state information pointer; 414), 기타 제어 포인터(miscellaneous control pointer; 416), 시작 데이터 메모리 포인터(beginning data memory pointer; 418) 및 연결 리스트 포인터(linked list pointer; 420)를 포함한다. 송신 프로그램 포인터(410)는 주어진 채널을 처리하기 위하여 사용될 코드에 대한 시작 어드레스를 가리킨다. 샘플 유형 기술(412)은 직렬 포트상의 데이터 샘플 유형이 4 비트, 8 비트, 16 비트인지의 여부 및 압축되었는지의 여부, 그리고 압축되었다면 어떠한 압축 유형이 사용되었는지에 관하여 기술한다. 상태 정보 포인터는, 현재 계산에서 사용될 수 있는 채널에 대한 데이터 샘플의 최종 계산 상태의 데이터를, 그것이 메모리 내에 존재하는 경우에 가리킨다. 기타 제어 포인터(416)는 데이터 샘플에 관한 기타 제어 정보, 코드, 상태 정보 또는 글로벌 버퍼 메모리(210)의 안팎으로 직접 메모리 액세스되는 그 밖의 정보를 가리킨다. 시작 메모리 위치(418)는 예시적인 메모리 맵의 B1, B2 또는 B3과

같은 데이터의 첫 워드 또는 블록을 가리키는, 직접 메모리 액세스될 데이터의 시작 어드레스를 제공한다. 그 대신, 연결 리스트 포인터(420)는 글로벌 버퍼 메모리 안팎으로 직접 메모리 액세스될 필요가 있는 데이터 또는 그 밖의 정보를 위한 다른 DMA 기술 리스트(420N)를 가리킨다(422). 이러한 방식으로 DMA 기술 리스트들은 함께 연결될 수 있다.

도 5를 참조하면, 직렬 DMA 제어기(207)의 블록도가 도시되어 있다. 직렬 DMA 제어기(207)는 수신 FIFO 버퍼(502), 송신 FIFO 버퍼(504), 채널 레지스터(505), 데이터 카운터(506), 상태/제어 레지스터(507), 제어 로직(508) 및 TDM 리매퍼 메모리(510)를 포함한다. 직렬 포트를 통해 송신 및 수신된 데이터는 압축 알고리즘에 의하여 압축될 수 있다. 데이터가 음성 데이터인 경우에는, 압축 알고리즘은 A-law 압축 또는 Mu-law 압축일 수 있다. 압축되지 않은 형태의 음성 데이터는 PCM 또는 펄스 코드 변조된(pulse-code modulated) 데이터이다. 또한, 데이터는 각각의 통신 채널에 대하여 직렬 데이터 스트림 상에서 타임 슬롯(time slot)으로 시분할 멀티플렉싱된다. 본 발명의 분산형 DMA는 특히 주어진 채널에 대한 데이터 블록을 지원하기에 적합하므로, 직렬 포트는 각각의 채널에 대하여 직렬 데이터 스트림으로부터 데이터를 인터리빙(interleaving) 및 역인터리빙(deinterleaving)하는 것을 제공한다.

직렬 DMA 제어기(207)는 데이터의 병렬 비트를 수신 및 송신하기 위하여 버 스 중재기(201A)를 통해 시스템 버스(200)에 연결된다. 직렬 DMA 제어기(207)는 직렬 데이터 스트림을 수신 및 송신하기 위하여 전화 네트워크에 연결된다. 직렬 DMA 제어기(207)는 또한 시스템 버스(200)를 통해 직접 메모리 액세스를 요청 및 제어하기 위한 DMA 제어 신호를 포함한다. 이러한 DMA 제어 신호는 DMA 요청(DMA REQ), DMA 승인(DMA ACK), 인터럽트 요청(INT) 및 판독/기록 스트로브(strobe) 신호(R/W)를 포함하며, 이들은 버스 중재기(201A)를 통해 시스템 버스(200)의 제어 라인에 연결된다. DMA 제어 신호는 제어 로직 블록(508)에 연결된다. 제어 로직 블록(508)은 DMA REQ, INT 및 R/W 스트로브 신호를 생성하고, DMA ACK 신호를 수신한다. 직렬 데이터 제어기(207)의 병렬 데이터 포트는 버스 중재기(201A)에 연결된다. 수신 FIFO 버퍼(502), 송신 FIFO 버퍼(504), 채널 레지스터, 데이터 카운터(506) 및 상태/제어 레지스터(507)는 직렬 DMA 제어기(207)의 병렬 데이터 포트를 통해서 버스 중재기(201A)에 연결된다. 일 실시예에서 수신 FIFO 버퍼(502) 및 송신 FIFO 버퍼(504)는 1 내지 2 엔트리(64비트)의 깊이를 가지며, TDM 리매퍼 메모리(510) 및 글로벌 버퍼 메모리(210)에 대하여 버스트(burst) 기록 및 판독을 제공한다. 직렬 DMA 제어기(207)의 TDM 리매퍼 메모리(510)는 멀티채널 직렬 데이터 포트를 통해 전화 네트워크에 연결되어 전화 네트워크상의 직렬 데이터 스트림을 송신 및 수신한다.

TDM 리매퍼 메모리(510)는 전화 네트워크로부터 직렬 데이터 스트림을 수신하는 경우 직/병렬(serial to parallel) 변환을 제공한다. TDM 리매퍼 메모리(510)는 전화 네트워크 상으로 직렬 데이터를 송신하는 경우 병/직렬(parallel to serial) 변환을 제공한다. TDM 리매퍼 메모리는 송신을 위하여 직렬 데이터 스트림의 타임 슬롯에 데이터를 적절하게 인터리빙한다. TDM 리매퍼 메모리는 데이터를 직렬 데이터 스트림의 타임 슬롯으로부터, 수신된 데이터에 적절하게 역인터리빙한다. TDM 리매퍼 메모리(510)는 또한 적절한 데이터를 메모리 위치에 재매핑하여 복수의 직렬 통신 채널이 지원되도록 한다. 직렬 데이터를 수신한 경우에, 주어진 채널에 대한 데이터의 전체 프레임이 글로벌 버퍼 메모리로 DMA될 수 있도록, 데이터를 타임 슬롯으로부터 적절한 메모리 위치에 역인터리빙함으로써 직렬 데이터 스트림이 TDM 리매퍼 메모리 내로 판독된다. 직렬 데이터를 송신하기 위하여, 데이터의 전체 프레임은 TDM 리매퍼 메모리에 저장되고, 따라서 이는 한번에 여러 블록으로 판독될 수 있고, 주어진 채널에 대하여 적절한 타임 슬롯에 인터리빙될 수 있다.

직렬 DMA 제어기의 상태/제어 레지스터(507)는 판독되는 시점에 직렬 포트(206) 및 직렬 DMA 제어기에 관한 상태 정보를 제공한다. 상태/제어 레지스터(507)는 수신 FIFO 버퍼 전체 플래그(full flag) 및 송신 FIFO 버퍼 전체 플래그를 포함하는 다수의 플래그를 유지한다. 버퍼 전체 플래그가 세팅되면, 상기 플래그가 소거될 때까지 각각의 버퍼에 더 이상의 데이터가 저장되지 못한다. 채널에 관한 특정한 상태를 원하는 경우, 채널 레지스터(505)는 특정 채널 번호로 세팅된다. 채널 번호가 채널 레지스터에 의하여 선택되지 않은 경우, 일반적인 상태가 상태/제어 레지스터(507)가 판독될 때에 제공된다. 상태/제어 레지스터(507)에 기록하는 경우, 직렬 DMA 제어기(207) 및 직렬 포트(206)는 설정되거나 업데이 트될 수 있다. 데이터 카운터(506)는 전송이 완료되었는지를 판정하기 위하여 DMA 전송에서의 바이트 카운트(byte count)를 센다. 데이터 카운트(506)의 카운트 값은 또한 글로벌 버퍼 메모리(210)에 대한 어드레스의 생성을 용이하게 한다.

이제 도 6a 및 도 6b를 참조하면, TDM 리매핑 메모리(510)의 기능이 도시되어 있다. 도 6a는 멀티채널 직렬 포트(207)의 직렬 포트상의 예시적인 직렬 데이터 스트림을 도시하고 있다. 직렬 포트 상의 직렬 데이터 스트림은 시분할 멀티플렉싱(TDM)된다. 통신 채널(CH1 내지 CHN)은 프레임 1 내지 프레임 M상의 프레임 동기화 펄스(FS) 간에 지정된 타임 슬롯을 할당받는다. 일 실시예에서 각각의 채널 타임 슬롯은 8 데이터 비트를 갖는 하나의 DS0 데이터 바이트이다. 대체 실시예에서, 각각의 채널 타임 슬롯은 16 데이터 비트이다. 본 발명은 채널당 타임 슬롯당 다른 데이터 비트 수로 구성될 수도 있다.

도 6b는 데이터 프레임을 채널 1 내지 채널 N 각각의 데이터에 리매핑하는 것, 또는 그 역의 예를 도시하고 있다. 프레임 1 내지 프레임 M 상의 채널 1 데이터는 함께 조직된다. 프레임 1 내지 프레임 M 상의 채널 2 데이터는 함께 조직되고 이러한 과정이 채널 N까지 계속된다. N개의 통신 채널에 대하여 M개의 프레임이 TDM 리매퍼 메모리에 저장

된다. 수신 도중에, 채널 1의 N개의 DS0 워드가 TDM 리매퍼에 저장되자마자 채널 1 데이터의 하나의 프레임이 DMA를 통하여 글로벌 버퍼 메모리(210)에 기록될 수 있다. TDM 리매퍼 메모리는 수신된 DS0 워드를 각각의 채널에 대한 데이터 프레임에 매핑한다. 송신 도중에, 하나의 데이터 프레임이 N개의 채널 각각에 대한 TDM 리매퍼 메모리에 저장되자마자 직렬 TDM 데이터의 하 나의 프레임이 전화 네트워크에 전송될 수 있다. TDM 리매퍼 메모리는 N개의 채널에 대한 데이터 프레임을 각 채널에 대한 DS0 워드에 매핑한다.

이제 도 7을 참조하면, 코어 DMA 제어기(203)의 블록도가 도시되어 있다. 코어 DMA 제어기(203)는 하나 이상의 코어 DMA 제어기(203A 내지 203N)의 하나의 예이다. 코어 DMA 제어기(203)는 수신 FIFO 버퍼(702), 송신 FIFO 버퍼(704), 상태/제어 레지스터(705), 데이터 카운터(706), 큐(queue) 레지스터/카운터(707), 제어 로직(708), 디코더(711), 한 쌍의 멀티플렉서(712 내지 713) 및 인코더(714)를 포함한다. 제어 로직(708)은, 하나의 접속으로서 프로세서 측에서 데이터 메모리(302) 및 프로그램 메모리(304)에 연결되는 DMA 제어 신호들(DMA REQ PROC., DMA ACK PROC., INT PROC. 및 R/W PROC.)과, 제2의 접속으로서 시스템 버스를 통하여 글로벌 버퍼 메모리(210)에 연결되는, 버스 측에서의 DMA 제어 신호(DMA REQ BUS, DMA ACK BUS, INT BUS 및 R/W BUS)를 갖는다. 코어 DMA 제어기(203)의 수신 FIFO 버퍼(702), 송신 FIFO 버퍼(704), 상태/제어 레지스터(705), 데이터 카운터(706), 큐 레지스터/카운터(707), 디코더(711) 및 인코더(714)는 한 쪽에서 코어 프로세서(200)의 데이터 메모리(302) 및 프로그램 메모리(304)에 연결된다. 코어 DMA 제어기(203)의 수신 FIFO 버퍼(702), 송신 FIFO 버퍼(704), 제어/컨트롤 레지스터(705), 데이터 카운터(706) 및 큐 레지스터/카운터(707)는 버스 측의 하나 이상의 버스 중재기(201A 내지 201N) 및 시스템 버스(200)를 통하여 글로벌 메모리(210)에 연결된다. 일 실시예의 수신 FIFO 버퍼(702) 및 송신 FIFO 버퍼(704)는 64 바이트의 깊이를 가지며, 글로벌 버퍼 메모리 및 코어 프로세서에 대하여 버스트 기록 및 판독을 제공한다.

본 발명의 일 실시예를 상기하자면, 프로그램 메모리(304)의 어느 한 쪽 채널에 저장된 프로그램 코드의 워드의 폭은 40 비트인 반면, 데이터 메모리(302)의 어느 한 쪽 채널에 저장된 데이터의 워드의 폭은 64 비트이다. 어느 경우에나, 코어 DMA 제어기(203)는 글로벌 버퍼 메모리(210)로부터 수신한 64 비트의 프로그램 코드를 40 비트 워드 폭으로 변환시킨다. 64 비트 폭을 갖는 프로그램 코드는 8개의 8 비트 바이트로 수신 FIFO에 저장된다. 프로그램 코드는, 코드의 5개의 8 비트 바이트(즉 40 비트)가 한번에 판독되어 프로그램 메모리(304)에 저장될 수 있도록, 프로세서에 의해 수신 FIFO로부터 판독된다. 그 대신에, 이는 수신 FIFO에 저장되는 5개의 64 비트 워드(320 비트) 및 수신 FIFO로부터 판독되는 8개의 40 비트 워드(320 비트)로 간주될 수도 있다.

또한, 코어 DMA 제어기는 글로벌 버퍼 메모리로부터 수신한 데이터를 데이터 메모리(304)에 기록하기 전에 이를 압축 해제(decompress)할 수 있고, 상기 데이터 메모리(304)로부터 판독한 데이터를 글로벌 버퍼 메모리(210)에 기록하기 전에 이를 압축(compress)할 수 있다. 멀티플렉서(712)는 수신 FIFO 버퍼(702)로부터의 정보를 데이터 메모리 및 프로그램 메모리에 직접 연결할 것인지, 아니면 상기 정보를 디코딩하는 디코더(711)의 디코딩된 두 개의 출력 중 어느 하나를 연결할 것인지를 선택한다. 멀티플렉서(712)는 데이터 카운터(706), 상태/제어 레지스터(705) 및 큐 레지스터/카운터(707)로부터의 정보가 프로세서 또는 데이터 메모리 및 프로그램 메모리에 연결되도록 할 수도 있다. 멀티플렉서(713)는 데이 터 메모리, 프로그램 메모리 또는 DSP로부터의 정보가 송신 FIFO 버퍼(702)에 직접 연결될지, 아니면 상기 정보를 인코딩하는 인코더(714)의 두 개의 인코딩된 출력 중 어느 하나를 연결할지를 선택한다. 일 실시예에서, 압축 해체된 포맷이 선형 데이터 또는 펄스 코드 변조된 데이터인 반면, 압축 포맷은 A-law 또는 Mu-law 중 하나의 비선형 압축 포맷일 수 있다. DMA 기술자 리스트에 응답하여 상태/제어 레지스터(705) 내의 비트를 세팅함으로써 압축 및 압축 해제를 선택할 수 있다.

코어 DMA 제어기 내에서 데이터를 인코딩(압축) 및 디코딩(압축 해제)함으로써, 시리얼 포트로부터의 데이터는 시리얼 포트로부터 시스템 버스를 통해 글로벌 버퍼 메모리까지, 그리고 그 후 시스템 버스를 통해 코어 DMA 제어기에 이르기까지 압축된 상태를 유지할 수 있다. 데이터를 SOC(100)내에서 가능한 한 오랫동안 압축된 상태로 유지함으로써, 데이터 전송에 있어 대역폭을 증가시킬 수 있다.

코어 DMA 제어기의 상태/제어 레지스터(705)는 판독시에 코어 프로세서(200) 및 코어 DMA 제어기 각각에 관한 상태 정보를 제공한다. 상태/제어 레지스터(705)는 수신 FIFO 버퍼 전체 플래그 및 송신 FIFO 버퍼 전체 플래그를 포함하는 다수의 플래그를 유지한다. 버퍼 전체 플래그가 세팅되면, 상기 플래그가 소거될 때까지 각각의 버퍼에 더 이상의 데이터가 저장되지 못한다. 상태/제어 레지스터(705)에 기록하는 경우, 코어 DMA 제어기(203)의 제어가 설정 및 업데이트 될 수 있다. 데이터 카운터(706)는 DMA 전송시에 바이트 카운트를 세어 언제 전송이 완료될지를 판정한다. 또한 데이터 카운트(706)의 카운트 값은 글로벌 버퍼 메모리(210)에 대한 어드레스의 생성을 용이하게 한다.

글로벌 버퍼 메모리와 코어 프로세서의 데이터 메모리 또는 프로그램 메모리 사이에서 데이터를 직접 메모리 액세스하는 경우, 각각의 코어 DMA 제어기는 호스트 포트 또는 시리얼 포트로부터의 데이터를 글로벌 버퍼 메모리에 직접 메모리 액세스하는 데 사용되었던 것과 동일한 DMA 기술자 리스트를 사용한다. 부가적인 DMA 기술자 리스트는 불필요하다.

이제 도 8을 참조하면, 호스트 DMA 제어기(215)를 포함하는 호스트 포트(214)가 도시되어 있다. 호스트 DMA 제어기(215)는 수신 FIFO 버퍼(802), 송신 FIFO 버퍼(804), 상태/제어 레지스터(805), 데이터 카운터(806), 큐 레지스터/카운터(807), 흐름 제어를 제공하는 제어 로직(808) 및 I/O 포트 레지스터를 포함한다. 제어 로직(808)은 데이터 흐름 제어를 제공하고 DMA 제어 구조를 제공하기 위한 로직을 포함한다. DMA 제어 구조는 큐(queue)를 처리하는 데 사용된다. 상태/제어 레지스터(807)는 제어를 위한 액세스 및 상태 정보를 외부 호스트에 제공하고, 상태 통지(status notification) 및 큐 카운터 정보를 내부 마이크로 제어기(223)에 제공한다. 일 실시예의 수신 FIFO 버퍼(802) 및 송신 FIFO 버퍼(804)는 64 바이트의 깊이를 갖는다. 수신 FIFO 버퍼(802) 및 송신 FIFO 버퍼(804)는 호스트 프로세서와 SCO(100) 사이에서의 버스트 기록 및 판독을 제공한다. 도 8에 도시된 호스트 포트(214)는 호스트 프로세서에 대하여 선택적으로 인터페이스하기 위한 I/O 포트 레지스터(810)를 더 포함한다. DMA 제어기(215)의 제어 로직(808)은 중재기(201) 및 시스템 버스(200)를 통하여 글로벌 버퍼 메모리(210)에 연결되는 버스 측에 DMA 제어 신호들(DMA REQ BUS, DMA ACK BUS, INT BUS 및 R/W BUS)을 갖는다. 제어 로직(808)은 호스트에 연결되는 호스트 측에 DMA 제어 신호들(DMA REQ HOST, DMA ACK HOST, INT HOST 및 R/W HOST)을 갖는다. 수신 FIFO(802), 송신 FIFO(804), 데이터 카운터(806), 상태/제어 레지스터(805) 및 큐 레지스터/카운터(807)는 한 쪽에서 버스 중재기(201)에 연결된다. 수신 FIFO(802), 송신 FIFO(804) 및 데이터 카운터(806)는 반대쪽에서 I/O 버퍼(810)에 연결된다. 이러한 방식으로, 패킷 페이로드의 병렬 데이터는 호스트와 시스템 버스(200) 사이에서 글로벌 버퍼 메모리(210) 안팎으로 흐를 수 있다.

호스트 포트(214)의 I/O 포트 레지스터(810)는 호스트 프로세서가 데이터를 기록하고 판독하는 지점을 제공한다.

호스트 DMA 제어기(215)의 상태/제어 레지스터(805)는 판독시에 호스트 포트(214) 및 호스트 DMA 제어기(215)에 관한 상태 정보를 제공한다. 상태/제어 레지스터(805)는 수신 FIFO 버퍼 전체 플래그 및 송신 FIFO 버퍼 전체 플래그를 포함하는 다수의 플래그를 유지한다. 버퍼 전체 플래그가 세팅되면, 상기 플래그가 소거될 때까지 각각의 버퍼에 더 이상의 데이터가 저장되지 못한다. 상태/제어 레지스터(805)에 기록하는 경우, 호스트 DMA 제어기(215) 및 호스트 포트(214)의 제어가 설정 및 업데이트될 수 있다. 데이터 카운터(806)는 DMA 전송시에 바이트 카운트를 세어 전송이 언제 완료될지를 판정한다. 데이터 카운트(806)의 카운트 값은 또한 글로벌 버퍼 메모리(210)에 대한 어드레스의 생성을 용이하게 한다.

이와 같이 본 발명의 바람직한 실시예가 설명된다. 당업자가 인식할 수 있는 바처럼, 본 발명은 다수의 장점을 갖는다. 본 발명의 한 가지 장점은 넓은 시 스템 버스, 시리얼 데이터의 재매핑 및 처리중(on the fly)인 데이터의 압축/압축해제로 인해 글로벌 버퍼 메모리로의 대역폭이 증가된다는 점이다. 본 발명의 다른 장점은 버스 중재에 대해 공통의 표준을 사용함으로써 중재가 단순화되고, 직접 메모리 액세스 제어기의 분산으로 인해 중재가 개선된다는 점이다.

본 발명을 VOP 시스템 온 칩의 특정한 실시예에 따라 설명하였지만, 이는 다른 시스템 온 칩에도 마찬가지로 적용될 수 있다. 또한, 본 발명은 하드웨어, 소프트웨어, 펌웨어(firmware) 또는 이들의 조합으로 구현될 수 있고, 시스템, 서브 시스템, 콤포넌트 또는 이들의 서브 콤포넌트에서 이용될 수 있다. 소프트웨어로 구현되는 경우, 본 발명의 요소들은 본질적으로 필요한 작업을 수행하기 위한 코드 세그먼트들이다. 프로그램 또는 코드 세그먼트는 프로세서 판독 가능 매체(processor readable medium)에 저장되거나, 반송파에 수록된 컴퓨터 데이터 신호에 의해 전송 매체 또는 통신 링크를 통해 전송될 수 있다. '프로세서 판독 가능 매체'는 정보를 저장하거나 전송할 수 있는 모든 매체를 포함할 수 있다. 프로세서 판독 가능 매체의 예에는 전자 회로, 반도체 메모리 소자, ROM, 플래시 메모리, EROM(Erasable ROM), 플로피 디스켓, CD-ROM, 광 디스크, 하드 디스크, 광섬유 매체, RF(radio frequency) 링크 등이 포함된다. 컴퓨터 데이터 신호는, 전자 네트워크 채널, 광섬유, 대기, 전자기, 라디오 링크 등과 같은 전송 매체를 통해 전달되는 모든 신호를 포함할 수 있다. 코드 세그먼트는 인터넷, 인트라넷 등과 같은 컴퓨터 네트워크를 통해 다운로드될 수 있다. 어느 경우나, 본 발명은 이러한 실시예에 의해 제한되도록 해석되어서는 아니되며, 오히려 이하의 청구범위에 따라 해석되어야 한다.

(57) 청구의 범위

청구항 1.

시스템 온 칩(system on chip)에 있어서,

버퍼 메모리와,

상기 버퍼 메모리에 연결된 시스템 버스와,

상기 시스템 버스와 연결된 복수의 버스 중재기들과,

상기 복수의 버스 중재기들에 연결된 복수의 기능 모듈들

을 포함하고,

상기 복수의 기능 모듈들 각각은 상기 복수의 버스 중재기들 중 하나의 버스 중재기에 연결하기 위한 DMA(Direct Memory Access) 제어기를 포함하며, 상기 DMA 제어기는 상기 버퍼 메모리에 대한 직접 메모리 액세스를 제공하는 시스템 온 칩.

청구항 2.
제1항에 있어서,

상기 복수의 버스 중재기들은 상기 복수의 기능 모듈들에 의해 상기 시스템 버스에 대한 액세스를 교섭하는 시스템 온 칩.

청구항 3.
제1항에 있어서,

상기 시스템 버스에 연결된 마이크로 제어기를 더 포함하는 시스템 온 칩.

청구항 4.
제1항에 있어서,

상기 시스템 온 칩은 VOP(Voice Over Packet) 시스템 온 칩이고,

상기 복수의 기능 모듈들은,

복수의 통신 채널들의 데이터를 처리하고, 처리하기 위한 데이터를 판독하고 처리 후에는 데이터를 기록하도록 상기 버퍼 메모리를 직접 메모리 액세스하기 위한 하나 이상의 신호 프로세서들과,

패킷 네트워크(packet network)에 연결되어 상기 패킷 네트워크와 패킷 페이로드(payload)를 송수신하고, 상기 버퍼 메모리로부터 데이터를 판독하고 상기 버퍼 메모리에 데이터를 기록하도록 상기 버퍼 메모리를 직접 메모리 액세스하기 위한 호스트 포트(host port)와,

전화 네트워크에 연결되어 상기 전화 네트워크와 데이터를 송수신하고, 상기 버퍼 메모리로부터 데이터를 판독하고 상기 버퍼 메모리에 데이터를 기록하도록 상기 버퍼 메모리를 직접 메모리 액세스하기 위한 멀티채널(multichannel) 직렬 포트

를 포함하는 시스템 온 칩.

청구항 5.
글로벌 버퍼 메모리에 대한 분산형 직접 메모리 액세스 방법에 있어서,

각각 직접 메모리 액세스 제어기를 갖는 하나 이상의 기능 유닛들을 제공하는 단계와,

상기 글로벌 버퍼 메모리를 직접 메모리 액세스하기 위하여 상기 하나 이상의 기능 유닛들에 의한 시스템 버스에 대한 액세스를 요청하는 단계와,

상기 하나 이상의 기능 유닛들에 의한 상기 시스템 버스에 대한 액세스를 중재하는 단계와,

상기 하나 이상의 기능 유닛들 중 하나에 의한 상기 시스템 버스에 대한 액세스를 취득하는 단계와,

상기 하나 이상의 기능 유닛들 중 상기 하나에 의해 상기 글로벌 버퍼 메모리에 대한 직접 메모리 액세스 접속을 설정하고, 상기 글로벌 버퍼 메모리 내의 메모리 위치로부터 데이터를 판독하거나 또는 상기 메모리 위치에 데이터를 기록하는 단계

를 포함하는 방법.

청구항 6.
제5항에 있어서,

상기 하나 이상의 기능 유닛들 중 하나에 의해 상기 글로벌 버퍼 메모리를 직접 메모리 액세스하기 전에 병렬 데이터 스트림(parallel data stream)으로 직렬 데이터 스트림을 재매핑(remapping)하는 단계

를 더 포함하는 방법.

청구항 7.
제5항에 있어서,

상기 하나 이상의 기능 유닛들 중 하나에 의해 상기 글로벌 버퍼 메모리를 직접 메모리 액세스한 후에, 직렬 데이터 스트림으로 병렬 데이터 스트림을 재매핑하는 단계

를 더 포함하는 방법.

청구항 8.
제5항에 있어서,

상기 단계들은 시스템 온 칩에서 수행되는 방법.

청구항 9.
제8항에 있어서,

상기 시스템 온 칩은 한 쪽 종단에서는 패킷 네트워크에 대하여, 반대 쪽 종단에서는 전화 네트워크에 대하여 인터페이스하는 VOP 시스템인 방법.

청구항 10.
분산형 직접 메모리 액세스 제어 시스템에 있어서,

메모리와,

시스템의 기능 블록들 사이에 분산된 복수의 직접 메모리 액세스 제어기들

을 포함하고,

상기 복수의 직접 메모리 액세스 제어기들은 상기 시스템의 상기 기능 블록들의 상기 메모리에 대한 직접 메모리 액세스를 제어하는 분산형 직접 메모리 액세스 제어 시스템.

청구항 11.
제10항에 있어서,

상기 메모리와 상기 복수의 직접 메모리 액세스 제어기들 사이에 결합된 시스템 버스

를 더 포함하고,

상기 시스템 버스는 데이터가 상기 메모리와 상기 시스템의 상기 기능 블록들 사이에서 흐를 수 있도록 하는 데이터 버스를 포함하는 분산형 직접 메모리 액세스 제어 시스템.

청구항 12.
제11항에 있어서,

상기 시스템 버스는 상기 메모리에 대한 직접 메모리 액세스를 요청하고 상기 메모리에 대한 직접 메모리 데이터 흐름을 제어하기 위한 DMA 제어 라인을 더 포함하는 분산형 직접 메모리 액세스 제어 시스템.

청구항 13.
제10항에 있어서,

상기 메모리는 직접 액세스 가능한 메모리이고, 직접 메모리 액세스 제어기를 포함하는 분산형 직접 메모리 액세스 제어 시스템.

청구항 14.
제13항에 있어서,

상기 메모리의 상기 직접 메모리 액세스 제어기는,

수신 데이터를 저장하기 위한 수신 FIFO(First In First Out) 버퍼와,

송신 데이터를 저장하기 위한 송신 FIFO 버퍼와,

직접 메모리 액세스에 의해 상기 메모리로 송신되고 상기 메모리로부터 수신된 데이터 유닛들의 수를 세기 위한 데이터 카운터와,

직접 메모리 액세스에 의한 데이터 송신 및 수신을 제어하기 위하여 상기 수신 FIFO 버퍼, 상기 송신 FIFO 버퍼 및 상기 데이터 카운터에 연결된 제어 로직(control logic)

을 포함하는 분산형 직접 메모리 액세스 제어 시스템.

청구항 15.
제10항에 있어서,

상기 복수의 직접 메모리 액세스 제어기들은,

수신 데이터를 저장하기 위한 수신 FIFO(First In First Out)와,

송신 데이터를 저장하기 위한 송신 FIFO와,

직접 메모리 액세스에 의해 상기 메모리로 송신되고 상기 메모리로부터 수신된 데이터 유닛들의 수를 세기 위한 데이터 카운터와,

직접 메모리 액세스에 의한 데이터 송신 및 수신을 제어하기 위하여 상기 수신 FIFO 버퍼, 상기 송신 FIFO 버퍼 및 상기 데이터 카운터에 연결된 제어 로직

을 포함하는 분산형 직접 메모리 액세스 제어 시스템.

청구항 16.
제15항에 있어서,

상기 복수의 직접 메모리 액세스 제어기들 중 하나는 병렬 데이터를 직렬 데이터로, 직렬 데이터를 병렬 데이터로 재매핑하기 위한 리매퍼 메모리(remapper memory)를 더 포함하는 분산형 직접 메모리 액세스 제어 시스템.

청구항 17.
메모리 내의 정보에 대한 직접 메모리 액세스를 제어하기 위한 직접 메모리 액세스 기술자(descriptor) 테이블에 있어서,

상기 메모리 내에 저장된 하나 이상의 직접 메모리 액세스 기술자 리스트들

을 포함하고,

상기 하나 이상의 직접 메모리 액세스 기술자 리스트들 각각은,

상기 메모리로부터 또는 상기 메모리로 데이터를 직접 메모리 액세스하기 위한, 상기 메모리 내의 시작 어드레스를 가리키는 포인터(pointer)

를 포함하는 직접 메모리 액세스 기술자 테이블.

청구항 18.
제17항에 있어서,

상기 직접 메모리 액세스 기술자 리스트들 중 하나는, 직접 메모리 액세스될 데이터의 유형에 관한 정보를 더 포함하는 직접 메모리 액세스 기술자 테이블.

청구항 19.
제17항에 있어서,

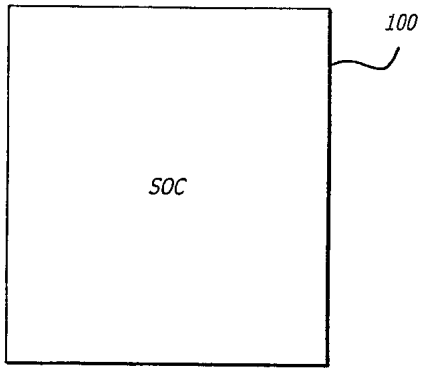
상기 직접 메모리 액세스 기술자 리스트들 중 하나는, 상기 메모리로부터 또는 상기 메모리로 이전 상태 정보(prior state information)를 직접 메모리 액세스하기 위한, 상기 메모리 내의 시작 어드레스를 가리키는 포인터를 더 포함하는 직접 메모리 액세스 기술자 테이블.

청구항 20.
제17항에 있어서,

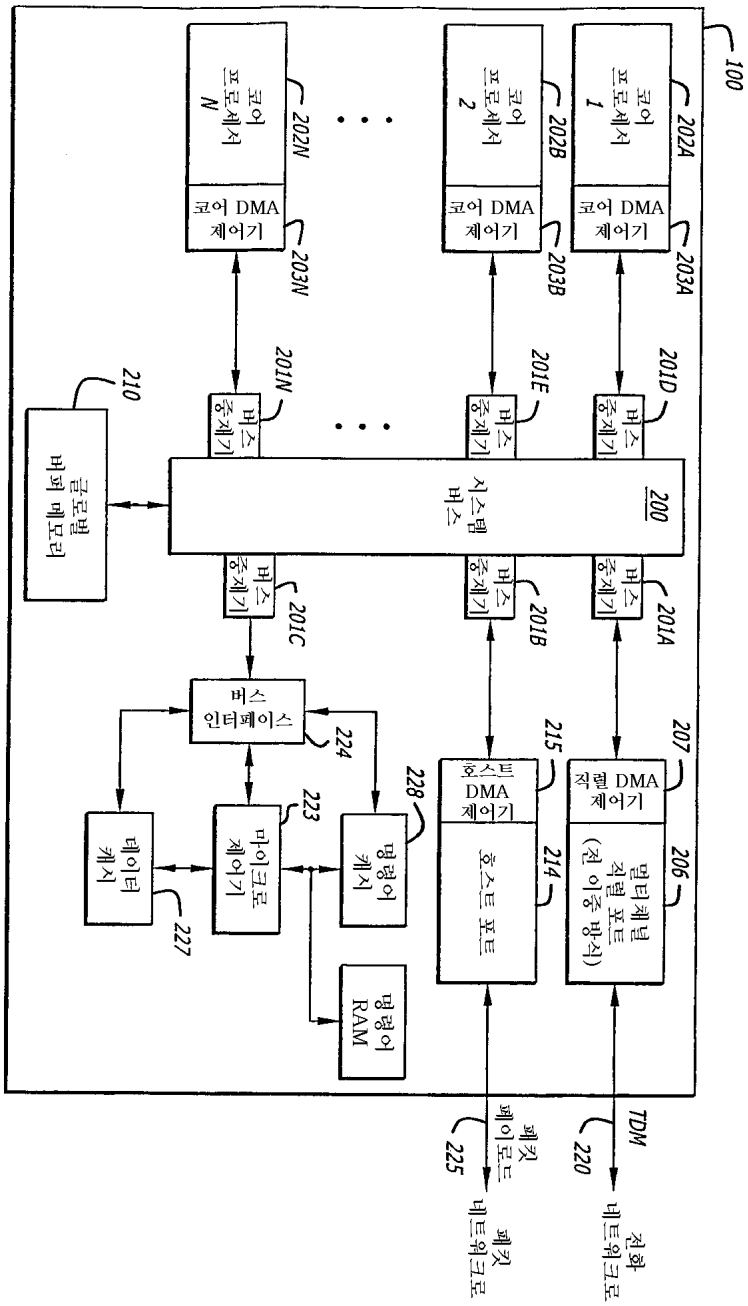
상기 직접 메모리 액세스 기술자 리스트들 중 하나는, 상기 메모리로부터 또는 상기 메모리로 프로그램 코드를 직접 메모리 액세스하기 위한, 상기 메모리 내의 시작 어드레스를 가리키는 포인터를 더 포함하는 직접 메모리 액세스 기술자 테이블.

도면

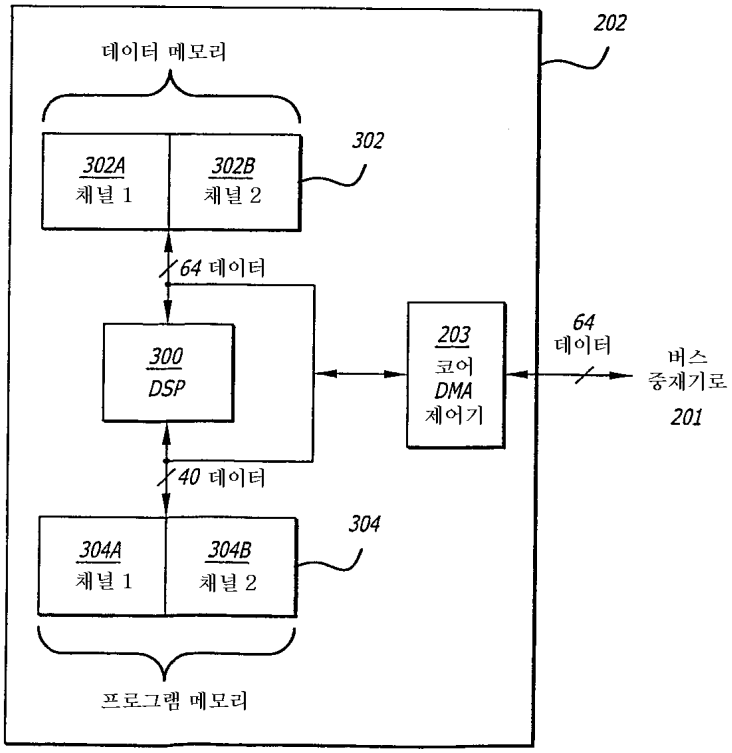
도면1



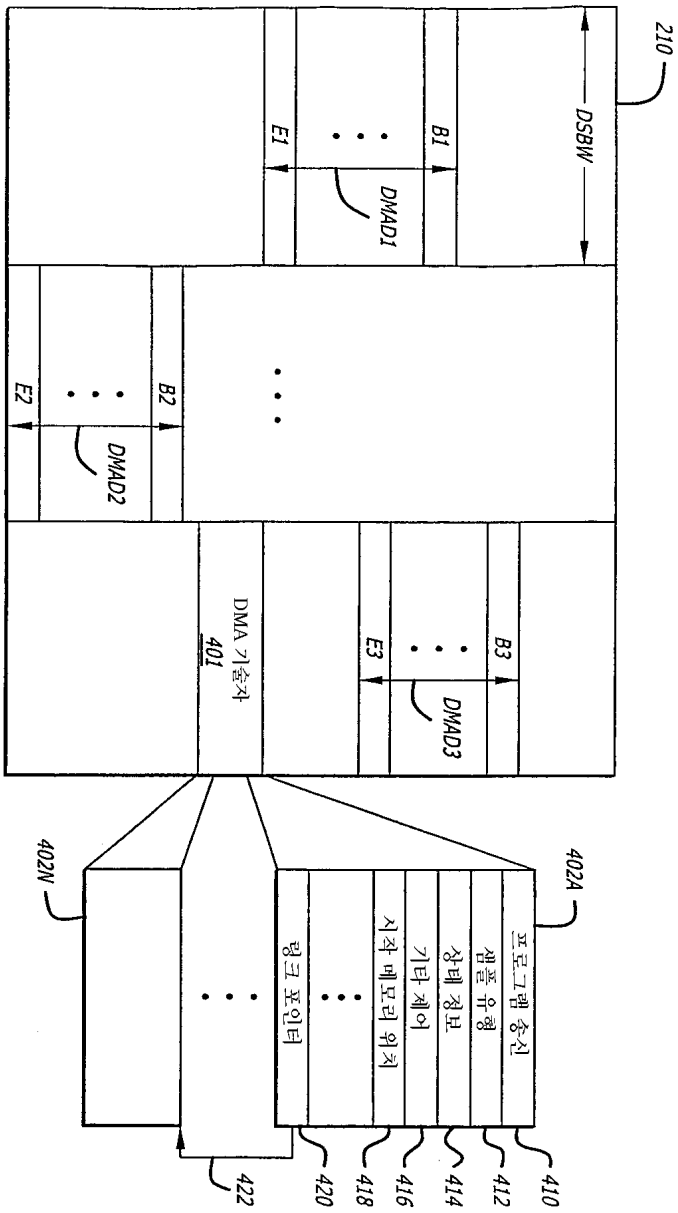
도면2


100
코어 프로세서 1
2024A
코어 DMA 제어기
203A
코어 프로세서 2
202B
코어 DMA 제어기
203B
코어 프로세서 N
202N
코어 DMA 제어기
203N
버스 중재기
201D
201E
201N
시스템 버스
200
글로벌 버퍼 메모리
210
버스 중재기
201A
201B
201C
버스 인터페이스
224
직렬 DMA 제어기
207
멀티채널 직렬 포트 (전 이중 방식)
206
호스트 DMA 제어기
215
호스트 포트
214
명령어 캐시
228
마이크로 제어기
223
데이터 캐시
227
명령어 RAM
TDM
220
패킷 페이로드
225
전화 네트워크로
패킷 네트워크로

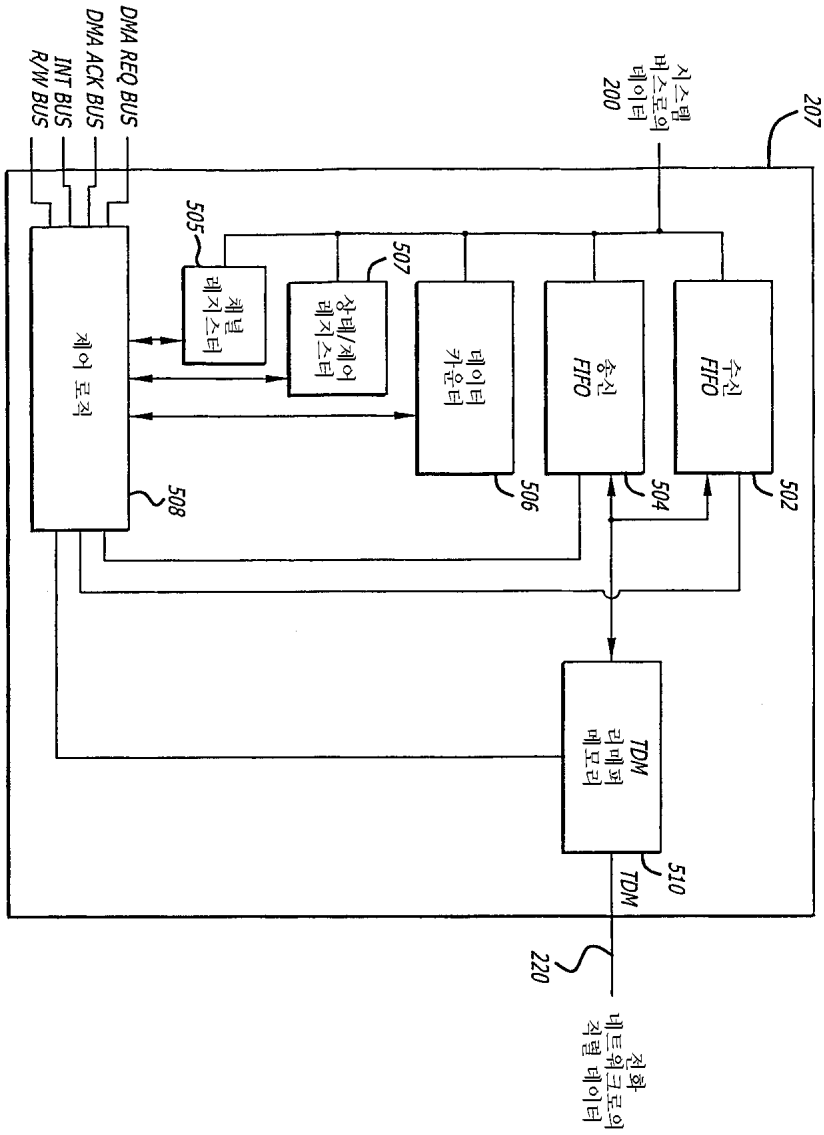
도면3


202
데이터 메모리
302
302A
채널 1
302B
채널 2
64 데이터
300
DSP
203
코어
DMA
제어기
64
데이터
버스
중재기로
201
40 데이터
304
304A
채널 1
304B
채널 2
프로그램 메모리

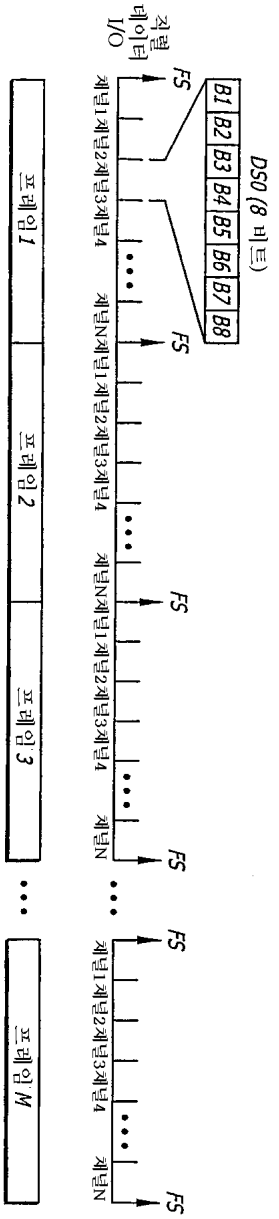
도면4


210
DSBW
B1
E1
DMAD1
B2
E2
DMAD2
B3
E3
DMAD3
DMA 기술자
401
402A
402N
프로그램 송신
샘플 유형
상태 정보
기타 제어
시작 메모리 위치
링크 포인터
410
412
414
416
418
420
422

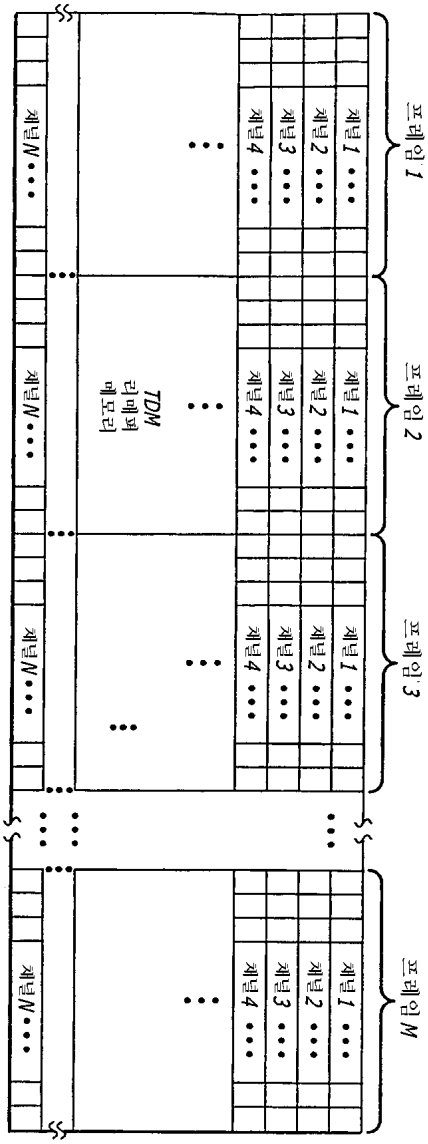
도면5


DMA REQ BUS
DMA ACK BUS
INT BUS
R/W BUS
시스템 버스로의 데이터
200
207
505
507
제어 로직
채널 레지스터
상태/제어 레지스터
데이터 카운터
송신 FIFO
수신 FIFO
508
506
504
502
TDM 리메퍼 메모리
TDM
510
220
직렬 네트워크로의 데이터

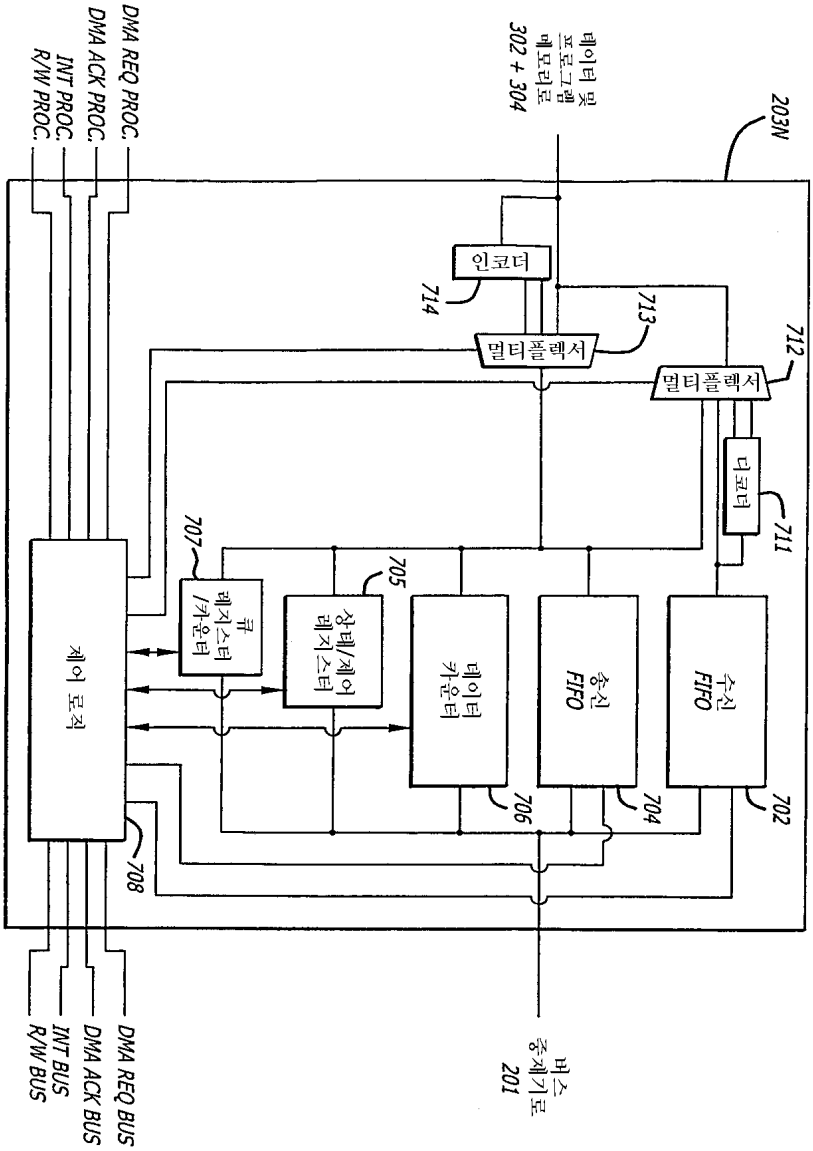
도면6a


직렬 데이터 I/O
FS
B1 B2 B3 B4 B5 B6 B7 B8
DS0 (8 비트)
채널1 채널2 채널3 채널4 ... 채널N
프레임 1
프레임 2
프레임 3
프레임 M

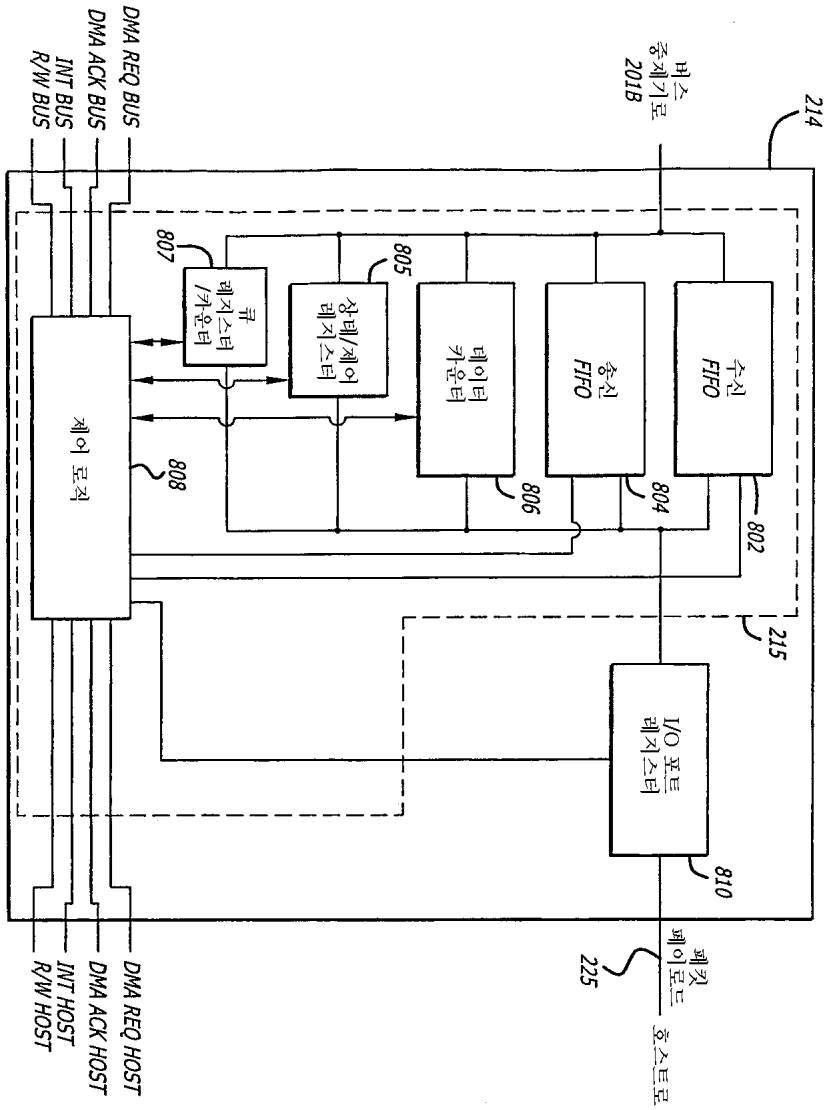
도면6b



도면7


DMA REQ PROC.
DMA ACK PROC.
INT PROC.
R/W PROC.
데이터 및 프로그램 메모리로
302 + 304
203N
인코더
714
713
712
멀티플렉서
멀티플렉서
디코더
711
707
705
큐 레지스터 /카운터
상태/제어 레지스터
데이터 카운터
송신 FIFO
수신 FIFO
제어 로직
706
704
702
708
DMA REQ BUS
DMA ACK BUS
INT BUS
R/W BUS
버스 중재기로
201

도면8


214
버스 중재기로 201B
DMA REQ BUS
DMA ACK BUS
INT BUS
R/W BUS
807
큐 레지스터 /카운터
805
상태/제어 레지스터
806
데이터 카운터
804
송신 FIFO
802
수신 FIFO
808
제어 로직
215
I/O 포트 레지스터
810
225
패킷 페이로드
호스트로
DMA REQ HOST
DMA ACK HOST
INT HOST
R/W HOST