



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2011년06월30일  
(11) 등록번호 10-1045475  
(24) 등록일자 2011년06월23일

(51) Int. Cl.

G06F 11/26 (2006.01) G06F 11/36 (2006.01)

G06F 11/00 (2006.01) G06F 13/28 (2006.01)

(21) 출원번호 10-2006-7014758

(22) 출원일자(국제출원일자) 2004년12월21일

심사청구일자 2009년12월17일

(85) 번역문제출일자 2006년07월21일

(65) 공개번호 10-2006-0126734

(43) 공개일자 2006년12월08일

(86) 국제출원번호 PCT/US2004/043491

(87) 국제공개번호 WO 2005/073855

국제공개일자 2005년08월11일

(30) 우선권주장

10/764,110 2004년01월23일 미국(US)

(56) 선행기술조사문헌

US6145007 A

US20020178396 A1

전체 청구항 수 : 총 5 항

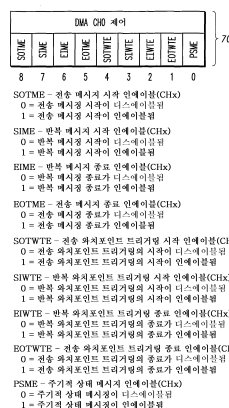
심사관 : 정현수

(54) DMA 디바이스를 위한 실시간 디버그 지원과 그 방법

(57) 요약

데이터 프로세싱 시스템(10)은 시스템에서 DMA(Directed Memory Access) 제어기 디바이스(16)에 특정한 한 개 이상의 디버그 메시지들을 선택적으로 생성하는 디버그 모듈(26)을 갖는다. 제어 레지스터(70)는 DMA 디버그 메시지들 중의 어떤 것이 제공되는지를 인에이블한다. 언제 소 루프 반복들이 시작하고 완료하는지를 포함하는 DMA 전송 동작의 시작과 종료에 제공된다. 각 DMA 전송에 대한 채널 요청과 그 요청의 실제 시작 사이의 시스템 지연을 나타내는 지연 정보도 또한 디버그 메시지에 포함될 수 있다. 디버그 메시지들 중의 하나는 제어 레지스터(80)의 제어하의 미리 결정된 DMA 채널의 주기적 상태를 제공한다. 디버그 메시지들 중의 적어도 하나는, 언제 전송이 시작하고 종료하는지를 나타내는 것과 같은, 와치포인트(watchpoint) 기능을 구현한다. 디버그 모듈은 시스템에서 중앙집중형이 되거나 또는 미리 결정된 시스템 유닛들 사이에 분산될 수 있다.

대표도 - 도9



## 특허청구의 범위

### 청구항 1

데이터 프로세싱 시스템으로서,

통신 버스;

상기 통신 버스와 결합된 DMA(direct memory access) 디바이스 - 상기 DMA 디바이스는 정보 채널들을 제어하고, 상기 정보 채널들 각각은 채널 전송(channel transfer)을 통해 상기 시스템의 소스에서 목적지로 정보를 전송함 - ; 및

상기 DMA 디바이스와 결합된 디버그 제어 회로 - 상기 디버그 제어 회로는 채널당 기준(per channel basis)으로 프로그래밍이 가능하여 상기 DMA 디바이스의 동작 파라미터들에 관련되는 디버그 메시지들을 제공함 - ;

를 포함하고,

상기 동작 파라미터들은,

- (1) DMA 전송의 시작 또는 종료;
- (2) 소 루프 반복(minor loop iteration) 전송이 언제 시작하는지와 언제 종료하는지;
- (3) 대 루프 반복(major loop iteration) 전송이 언제 시작하는지와 언제 종료하는지;
- (4) 정보 채널들의 상태 정보; 및
- (5) 각 DMA 채널 요청과 연관된 지연(latency)

중 적어도 하나를 포함하는 데이터 프로세싱 시스템.

### 청구항 2

제1항에 있어서, 상기 디버그 메시지들은 적어도 한 개의 미리 결정된 채널의 DMA 채널 전송 경계의 존재를 식별하고, 상기 디버그 메시지들 중 적어도 한 개는, 상기 DMA 디바이스가 정보 전송을 시작하라는 요청을 수신하는 것과 실제로 상기 정보를 전송하는 것 사이의 시스템 지연을 나타내는, 상기 채널 전송과 연관된 시간 지연을 나타내는 데이터 프로세싱 시스템.

### 청구항 3

제1항에 있어서, 상기 디버그 메시지들 각각은 각자의 채널의 사용 인자(utilization factor)(채널 상태 필드)를 나타내는 데이터 프로세싱 시스템.

### 청구항 4

제1항에 있어서, 상기 디버그 메시지들은 적어도 한 개의 미리 결정된 채널의 DMA 채널 전송 경계의 존재를 식별하고, 상기 디버그 제어 회로에 의해 제공되는 상기 디버그 메시지들 중의 하나는 상기 적어도 한 개의 미리 결정된 채널에 대해 채널 전송이 종료되었음을 나타내는 데이터 프로세싱 시스템.

### 청구항 5

제1항에 있어서, 상기 디버그 메시지들은 적어도 한 개의 미리 결정된 채널에 대해 적어도 한 개의 상태 파라미터를 주기적으로 제공하고, 상기 적어도 한 개의 상태 파라미터는, (1) 채널 우선순위, (2) 사용 인자, 및 (3) 상기 적어도 한 개의 미리 결정된 채널과 관련된 전송 오류가 이전에 발생했는지의 여부 중의 하나를 포함하는 데이터 프로세싱 시스템.

### 청구항 6

삭제

### 청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

청구항 45

삭제

청구항 46

삭제

청구항 47

삭제

청구항 48

삭제

## 명세서

### 기술분야

[0001] 본 발명은 데이터 프로세싱 시스템에 관한 것이고, 더 구체적으로는, 데이터 프로세싱 시스템의 특징들을 테스트하고 디버그(debug)하는 것에 관한 것이다.

### 배경기술

[0002] 데이터 프로세싱 시스템은 시스템을 위한 통신 고속도로로서 기능하는 한 개 이상의 버스들을 갖는다. 한 개 이상의 버스들은 클록 싸이클로 정의되는 시간 기간 동안 주소, 데이터, 및/또는 제어 정보를 전송하기 위해 기능한다. 데이터 프로세싱 시스템 내의 회로의 기능을 테스트하기 위해서, 및 소프트웨어 프로그램 디버깅으로서 알려진, 소프트웨어의 실행 기능을 검사하고 확인하기 위해서 다양한 기술들을 적용하는 것이 문헌들에 잘 문서화되어 있다. 테스트 방법들의 중요 구성요소는, 레지스터들의 내용과 시스템 동작의 결과와 같은, 데이터 프로세싱 시스템의 부분들에 대한 외부 가시성을 얻는 것이다. 시스템의 버스(들)에 의해 전송되고 있는 주소, 데이터, 및 제어 정보의 값들을 모니터하기 위해 데이터 프로세싱 시스템의 외부 테스트 및 디버그 장치가 사용된다.

[0003] 통상적 데이터 프로세싱 시스템은 DMA(direct memory access)로서 알려진 기술을 사용한다. DMA를 사용하여, CPU에 의한 최소의 관련으로 입/출력 디바이스들과 메모리 사이에 정보가 직접적으로 전송될 수 있다. 그러므로, DMA 제어기는 데이터 프로세싱 시스템의 소스로부터 목적지까지 정보 스트림을 전송하기 위해 기능한다.

[0004] SOC(system-on-chip) 설계는 통상적으로 실시간 기술을 사용하여 디버그 메시지를 생성할 수 있는 복수 개의 디

바이스들과 버스들로 구성된다. 예를 들어, IEEE ISTO5001로서 알려진 IEEE 표준, 또는 넥서스(Nexus) 디버그 표준은 복수 개의 온칩(on-chip) 디바이스들에 의한 실시간 디버그 메시지 생성을 지원하는 설정된 실시간 디버그 표준이다. 넥서스 디버그 표준은 외부 추적 재구성 툴에게 시스템 내의 미리 결정된 동작 조건을 식별하는 메커니즘을 명시한다. 그러나, 이 디버그 정보는 통상적으로, 시스템 메모리로 관독되거나 기록되는 정보의 모니터링 및, 최종 채택된 분지(branch)에서부터 실행된 다수의 순차적 명령들과 같이 실행되는 소프트웨어의 실행 흐름의 변경의 추적에 제한된다. 현재 디버그 프로세서는 DMA 동작에 대한 매우 제한된 양의 정보를 제공할 수 있다.

## 실시예

- [0017] 본 명세서에 사용된 것처럼, "버스"라는 용어는, 데이터, 주소, 제어, 또는 상태와 같은 한 개 이상의 다양한 유형의 정보를 전송하기 위해 사용될 수 있는 복수 개의 신호들이나 도선들을 지칭하기 위해 사용될 수 있다. "어설트(assert)"나 "니게이트(negate)"라는 용어들은 신호, 비트, 또는 유사한 장치를 그것의 논리적인 참 또는 거짓 상태로 만드는 것을 지칭할 때 사용된다. 논리적으로 참 상태가 로직 레벨 1일 때, 논리적으로 거짓 상태는 로직 레벨 0이다. 또한, 본 명세서에 기재된 메모리는, 예를 들어, ROM(read only memory), RAM(random access memory), SRAM(static RAM), 비휘발성 메모리(예를 들어, 플래시(Flash)), 및 MRAM 등과 같은, 임의의 유형의 메모리일 수 있다.
- [0018] 도 1은 실시간 DMA-고유한 디버그 기능을 제공하는 데이터 프로세싱 시스템(10)을, 블록도 형태로, 도시한다. 데이터 프로세싱 시스템(10)은 DMA 버스 마스터(master)들에 고유하고 중요 상태 파라미터들을 나타내는 실시간 디버그 메시지 유형들의 클래스를 허용한다. 그러므로, 데이터 프로세싱 시스템(10)은 DMA 유닛에 대한 실시간 메시지를 통해 디버그 정보의 향상된 소스를 제공한다.
- [0019] 일 형태에서, 모든 시스템 디바이스들 간의 통신을 위해 복수-도선 시스템 버스(12)가 제공된다. 프로세서(14)가 시스템 버스(12)에 양방향으로 접속된다. 본 명세서에 기재된 양방향 버스들이 복수 도선 버스이도록 해시 마크(hash mark)로 나타내지지만, 또한, 일 도선 버스 및/또는 복수 양방향 버스들도 구현될 수 있음을 쉽게 이해할 수 있어야 한다. 또한, DMA 제어기 디바이스(16), 메모리(18, 20), 주변기기(22, 24), 디버그 모듈(26), 및 버스 중재기(28)가 시스템 버스(12)에 양방향으로 접속된다. 디버그 모듈(26)은 그것의 출력에 디버그 정보를 제공하기 위한 디버그 포트(debug port)를 제공한다. DMA 제어기 디바이스(16)와 프로세서(14)의 각각은 디버그 모듈(26)에 복수 비트 상태 신호를 제공하기 위한 출력을 갖는다. 프로세서(14)와 DMA 제어기 디바이스(16) 각각은 버스 중재자(28)와 결합된 복수 비트 양방향 버스를 갖는다. 주변기기(22)의 출력은 DMA\_REQ로 레이블된 DMA 요청 신호를 제공하기 위해 DMA 제어기 디바이스(16)에 접속된다. 유사하게, DMA\_REQ로 레이블된 DMA 요청 신호를 제공하기 위해 DMA 제어기 디바이스(16)에 주변기기(24)의 출력이 접속된다. 본 명세서에 기재된 시스템 유닛들이, 시스템 버스(12)에 직접 접속된 것으로서, 도 1에 도시되지만, 드라이버, 버퍼 등과 같은, 결합 회로(도시 생략)가 시스템 버스(12)와 도시된 시스템 유닛들 간에 존재할 수 있음을 이해해야 할 것이다.
- [0020] 동작에서, 데이터 프로세싱 시스템(10)은 프로세서(14)를 사용하여 미리 결정된 명령 세트를 실행한다. 정보는 메모리(18) 또는 메모리(20) 중의 하나에 저장된다. 단지 한 개의 메모리 또는 임의의 다른 수의 메모리가 구현되어 시스템 버스(12)에 접속될 수 있음을 이해해야 할 것이다. DMA 제어기 디바이스(16)는 데이터 프로세싱 시스템(10) 내에 소스에서 목적지로 정보 스트림의 전송을 제어하기 위해 기능한다. 데이터 프로세싱 시스템(10) 내의 정보는 소스와 목적지 사이에 연관된 스트림인 채널 내에서 통신된다. 주변기기(22) 또는 주변기기(24)와 같은 데이터 프로세싱 시스템(10) 내의 주변기기는 다양한 회로 디바이스들 중의 임의의 것일 수 있지만, 통상적으로 시스템 버스(12)의 제어를 위해 버스 중재자(28)를 요청하지 않고 시스템 버스(12)에 슬레이브(slaves)로서 기능할 것이다. 주변기기가 무슨 유형의 디바이스인지에 무관하게, 주변기기는 시스템 메모리 매핑의 부분을 형성한다. 버스 중재자(28)는 시스템 버스(12)의 제어를 요청하는 데이터 프로세싱 시스템(10) 내의 임의의 디바이스들 간의 시스템 버스(12)의 제어를 중재하기 위해 기능한다. 다수의 종래 중재 알고리즘 중의 임의의 것이 데이터 프로세싱 시스템(10)을 구현하기 위해 사용될 수 있다. 디버그 모듈(26)은 아래 기재된 것처럼 DMA 제어기 디바이스(16)와 연관된 한 개 이상의 실시간 디버그 메시지를 제공하기 위해 기능한다. 환언하면, 디버그 모듈(26)은 DMA 제어기 디바이스(16)와 프로세서(14) 모두로부터 상태 신호를 수신하여 DMA 제어기 디바이스(16)를 모니터링하고, 디버그 포트에 출력하기 위해 아래 기재된 바와 같이 실시간 디버그 메시지를 생성한다.
- [0021] 도 2에는, 도 1의 데이터 프로세싱 시스템(10)의 구성의 다른 실시예가 도시된다. 시스템 버스(30)는 DMA 제어

기(44)와 시스템 유닛(40)과 시스템 유닛(42)과 같은 미리 결정된 수의 다른 시스템 유닛에 양방향으로 접속된다. 시스템 유닛(40)과 시스템 유닛(42)은 임의의 유형의 버스 마스터 또는 버스 슬레이브일 수 있다. 시스템 유닛(40), DMA 제어기(44), 및 시스템 유닛(42)의 각각은 디버그 모듈(32), 디버그 모듈(34), 및 디버그 모듈(36)과 양방향으로 결합된다. 디버그 모듈(32), 디버그 모듈(34), 및 디버그 모듈(36)의 각각의 출력은 디버그 포트 로직(38)에 접속된다. 디버그 포트 로직(38)은 실시간 디버그 메시지를 제공하기 위해 디버그 포트를 제공한다. 도 2의 점선으로 나타난 것처럼, 시스템 버스(30)와 디버그 포트 로직(38) 사이에 임의의 수의 디버그 모듈과 시스템 유닛이 결합될 수 있다.

[0022] 다른 형태에서, 디버그 모듈의 기능은, DMA 제어기(44)를 포함하는, 시스템 버스(30)와 결합되는 각 시스템 유닛에 대해 디버그 모듈을 할당하기 위한 방식으로 분리된다. 디버그 포트 로직(38)은 디버그 모듈(32), 디버그 모듈(34), 및 디버그 모듈(36)의 각각으로부터 메시지를 수신하고, 디버그 포트 로직에서 메시지를 제공한다. 일 형태에서, 메시지들이 다양한 디버그 모듈로부터 수신되는 순서로 출력되는 디버그 메시지들을, 시간 순차적으로, 제공하기 위해 디버그 포트 로직(38)의 로직이 구성된다. 그러나, 디버그 출력들의 다른 순서를 구현하기 위해 다른 로직 구성이 사용될 수도 있다.

[0023] 도 3은, 도 1이나 도 2의 시스템 실시예 내에, 채널 X로서 지시된, 일 채널 상의 전송 동작을 나타내는 타이밍도이다. 채널 X가 소스와 목적지와 연관된 시스템 실시예 내의 미리 결정된 설명적 데이터 스트림임을 가정한다. 소스에서 목적지로 정보 스트림을 전송하기 위해 복수 채널을 지원하기 위해 DMA 제어기 디바이스(16)와 DMA 제어기(44)의 각각이 구성된다. 소스나 목적지 디바이스로부터, DMA\_REQ와 같은 DMA 요청 신호를 수신시에 채널 내의 전송이 발생된다. 도 3의 예에서, 3개의 개별적 요청이 도시되었다. 각 요청은, 제1 요청과 연관된 지연(50), 제2 요청과 연관된 지연(52), 및 제3 요청과 연관된 지연(54)과 같은 연관된 지연을 갖는다. 버스 중재자(28)에 의해 결정되는 것처럼, DMA 제어기 디바이스(16)의 현재 우선순위, 및 DMA 제어기 디바이스(16)의 동작에 의해 결정되는 것처럼, 채널 X의 우선순위와 같은, 여러 시스템 요인들로 인해 각 지연의 길이가 달라진다. 이 지연이 시스템의 외부이고 시스템을 인터럽트하지 않는 것이 바람직한 시스템의 일 특성이다.

[0024] 정보 스트림의 전송은 요청에 따라 특정 수의 데이터 바이트들이 전송될 때까지 계속된다. 도 3에 도시된 형태에서, DMA 제어기 디바이스(16)와 DMA 제어기(44)의 각각이 요청에 따라 N 바이트의 정보를 전송하고, 여기서, N은 정수임을 가정한다. 예를 들어, 전송(60, 62, 및 64)은 3개의 분리된 순차적 채널 요청에 응답하여 발생한다. 전송(62)과 연관된 인터럽트 기간(66)과 같이, 전송이 인터럽트될 수 있음을 주목해야 한다. 인터럽트 기간(66)은, 버스 중재자(28)에 의해 결정되는 것처럼, 버스 마스터십(mastership)의 변화와 같이, 다수의 시스템 요인들로부터 결과가 될 수 있다. 각 N 바이트의 전송은 "반복(iteration)" 또는 "소 루프 반복(minor loop iteration)"으로서 지칭된다. "대 루프 반복(major loop iteration)" 내에서 다수의 소 루프 반복이 실행된다. '반복'이라는 용어의 사용이 동일한 소 루프 또는 대 루프 전송의 반복을 내포하거나 또는 요구하지 않음을 주목해야 한다. 그보다는, 이 컨텍스트의 반복은, 도 3의 채널 X와 같은 임의의 채널에 대해, 전송 동작이 일반적이므로, 복수의 연속적 전송이 발생한다는 사실을 일컫는다. 도 3의 예에서, 대 루프 반복(65) 내에, 3개의 소 루프 반복인 소 루프 반복(59, 61, 및 63)이 존재한다. 그러므로, 설명된 채널 동작은, 2개의 레벨의 네스팅(nesting)이 소 루프와 대 루프인 2-깊이 네스팅(2-deep nested) 전송 동작인 것으로 간주될 수 있다. 채널 요청 신호의 점선에 의해 나타난 것처럼 임의의 수의 소 루프 반복이 존재할 수 있음을 이해해야 한다. 소 루프 반복의 완료 후에, 계류 요청을 갖는 더 높은 우선순위 채널로 시스템 버스(12)의 제어가 전송될 수 있다. 다른 경우, 소 루프 반복을 시작하기 위해 다른 채널을 트리거링(trigger)하기 위해 트리거링 이벤트가 발생할 수 있다. 또한, 대 루프 반복의 종료에 시스템 버스(12)의 제어가 전송될 수 있다.

[0025] 전체 시스템 동작을 최적화하기 위해, 시스템이 실행하는 DMA 동작의 다양한 양태들의 외부 가시성은 매우 중요하다. 그러므로, 본 명세서에 DMA 동작과 관련된 실시간 추적 메시지의 세트가 제공된다.

[0026] 도 4는, 도 3의 대 루프 반복(65)의 제1 전송과 같은, 제1 전송의 시작을 식별하는 제1 실시간 DMA 제어기 메시지를 도시한다. 제1 전송 메시지는 선택적인 2개의 필드들을 갖춘 5개의 필드들을 갖는 것으로서 도시된다. 제1 필드는, 메시지의 유형이나 기능을 식별하는 TCODE(type code) 필드로서 식별된다. 단지 예로서, 110111의 6 비트 TCODE 값이 메시지를 "제1 전송의 시작" 메시지로 식별하기 위해 선택된다. 제2 필드는, 일 형태에서, 4 내지 8 비트 길이이고, 무슨 채널 또는 무슨 유형의 채널에 메시지가 연관되었는지의 정보를 알려주는 채널 ID 필드이다. 제3 필드는 채널 상태 필드이고, 단지 예로서, 1 내지 8비트 길이일 수 있다. 채널 상태의 예들로는, 제1 전송의 시작에서 채널의 우선순위, 채널의 사용 인자(예를 들어, 사용 퍼센트), 채널에 대한 전송 오류가 이전에 발생했는지의 여부, 및 다수의 다른 상태나 정보 세부사항을 포함한다. 선택적인 제4 필드는, 채널에 얼마나 많은 정보가 전송되는지를 나타내는 전송 카운트이다. 이 필드는 채널과 연관된 각 소



루프 반복에 대한 전송 카운트 정보를 포함할 수 있고, 및/또는 채널과 연관된 대 루프 반복에 대한 전송 카운트 정보를 포함할 수 있다. 단지 예로서, 전송 카운트 필드는 8 내지 64 비트 길이일 수 있지만, 다른 비트 길이가 사용될 수도 있다. 또한 선택적인 제5 필드는 전송 서비스 지연이다. 전송 서비스 지연은, 전송 동작을 시작하기 위해 초기 채널 요청으로부터 얼마나 시간이 소요되는지를 식별한다. 환언하면, 이 필드의 값은, 지연(50)과 같이, 제1 전송과 연관된 인터럽트 지연이다. 이 필드의 값은, 지연(50)과 연관된 시간 기간을 카운트하기 위해 디버그 모듈이나 다른 곳에 있을 수 있는 카운터(도시 생략)를 사용하여 측정될 수 있다. 일 형태에서, 전송 서비스 지연 필드는 8 내지 64 비트 길이일 수 있지만, 다른 비트 길이가 사용될 수도 있다.

[0027] 도 5에는, 도 3의 소 루프 반복(59, 61, 및 63) 중의 임의의 것과 같은, 소 루프 반복의 시작을 식별하는 제2 실시간 DMA 제어기 메시지가 도시된다. 이 제2 전송 메시지는 선택적인 2개의 필드들을 갖춘 5개의 필드들을 갖는 것으로서 도시된다. 제1 필드는, 메시지를 '소 루프 반복의 시작' 메시지로 식별하는 TCODE(type code) 필드이다. 단지 설명 목적으로, TCODE는 111000으로서 도시되지만, 다른 값과 비트 길이가 사용될 수도 있다. 제2 필드는 소 루프 반복의 시작과 연관된 채널을 식별하는 채널 ID 필드이다. 제3 필드는, 제1 메시지에 대해 위에 기재된 상태 정보와 같은, 채널의 상태 파라미터나 파라미터들을 식별하는 채널 상태 필드이다. 제4 필드는, 제1 메시지의 전송 카운트 필드에 대해 위에 기재된 바와 같이 채널이 얼마나 많은 정보를 전송할지를 식별하는 전송 카운트이다. 제5 필드는, 지연(50, 52, 및 54)과 같은 지연을 식별하는 전송 서비스 지연 필드이다. 또한, 각 필드에 제공되는 비트 길이는 단지 예일 뿐이다.

[0028] 도 6에는, 도 3의 소 루프 반복(59, 61, 및 63) 중의 임의의 것과 같은, 소 루프 반복의 종료를 식별하는 제3 실시간 DMA 제어기 메시지가 도시된다. 이 제3 전송 메시지는 선택적인 2개의 필드들을 갖춘 5개의 필드들을 갖는 것으로서 도시된다. 제1 필드는, 메시지를 '소 루프 반복의 종료' 메시지로 식별하는 TCODE 필드이다. 단지 예로서, TCODE는 111001로서 도시되지만, 다른 값이나 비트 길이가 사용될 수도 있다. 제2 필드는 소 루프 반복의 종료와 연관된 채널을 식별하는 채널 ID 필드이다. 제3 필드는, 제1 메시지에 대해 상술된 상태 정보와 같은, 채널의 상태 파라미터나 파라미터들을 식별하는 채널 상태 필드이다. 제4 필드는, 어느 소 루프 반복이 현재 시스템 버스 상에서 전송되고 있는지를 식별하는 반복 카운트이다. 제5 필드는, 소 루프 반복이 얼마나 오래 지속되었는지의 카운트 값인 반복 간격이다. 시스템의 임의의 장소에 위치되는 카운터(도시 생략)가 사용될 수 있지만, 카운터는 디버그 모듈 내에 있거나, 또는 거기에 가까이 위치되는 것이 바람직하다. 또한, 각 필드에 제공되는 비트 길이는 단지 예일 뿐이다. 또한, 전체 소 루프가 전송 오류로 인해 정상적으로 완전히 완료되지 않아도 제3 전송 메시지가 생성되도록, 디버그 모듈(26)이 구성될 수 있다.

[0029] 도 7에는, 도 3의 대 루프 반복(65)과 같은, 대 루프 반복의 완료를 식별하는 제4 실시간 DMA 제어기 메시지가 도시된다. 이 제4 전송 메시지는 선택적인 2개의 필드들을 갖춘 5개의 필드들을 갖는 것으로서 도시된다. 제1 필드는, 메시지를 '대 루프 반복의 종료' 메시지로 식별하는 TCODE 필드이다. 단지 예로서, TCODE는 111010으로서 도시되지만, 다른 값이나 비트 길이가 사용될 수도 있다. 제2 필드는 대 루프 반복의 종료와 연관된 채널을 식별하는 채널 ID 필드이다. 제3 필드는, 제1 메시지에 대해 상술된 상태 정보와 같은, 채널의 상태 파라미터나 파라미터들을 식별하는 채널 상태 필드이다. 제4 필드는, 제1 메시지의 전송 카운트 필드에 대해 상술된 바와 같이 채널이 얼마나 많은 정보를 전송했는지를 식별하는 전송 카운트이다. 제5 필드는 대 루프 반복이 얼마나 오랫동안 지속되었는지의 카운트 값인 전송 간격 필드이다. 시스템의 임의의 곳에 위치되는 카운터(도시 생략)가 사용될 수 있지만, 카운터가 디버그 모듈(26) 내에 또는 그 가까이 위치되는 것이 바람직하다. 또한, 각 필드에 제공되는 비트 길이는 단지 예일 뿐이다. 또한, 전체 대 루프가 전송 오류로 인해 정상적으로 완전히 완료되지 않아도 제4 전송 메시지가 생성될 수 있도록 디버그 모듈(26)이 구성될 수 있다.

[0030] 도 8에는, 주기적으로 식별된 채널의 상태를 제공하는 제5 실시간 DMA 제어기 메시지가 도시된다. 이 주기적 메시지는 식별된 채널의 한 개 이상의 상태 기준을 제공하기 위해 기능한다. 이 제5 전송 메시지는 선택적인 필드들 중의 하나를 갖춘 4개의 필드들을 갖는 것으로서 도시된다. 제1 필드는 메시지를 '주기적인 상태' 메시지로 식별하는 TCODE 필드이다. 단지 설명 목적으로, TCODE는 111011로서 도시되지만, 다른 값들과 비트 길이도 사용될 수 있다. 제2 필드는, 제공되는 상태 정보와 연관되는 채널을 식별하는 채널 ID 필드이다. 제3 필드는, 제1 메시지의 논의와 연결하여 상술된 상태 정보와 같은, 채널의 상태 파라미터나 파라미터들을 식별하는 채널 상태 필드이다. 제4 필드는, 상태가 관련된 시점에서 현재 전송이 얼마나 멀리 진행했는지를 식별하는 현재 전송 카운트이다. 또한, 각 필드에 제공된 비트 길이는 단지 예일 뿐이다. 주기적 상태 메시지는 시스템에서 채널들의 상대적 진행을 결정하기 위해서 그리고 DMA 채널들의 채널 우선순위와 버스 마스터 우선순위의 수정을 허용하기 위해 사용될 수 있다. 주기적 상태 메시지가 생성되기 위해서, 그 기능이 아래 기재되는 레지스터의 프로그램가능한 제어에 의해 인에이블(enable)된다. 그 기능이 인에이블될 때, 주기적 상태 메시지



가 제공되는 타이밍은 구현에 특정하다. 주기적 상태 메시지는, 미리 결정되거나 미리 정의된 수의 시스템 클럭이나 버스 사이클이 발생할 때마다 생성될 수 있다. 다른 경우, 주기적 상태 메시지는 시스템의 특정한 하드웨어 유선에 의해 결정되는 것처럼 고정된 주기에서 생성될 수 있다.

[0031] 도 9는 한 개의 DMA 채널을 위한 제어 레지스터(70)이다. 단지 설명 형태로서, 제어 레지스터(70)는 5개의 DMA 실시간 메시지를 인에이블 또는 디스에이블하는 5개의 비트들, 및 와치포인트(watchpoint)의 트리거의 생성을 인에이블 또는 디스에이블하는 4개의 비트들을 갖는다. 일 형태에서, 각각이 제어 레지스터(70)와 유사한, 다양한 DMA 채널들을 위한 제어 레지스터들이 디버그 모듈(26) 내에 제공되어 위치된다. 제어 레지스터(70)는 채널 "CH0"로서 나타낸 DMA 채널을 위한 것이다. 채널 x로서 표현된, CHx, 임의의 채널에 대한 제어 비트 기능의 설명이 제어 레지스터(70)의 바로 아래에 있다. 제어 레지스터(70)의 제1 비트, 비트 0은 주기적 상태 메시지 기능을 인에이블한다. 비트 0이 어설트된 로직 1 값을 가질 때, 디버그 모듈(26)은 도 8에 도시된 것처럼 형식을 갖는 주기적 상태 메시지를 생성할 것이다. 유사한 방식으로, 제어 레지스터(70)의 5, 6, 7, 및 8 비트는 도 7, 6, 5, 및 4 각각의 메시지들을 인에이블하고 디스에이블한다. 도 4 내지 도 8의 DMA 메시지들에 추가하여, 데이터 프로세싱 시스템(10)의 외부 분석을 위해 와치포인트의 트리거링을 인에이블하기 위해 제어 레지스터(70)가 기능한다. 본 명세서에 사용되는 것처럼, "와치포인트"라는 용어는 미리 결정된 간격 시스템 이벤트 그리고 언제 그런 간격 시스템 이벤트가 발생하는지의 신호나 지시를 모니터링하는 것을 일컫는다. 제어 레지스터(70)의 1, 2, 3, 및 4 비트는 와치포인트 기능들을 인에이블하는 것과 연관된다. 이들 와치포인트들이 인에이블될 때, 그 이벤트가 언제 발생했는지를 지시하는 것에 반하여, 디버그 모듈(26)에 의해 디버그 포트, 연관된 기능이 발생했다는 지시가 제공된다. 예를 들어, SOTWTE 비트는 연관된 채널에서의 전송의 시작을 결정하고, 그 전송이 발생했으면 디버그 포트에 신호를 어설트하거나 또는 메시지를 제공하는 기능을 인에이블한다. 도 9에 제공된 다른 와치포인트 비트들은, 반복 시작(소 루프나 대 루프가 정의될 수 있음)이 발생했다는 것(비트 3), 반복의 종료(소 루프 또는 대 루프)가 발생했다는 것(비트 2), 및 전송의 종료(소 루프 또는 대 루프)가 발생했다는 것(비트 1)을 지시하기 위해 기능한다. 와치포인트 기능들의 선택적 사용이 다른 디버그 기능들을 인에이블하기 위해 사용될 수 있다. 예를 들어, "전송의 종료" 와치포인트는, 시스템 유닛(40), 시스템 유닛(42), 프로세서(14), 또는 데이터 프로세싱 시스템(10) 내의 임의의 다른 원하는 기능과 연관된 다른 디버그 기능을 트리거하기 위해 데이터 프로세싱 시스템(10) 내에서 사용될 수 있다. 제어 레지스터(70)의 기능은 원하는 애플리케이션 요구사항에 따라 도 9에 도시된 것으로부터 확장 또는 축소될 수 있다. 또한, 제어 레지스터(70)는 디버그 모듈(26) 내보다는 데이터 프로세싱 시스템(10)의 다른 영역들에 위치될 수 있다.

[0032] 도 10에서는, DMA 제어기 디바이스(16)의 원하는 수의 채널들의 각각에 대해 필드들을 갖는 주기적 상태 메시징 제어 레지스터(80)가 도시된다. 설명된 형태에서, 4개의 채널들이 제공되고, 32 비트가 할당되지만, 임의의 수의 채널과 비트가 사용될 수 있다. CH0으로 표현된, 제1 채널 필드에서, 거기에 저장된 디지털 값은 디버그 모듈(26)에 의해 주기적 상태 메시지가 발생하는 비율을 제어한다. 그러므로, 주기적 상태 메시징 제어 레지스터(80)는 생성될 각 채널에 대한 프로그램가능하고 독립적인 보고 비율을 허용하기 위해 기능한다.

[0033] 지금까지, DMA 동작 파라미터들에 특정한 메시지들의 형태로 정보를 제공하는 실시간 디버그 지원 방법과 구조가 제공되었음을 이해해야 할 것이다. DMA 동작 파라미터 정보 내에는, 언제 DMA 전송이 시작하고 종료하는지, 언제 각 소 루프 반복 전송이 시작하고 종료하는지, 언제 대 루프 반복이 시작하고 종료하는지, 미리 결정된 DMA 채널의 주기적 상태, 및 각 DMA 채널 요청과 얼마나 많은 지연이 연관되는지를 아는 것이 포함된다. DMA 제어기 디바이스(16) 내의 기존 신호들에 기초하여 본 명세서에 기재된 DMA 디버그 메시지들이 실시간으로 생성되어, DMA 제어기 디바이스(16) 또는 프로세서(14)의 동작이 인터럽트되지 않는다. 본 명세서에 기재된 DMA 메시지들에 포함된 정보에 외부 액세스를 가짐으로써, 향상된 레벨의 SOC(system-on-chip) 디버그, 확인, 및 수행이 쉽게 달성될 수 있다.

[0034] 일 형태에서, 통신 버스 및 통신 버스와 결합된 DMA 디바이스를 갖는 시스템과 그것의 동작의 방법이 제공되었다. DMA 디바이스는 정보 채널들을 제어하고, 정보 채널들의 각 채널은 채널 전송을 통해 시스템의 소스에서 목적지로 정보를 전송한다. 디버그 제어 회로는 DMA 디바이스와 결합되고, 디버그 제어 회로는 적어도 한 개의 미리 결정된 채널에 대한 DMA 채널 전송 경계의 존재를 식별하는 디버그 메시지들을 제공한다. 또한, 디버그 제어 회로는, 디버그 메시지가 채널 전송 경계의 존재를 식별하는 DMA 디바이스에 의해 정보 채널들 중의 어떤 것이 제어되는지를 선택하는 프로그램가능한 제어 메커니즘을 포함한다. 디버그 제어 회로에 의해 제공되는 디버그 메시지들 중의 하나는 채널 전송이 적어도 한 개의 미리 결정된 채널에 대해 시작했음을 나타낸다. 또한, 디버그 메시지들 중의 다른 하나는 적어도 한 개의 미리 결정된 채널의 상태 파라미터를 나타낸다. 일 형태에서, 상태 파라미터는 다음 것들 중의 하나이다: (1) 적어도 한 개의 미리 결정된 채널의 채널 우선순위, (2) 적

어도 한 개의 미리 결정된 채널의 사용 인자, 및 (3) 적어도 한 개의 미리 결정된 채널과 연결하여 전송 오류가 이전에 발생했는지의 여부. 다른 형태에서, 또한 디버그 메시지들 중의 하나는 DMA 디바이스에서 정보 전송을 시작하기 위한 요청의 수신과 실제로 그 정보의 전송 사이의 시스템 지연을 나타내는 채널 전송과 연관되는 시간 지연을 나타낸다. 다른 형태에서, 디버그 제어 회로에 의해 제공된 디버그 메시지들 중의 하나는 채널 전송이 적어도 한 개의 미리 결정된 채널에 대해 종료했음을 나타낸다. 다른 형태에서, 디버그 제어 회로에 의해 제공된 디버그 메시지들 중의 하나는 적어도 한 개의 미리 결정된 채널의 복수 개의 소 루프 반복들의 각각이 시작했음을 나타낸다. 다른 형태에서, 디버그 제어 회로에 의해 제공되는 디버그 메시지들 중의 하나는 적어도 한 개의 미리 결정된 채널의 주기적 상태를 나타낸다. 다른 경우, 시스템은 복수 개의 시스템 유닛들을 갖도록 구성될 수 있고, 복수 개의 시스템 유닛들의 각각은 통신 버스와 결합될 수 있다. 다른 형태에서, 또한, 디버그 제어 회로는 디버그 메시지를 제공하는 복수 개의 디버그 모듈들을 포함하고, 복수 개의 디버그 모듈들의 각각은 복수 개의 시스템 유닛들 중의 미리 결정된 하나와 결합된다. 디버그 포트 로직은 디버그 포트에 디버그 메시지를 제공하는 복수 개의 디버그 모듈들과 결합된다.

[0035] 다른 형태에서, 통신 버스와 통신 버스와 결합된 DMA 디바이스를 갖는 시스템과 그것의 동작 방법이 제공되어왔다. DMA 디바이스는 정보 채널들을 제어하고, 정보 채널들의 각 채널은 채널 전송을 통해 시스템의 소스에서 목적지로 정보를 전송한다. 디버그 제어 회로는 DMA 디바이스와 결합되고, 디버그 제어 회로는 적어도 한 개의 미리 결정된 채널에 대한 적어도 한 개의 상태 파라미터를 주기적으로 제공하는 디버그 메시지를 제공한다. 일 형태에서, 적어도 한 개의 상태 파라미터는 다음 것들 중에 적어도 하나이다: (1) 적어도 한 개의 미리 결정된 채널의 채널 우선순위, (2) 적어도 한 개의 미리 결정된 채널의 사용 인자, 및 (3) 적어도 한 개의 미리 결정된 채널과 연결하여 전송 오류가 이전에 발생했는지의 여부. 또한, 디버그 제어 회로는 채널 전송이 시작했음을 나타내는 디버그 메시지를 제공한다. 또한, 디버그 제어 회로는 채널 전송이 종료했음을 나타내는 디버그 메시지를 제공한다. 디버그 제어 회로는 정보 채널들의 각각에 대한 디버그 메시지의 선택적 생성을 인에이블하기 위해 프로그래밍이 가능하다. 또한, 디버그 제어 회로는 적어도 한 개의 미리 결정된 채널의 미리 결정된 와치포인트 일 조건인 와치포인트 지시자를 생성하고, 그 미리 결정된 와치포인트 조건은 DMA 디바이스의 동작에 기초한 와치포인트 조건이다. 또한, 디버그 제어 회로는 복수 개의 와치포인트 지시자를 생성하고, 이들 각각은 정보 채널들 중의 구별하는 하나의 와치포인트 조건을 나타낸다. 적어도 한 개의 미리 결정된 채널에 대한 적어도 한 개의 상태 파라미터를 주기적으로 제공하는 디버그 메시지의 제공을 인에이블하고 디스에이블하는 제어 신호를 제어 레지스터에 저장한다. DMA 디바이스는, 대 루프 반복을 형성하는 복수 개의 소 루프 반복을 포함하는 동일 채널 내에 네스트된 정보 전송을 구성한다. 디버그 메시지 각각은 메시지 유형 필드, 채널 ID 필드, 및 상태 정보 필드를 포함하는 미리 결정된 비트 필드들을 갖는 복수 비트 메시지를 포함한다.

[0036] 다른 형태에서, 통신 버스와 통신 버스와 결합된 DMA 디바이스를 갖는 시스템과 그 동작 방법이 제공된다. DMA 디바이스는 복수 개의 정보 채널들을 제어하고, 정보 채널들의 각 채널은 채널 전송을 통해 시스템에서 소스에서 목적지로 정보를 전송한다. 디버그 제어 회로는 DMA 디바이스와 결합된다. 디버그 제어 회로는 채널에 기초하여 프로그래밍이 가능하여 DMA 디바이스의 동작 파라미터들과 관련된 디버그 메시지를 선택적으로 제공한다. DMA 디바이스의 동작 파라미터들은 전송 경계가 발생했는지의 여부와 주기적 상태 정보 중의 적어도 하나에 관한 정보를 포함한다. 또한, 디버그 제어 회로는, DMA 디바이스에 의해 채널 전송 요청이 수신된 후에 채널 전송을 시작하는 DMA 디바이스의 시스템 지연과 관련된 지연 정보를 포함하는 적어도 한 개의 디버그 메시지를 제공한다.

[0037] 개시된 실시예들에 다양한 수정이 만들어질 수 있음이 명백해야 할 것이다. 예를 들어, 다양한 유형들의 프로세싱 기능들을 수행하는 다양한 상이한 유형의 데이터 프로세싱 회로들 중의 임의의 것으로서 프로세스(14)가 구현될 수 있다. 임의의 유형의 주변 디바이스는, 시스템 버스를 통해, 부프로세서들을 포함하는, 프로세서(14)와 결합될 수 있다. 이 시스템은 다양한 상이한 비트 크기들 중의 임의의 것으로 구현될 수 있다. 임의의 유형의 저장 디바이스가 설명된 저장소, 레지스터, 및 메모리를 위해 사용될 수 있다. 메시지의 필드들의 순서는 임의의 방식으로 변경될 수 있다. 제어 레지스터의 제어 비트들의 순서는 변경될 수 있다.

[0038] 특정 실시예들에 대해 이점, 다른 장점, 및 문제의 해결책이 상술되었다. 그러나, 이점, 장점, 문제의 해결책, 그리고 임의의 이점, 장점, 또는 해결책이 발생하거나 더 공포되도록 하는 임의의 요소(들)가 일부 또는 전체의 청구항들의 중요한, 필수의, 또는 기본 특징이나 요소로서 해석되어서는 안 된다. 본 명세서에 사용된 바와 같이, "포함한다(comprises)", "포함하는(comprising)"이라는 용어들, 또는 그것의 임의의 다른 변형은 비배타적

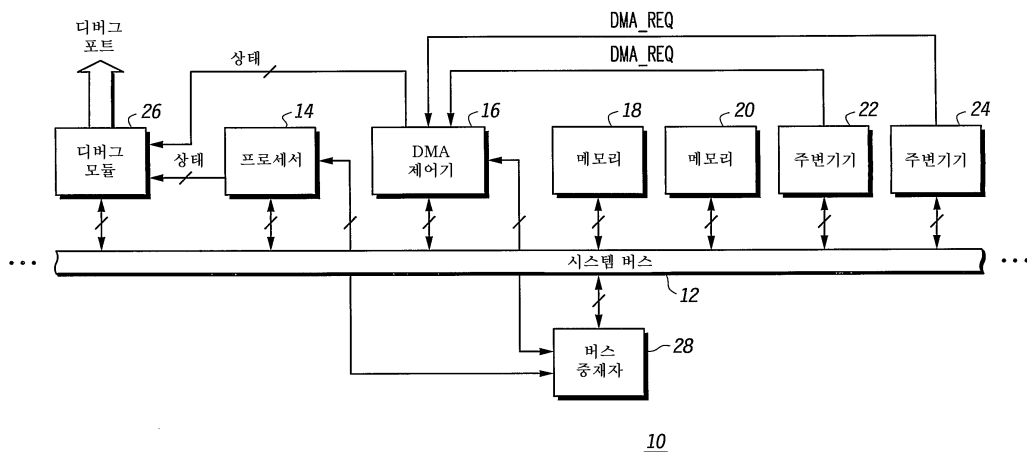
포함관계를 망라하려고 의도되어서, 요소들의 리스트를 포함하는 프로세스, 방법, 제품, 또는 장치가 단지 이들 요소들만을 포함하는 것뿐만이 아니라 그런 프로세스, 방법, 제품, 또는 장치에 명백히 리스트되거나 고유하지 않은 다른 요소들도 포함할 수 있다. 본 명세서에 기재된 바와 같이 한 또는 하나의 (a or an)라는 용어들은 한 개 또는 한 개 이상으로서 정의된다. 본 명세서에 사용되는 바와 같이, 복수(plurality)라는 용어는 2개 또는 2개 이상으로서 정의된다. 본 명세서에 사용되는 바와 같이, 다른 하나(another)라는 용어는 적어도 제2 또는 그 이상으로서 정의된다. 본 명세서에 사용되는 바와 같이, 포함하는(including) 및/또는 갖는(having)이라는 용어들은 포함하는(comprising)(즉, 개방 언어(open language))으로서 정의된다. 본 명세서에 사용된 바와 같이, 결합된(coupled)이라는 용어는, 직접적일 필요가 없고 기계적일 필요도 없지만, 접속된(connected)으로서 정의된다.

## 도면의 간단한 설명

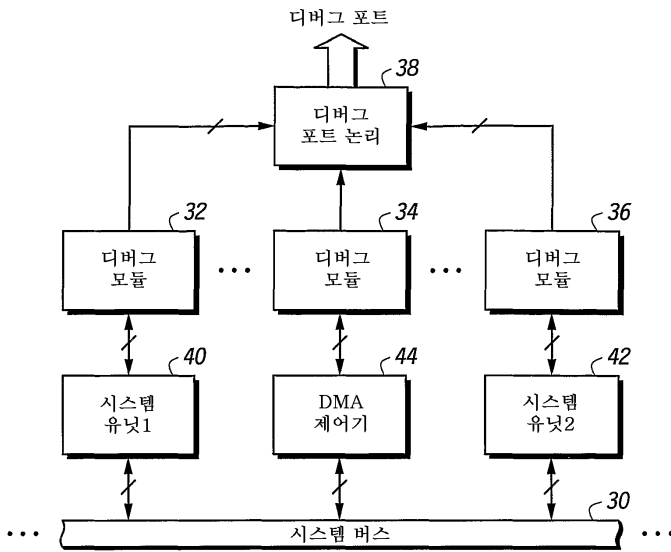
- [0005] 본 발명은 예를 통해 설명되고, 유사한 참조 부호들이 유사한 소자들을 나타내는, 동반하는 도면들에 의해 제한되지는 않는다.
- [0006] 도 1은 본 발명의 일 실시예에 따른 데이터 프로세싱 시스템을 블록도 형태로 도시한다.
- [0007] 도 2는 본 발명에 따른 다른 데이터 프로세싱 시스템을 블록도 형태로 도시한다.
- [0008] 도 3은 채널의 정보 전송을 시간 함수로서 타이밍도 형태로 도시한다.
- [0009] 도 4는 제1 전송 메시지의 형식을 다이어그램 형태로 도시한다.
- [0010] 도 5는 제2 전송 메시지의 형식을 다이어그램 형태로 도시한다.
- [0011] 도 6는 제3 전송 메시지의 형식을 다이어그램 형태로 도시한다.
- [0012] 도 7는 제4 전송 메시지의 형식을 다이어그램 형태로 도시한다.
- [0013] 도 8는 제5 전송 메시지의 형식을 다이어그램 형태로 도시한다.
- [0014] 도 9는 DMA 메시지를 위한 제어 레지스터를 레이아웃 형태로 도시한다.
- [0015] 도 10은 주기적 상태 메시지를 위한 제어 레지스터를 레이아웃 형태로 도시한다.
- [0016] 당업자라면, 도면의 소자들이 단순성과 명료성을 위해 도시되었고 반드시 스케일링하기 위해 그려진 것은 아님을 이해할 것이다. 예를 들어, 도면들의 일부 소자들의 크기는 본 발명의 실시예들에 대한 이해를 돕기 위해 다른 소자들과 비교하여 과장될 수 있다.

## 도면

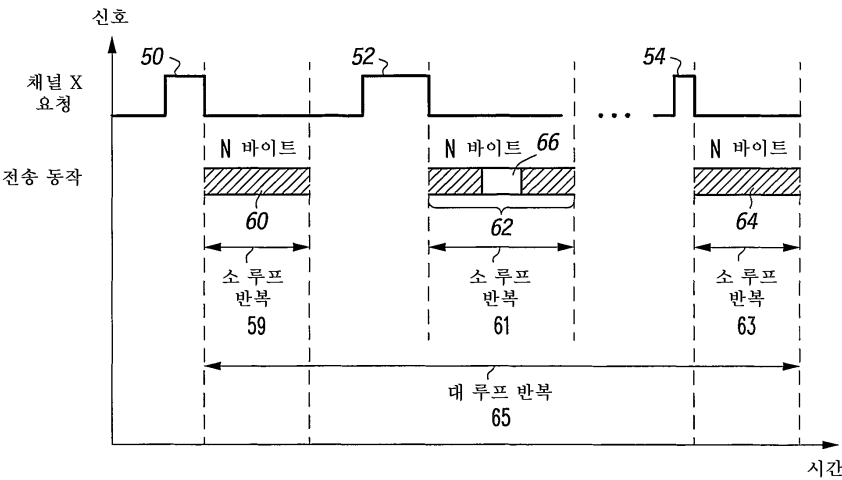
### 도면1



도면2



도면3



도면4

(8-64 비트)	(8-64 비트)	(1-8 비트)	(4-8 비트)	(6 비트)
전송 서비스 지연(선택적)	전송 카운트(선택적)	채널 상태	채널 ID	TCODE (110111)

도면5

(8-64 비트 )	(8-64 비트 )	(1-8 비트 )	(4-8 비트 )	(6 비트 )
전송 서비스 지연(선택적)	전송 카운트(선택적)	채널 상태	채널 ID	TCODE (111000)

도면6

(8-64 비트 )	(8-64 비트 )	(1-8 비트 )	(4-8 비트 )	(6 비트 )
반복 간격(선택적)	반복 카운트(선택적)	채널 상태	채널 ID	TCODE (111001)

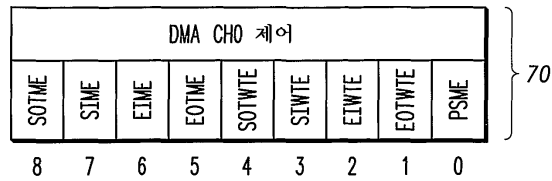
도면7

(8-64 비트 )	(8-64 비트 )	(1-8 비트 )	(4-8 비트 )	(6 비트 )
전송 간격(선택적)	전송 카운트(선택적)	채널 상태	채널 ID	TCODE (111010)

도면8

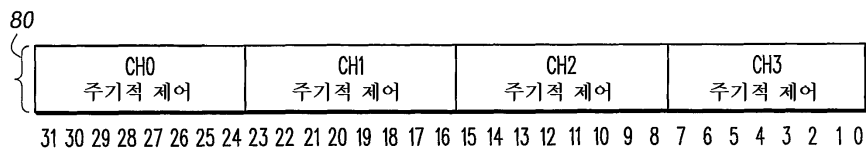
(8-64 비트 )	(1-8 비트 )	(4-8 비트 )	(6 비트 )
현재 전송 카운트(선택적)	채널 상태	채널 ID	TCODE (111011)

도면9



- SOTME - 전송 메시지 시작 인에이블(CHx)  
 0 = 전송 메시징 시작이 디스에이블됨  
 1 = 전송 메시징 시작이 인에이블됨
- SIME - 반복 메시지 시작 인에이블(CHx)  
 0 = 반복 메시징 시작이 디스에이블됨  
 1 = 반복 메시징 시작이 인에이블됨
- EIME - 반복 메시지 종료 인에이블(CHx)  
 0 = 반복 메시징 종료가 디스에이블됨  
 1 = 반복 메시징 종료가 인에이블됨
- EOTME - 전송 메시지 종료 인에이블(CHx)  
 0 = 전송 메시징 종료가 디스에이블됨  
 1 = 전송 메시징 종료가 인에이블됨
- SOTWTE - 전송 와치포인트 트리거링 시작 인에이블(CHx)  
 0 = 전송 와치포인트 트리거링의 시작이 디스에이블됨  
 1 = 전송 와치포인트 트리거링의 시작이 인에이블됨
- SIWTE - 반복 와치포인트 트리거링 시작 인에이블(CHx)  
 0 = 반복 와치포인트 트리거링의 시작이 디스에이블됨  
 1 = 반복 와치포인트 트리거링의 시작이 인에이블됨
- EIWTE - 반복 와치포인트 트리거링 종료 인에이블(CHx)  
 0 = 반복 와치포인트 트리거링의 종료가 디스에이블됨  
 1 = 반복 와치포인트 트리거링의 종료가 인에이블됨
- EOTWTE - 전송 와치포인트 트리거링 종료 인에이블(CHx)  
 0 = 전송 와치포인트 트리거링의 종료가 디스에이블됨  
 1 = 전송 와치포인트 트리거링의 종료가 인에이블됨
- PSME - 주기적 상태 메시지 인에이블(CHx)  
 0 = 주기적 상태 메시징이 디스에이블됨  
 1 = 주기적 상태 메시징이 인에이블됨

도면10



- CHO 주기적 제어  
 CHO 주기적 상태 메시지들의 비율을 제어한다
- CH1 주기적 제어  
 CH1 주기적 상태 메시지들의 비율을 제어한다
- CH2 주기적 제어  
 CH2 주기적 상태 메시지들의 비율을 제어한다
- CH3 주기적 제어  
 CH3 주기적 상태 메시지들의 비율을 제어한다