



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년10월04일
(11) 등록번호 10-1069120
(24) 등록일자 2011년09월26일

(51) Int. Cl.

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

(21) 출원번호 10-2006-7011044

(22) 출원일자(국제출원일자) 2004년11월04일

심사청구일자 2009년11월04일

(85) 번역문제출일자 2006년06월05일

(65) 공개번호 10-2007-0001895

(43) 공개일자 2007년01월04일

(86) 국제출원번호 PCT/US2004/037060

(87) 국제공개번호 WO 2005/062181

국제공개일자 2005년07월07일

(30) 우선권주장

10/728,398 2003년12월05일 미국(US)

(56) 선행기술조사문현

US6966015 B2

US6125368 A

US6477617 A

US5471631 A

전체 청구항 수 : 총 5 항

(73) 특허권자

프리스케일 세미컨덕터, 인크.

미국 텍사스 오스틴 월리엄 캐논 드라이브 웨스트
6501

(72) 발명자

모이어, 윌리엄 씨.

미국 텍사스 78620 드리핑 스프링스 피어 브랜치
로드 1005

콜린스 리차드, 지.

미국 텍사스 78732 오스틴, 컨트리 레이크 드라이
브 13405

(뒷면에 계속)

(74) 대리인

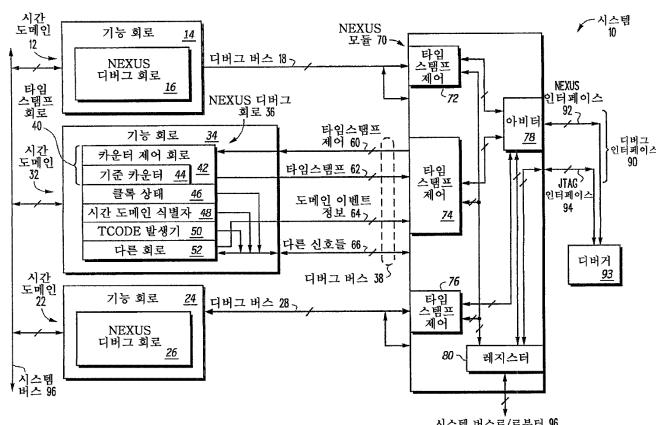
정상구, 장훈

심사관 : 신상길

(54) 다수의 시간 도메인들을 갖는 시스템에서 이벤트들을 시간 순서화하기 위한 장치 및 방법

(57) 요 약

본원은 상이한 시간 도메인들(12, 22, 32)가 존재하는 시스템(10)의 각종 부분들에서 발생되는 시간 순서 이벤트들을 위한 시스템 및 방법에 관한 것이다. 타임스탬핑 회로(40)는 복수의 기능 회로들 또는 모듈들(14, 24, 34) 각각에 제공된다. 이 타임스탬핑 회로는 미리 결정된 이벤트가 발생될 때의 시점을 나타내는 메시지를 제공한다. 인터페이스 모듈(70)은 복수의 기능 회로들(14, 24, 34) 각각에 결합된다. 인터페이스 모듈은 제어 정보를 복수의 기능 회로들에 제공하여, 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 나타내고 메시지 포맷을 선택적으로 규정한다. 인터페이스 모듈은 공통 인터페이스 포트(90)에서 하나, 여러 또는 모든 시간 도메인들로부터 타임스탬핑 메시지를 제공한다.

대 표 도

(72) 발명자

피즈심몬스 마이클 디.

미국, 텍사스 78731, 오스틴 #723, 엔. 로프 360
7700

니어링, 제이슨 티.

미국 텍사스 78729 오스킨, 마림바 트레일 12611

특허청구의 범위

청구항 1

삭제

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

시스템에서 이벤트들을 시간 순서화(time ordering)하기 위한 방법에 있어서:

복수의 시간 도메인들 각각에 대응하는 제어 정보를 제공하는 단계로서, 상기 제어 정보는 상기 복수의 시간 도메인들 각각에 대한 타임스탬프 메시지가 생성되는 때를 지시하는, 상기 제어 정보 제공 단계;

타임스탬프 메시지의 생성을 요구하는 시간 도메인 이벤트가 상기 복수의 시간 도메인들 중 어느 하나에서 발생하는 때를 결정하는 단계; 및

상기 시간 도메인 이벤트가 발생했다는 결정에 응답하여 상기 복수의 시간 도메인들 중 미리 결정된 하나에 대응하는 타임스탬프 메시지를 생성하는 단계를 포함하는, 이벤트들을 시간 순서화하기 위한 방법.

청구항 7

제 6 항에 있어서,

상기 타임스탬프 메시지가 갖는 복수의 미리 결정된 포맷들 중 하나를 식별하는 포맷 식별자 필드를 상기 타임스탬프 메시지 내에 포함하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.

청구항 8

제 6 항에 있어서,

타임스탬프 메시지의 생성을 요구하는 시간 도메인 이벤트가 상기 복수의 시간 도메인들 중 미리 결정된 하나에서 발생하는 때를 특정하기 위하여 상기 제어 정보를 사용하는 단계; 및

그 후, 상기 시간 도메인 이벤트가 상기 복수의 시간 도메인들 중 상기 미리 결정된 하나에서 발생하는 때를 결정하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.

청구항 9

이벤트들을 시간 순서화하기 위한 시스템에 있어서:

복수의 기능 회로 모듈들로서, 각각의 기능 회로 모듈은 다른 시간 도메인을 나타내는 클록(clock)에 의해 클록킹(clocking)되고 타임스탬핑 회로(timestamping circuitry)를 가지며, 상기 타임스탬핑 회로는 미리 결정된 이벤트가 발생하는 때의 시점을 지시하는 메시지를 제공하는, 상기 복수의 기능 회로 모듈들; 및

상기 복수의 기능 회로 모듈들 각각에 결합되는 인터페이스 모듈로서, 상기 인터페이스 모듈은 상기 복수의 기능 회로 모듈들에 제어 정보를 제공하여, 상기 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 지

시하며, 상기 인터페이스 모듈은 상기 미리 결정된 이벤트가 제 1 시간 도메인을 포함하는 복수의 시간 도메인들 중 하나에서 발생할 때 상기 제 1 시간 도메인으로부터 적어도 하나의 타임스탬핑 메시지를 수신하는, 상기 인터페이스 모듈을 포함하는, 이벤트들을 시간 순서화하기 위한 시스템.

청구항 10

제 9 항에 있어서,

상기 타임스탬핑 회로는:

대응하는 기능 회로 모듈에서 절대 또는 상대 시간 중 어느 하나를 결정하는 카운터;

시간 도메인 식별자를 제공하는 시간 도메인 식별 회로; 및

상기 대응하는 기능 회로 모듈에서 클록의 하나 이상의 동작 특성들을 제공하는 클록 상태 회로를 더 포함하는, 이벤트들을 시간 순서화하기 위한 시스템.

명세서

기술 분야

[0001]

본 발명은 일반적으로 시간 순서화 이벤트들에 관한 것으로, 보다 상세하게는 다수의 시간 도메인들을 갖는 시스템에서의 시간 순서화 이벤트들에 관한 것이다.

배경 기술

[0002]

타임스탬핑은 예를 들어 시스템을 디버깅할 때 원하는 이벤트가 언제 시스템에서 발생되는지를 나타내는 것과 같은 시스템에서 사용될 수 있는 유용한 기술이다. 단일 집적 회로 상에 집적되는 시스템들을 포함하는 일부 시스템들은 복수의 시간 도메인들을 포함한다.

발명의 상세한 설명

[0003]

몇몇 애플리케이션들을 위해, 시간 도메인들을 서로 직접 상관시키고 시스템 외부의 시간 기준들과 상관시키기 위한 순쉬운 방법이 존재하지 않을 수 있다.

[0004]

본 발명은 유사한 소자들에 동일한 참조번호들이 병기된 첨부 도면들에 예로서 도시되었지만 이로 제한되지 않는다.

실시예

[0010]

당업자는 전체 도면에 도시된 요소들이 원래 크기대로 도시한 것이 아니라 간결하고 명료하게 나타내도록 도시되었다는 것을 인지할 것이다. 예를 들어, 전체 도면들 내의 일부 요소들의 치수들은 본 발명의 실시예들의 이해를 증진시키기 위하여 다른 요소들에 비해서 확대되어 도시될 수 있다.

[0011]

본원에 사용된 바와 같이, 용어 "버스"는 예를 들어, 데이터, 어드레스들, 제어 또는 상태와 같은 하나 이상의 다양한 정보 유형들을 전달하는데 사용될 수 있는 다수의 신호들 또는 도체들을 참조하기 위해 사용된다.

[0012]

도 1에는 일반적으로 시간 도메인(12), 시간 도메인(32) 및 시간 도메인(22)을 갖는 데이터 처리 시스템(10)이 예시되어 있다. 시간 도메인(12)은 기능 회로(14)를 표시하는 모듈을 포함한다. 기능 회로(14) 내에는 NEXUS 디버그 회로(16)가 포함되는데, 여기서 "NEXUS"TM은 집적 회로들을 디버깅 및/또는 에뮬레이팅, 및/또는 테스팅하기 위해 공개적으로 이용가능한 IEEE IOSTO 5001 표준을 참조한다. 유사하게, 시간 도메인(32)은 기능 회로(34) 및 NEXUS 디버그 회로(36)를 갖는다. 시간 도메인(22)은 기능 회로(24) 및 NEXUS 디버그 회로(26)를 갖는다. NEXUS 디버그 회로(36)의 한 가지 가능한 실시예가 상세하게 도시된다. NEXUS 디버그 회로(16) 및 NEXUS 디버그 회로(26)가 동일한 방식으로 또는 여러 가지 변형들로 구현될 수 있다는 것이 명백하다. 기능 회로(14)는 디버그 버스(18)에 의해 NEXUS 모듈(70)에 결합된다. NEXUS 디버그 회로(36)는 디버그 버스(38)에 의해 NEXUS 모듈(70)에 결합되고 NEXUS 디버그 회로(26)는 디버그 버스(28)에 의해 NEXUS 모듈(70)에 결합된다.

[0013]

NEXUS 디버그 회로(36)는 카운터 제어 회로(42) 및 기준 카운터(44)를 갖는 타임스탬프 회로(40)를 포함한다. NEXUS 디버그 회로(36)는 또한 클록 상태 회로(46), 시간 도메인 식별자(48), TCODE 발생기(50), 및 다른 회로

(52)를 포함한다. 디버그 버스(38)는 NEXUS 모듈(70)로부터 카운터 제어 회로(42)에 결합되는 타임스탬프 제어 버스(60), 카운터 제어 회로(42)로부터 NEXUS 모듈(70)에 결합되는 타임스탬프 신호들(62), 다른 회로(52)로부터 NEXUS 모듈(70)에 결합되는 도메인 이벤트 정보 버스(64), 및 다른 신호들(66)을 위한 양방향 버스에 의해 형성된다. 일 실시예에서, 다른 신호들(66)은 클록 상태(46), 시간 도메인 식별자(48), TCODE 발생기(50), 및 다른 회로(52)를 NEXUS 모듈(70)에 결합시킨다. 다른 신호들(66)은 다른 회로(52) 및 NEXUS 모듈(70) 간에 양방향으로 결합된다.

[0014] NEXUS 모듈(70)은 일반적으로 각 시간 도메인과 관련된 타임스탬프 제어 회로를 포함한다. 도시된 형태에서, 타임스탬프 제어(72)는 디버그 버스(18)에 결합된다. 타임스탬프 제어(74)는 본원에 기재된 디버그 기능에 관련되지 않는 NEXUS 모듈(70) 내의 다른 제어 회로(도시되지 않음)에 결합하는 다른 신호들(66) 버스를 제외하고, 디버그 버스(38)를 형성하는 각 버스에 결합된다. 타임스탬프 제어(76)는 디버그 버스(28)에 결합된다. 아비터(arbiter)(78)는 타임스탬프 제어(72), 타임스탬프 제어(74) 및 타임스탬프 제어(76) 각각에, 그리고 다수의 레지스터들(80)에 양방향으로 결합된다. 타임스탬프 제어(72, 74, 76)의 각각은 다수의 레지스터들(80)에 양방향으로 결합된다. 양방향의 NEXUS 인터페이스 포트(92)는 아비터(78)에 결합된다. 양방향의 JTAG 인터페이스(94)는 다수의 레지스터들(80)에 결합된다. 다수의 레지스터들(80)은 시스템 버스(96)에 그리고 그로부터 양방향으로 결합된다. 시스템 버스(96)는 또한 시간 도메인(12), 시간 도메인(32) 및 시간 도메인(22)의 각각에 양방향으로 결합된다.

[0015] 접적 회로 디버그, 에뮬레이션 및/또는 테스팅 목적들을 위하여 사용되는 공통 표준은 잘 알려진 JTAG(Joint Test Action Group) IEEE(Institute of Electrical and Electronic Engineers) 1194.1라는 점에 주의한다. 디버그 인터페이스(90)가 JTAG 인터페이스를 갖는 것으로서 도시되었지만, 시스템의 대안적인 실시예들은 임의의 소망 인터페이스를 사용할 수 있다. 예를 들어, 표준 JTAG 인터페이스 이외에도, 접적 회로들을 위하여 사용되는 광범위의 다양한 다른 디버그, 에뮬레이션 및/또는 테스트 인터페이스들이 존재한다. 유사하게, 대안적인 실시예들은 NEXUS 인터페이스(92) 대신에 광범위한 다른 디버그, 에뮬레이션 및/또는 테스트 인터페이스들을 사용할 수 있다.

[0016] 시스템(10)의 일 실시예에서, 디버거(93)는 시스템(10)을 위한 디버깅 프로세스의 적어도 일부분 동안 디버그 인터페이스(90)에 선택적으로 결합될 수 있다. 대안적인 실시예들은 디버깅 기능을 수행하기 위하여 외부 디버거(93)를 필요로 하지 않을 수 있다. 일 실시예에서, 시스템(10)은 단일 접적 회로 상에 구현될 수 있고 디버거(93)는 접적 회로의 외부에 구현될 수 있다. 몇몇 실시예들에 대해서, 디버거(93)는 시스템(10)의 정상 동작 동안 디버그 인터페이스(90)에 결합되지 않는다. 디버거(93)는 하드웨어 및 소프트웨어의 임의의 조합을 사용하여 구현될 수 있다.

[0017] 도 1의 시스템(10)의 동작이 지금부터 설명될 것이다. 도시된 일 실시예에서, 복수의 시간 도메인들(12, 22, 32)은 시간 도메인 내에서 이벤트가 발생될 때에 관한 시간-관련된 정보를 제공할 수 있다. 일 실시예에서, 시간 도메인들은 공통 전력 및/또는 시간 도메인을 갖는 시스템 I/O에서 로컬 영역들이다. 시간 도메인 내의 요소들은 설계에 의해 동기화된 채로 유지되지만, 시간 도메인들은 전형적으로 다른 시간 도메인들과 완전히 동기화된 채로 유지될 필요는 없다. 일 실시예에서, 시간 도메인들(12, 22, 32) 각각은 기능 회로(14, 24, 34)를 각각 포함하는데, 이들 기능 회로 각각은 독립 클록 또는 이 클록의 파생물들(derivatives)을 이용하여 기능한다. 기능 회로(14, 24 및 34)는 임의의 원하는 기능을 수행할 수 있다. 각 시간 도메인(12, 22, 32)은 또한 NEXUS 디버그 회로(16, 26 및 36) 각각을 포함한다는 점에 주의한다. 대안적인 실시예들은 NEXUS 디버그 회로가 아닌 다른 회로(16, 26, 36)를 사용할 수 있다.

[0018] 시스템(10)이 3개의 시간 도메인들, 즉 시간 도메인(12, 22 및 32)을 갖는 것으로 도시되었지만, 대안적인 실시예들은 임의의 수의 시간 도메인들을 가질 수 있다는 점에 주의한다. 복수의 시간 도메인들(12, 22, 32)을 가질 수 있는 시스템(10)의 한 가지 예는 시간 도메인(12)이 매우 높은 클록 레이트 또는 주파수로 동작하는 프로세서를 포함하며, 시간 도메인(22)이 전력을 보존하기 위하여 더 낮은 클록 레이트 또는 주파수로 동작하는 회로를 포함하고 시간 도메인(32)이 제3 클록 레이트 또는 주파수로 동작하도록 설계된 주변 모듈들을 포함하는 접적 회로이다. 대안적인 실시예들은 임의의 원하는 기준들을 사용하여 시간 도메인들의 경계들을 묘사할 수 있다.

[0019] 몇몇 실시예들에서, 상이한 시간 도메인들에서 회로는 독립적으로 그리고 다른 시간 도메인들의 클록 주파수들 또는 듀티 사이클들 간에 상대적인 제약들이 없이 동작한다. 게다가, 시간 도메인 내의 회로의 클록킹은 전체 시스템 상태(10)가 변화됨에 따라서 주파수 변화를 중단하거나 겪을 수 있고, 이는 임의의 다른 시간 도메인의

클록킹과 관계없이 발생될 수 있다. 시간 도메인들은 대안적인 실시예들에서 다르게 기능하고 규정될 수 있다. 일 실시예에서, 독립적인 클록 레이트들은 예를 들어 공통 클록의 약수들인 클록들, 개별적으로 발생되고 의도적으로 관련되지 않는 클록들 및 특정 시간 도메인에 대해 일시적으로 중지되는 공통 클록 소스와 같은 다양한 방식들 중 한 가지 이상의 방식들로 분리 및 구별되는 클록 레이트들을 포함한다.

[0020] 일 실시예에서, 독립적인 클록 레이트들은 몇몇 방식으로 개별적으로 제어가능한 클록 레이트들인데, 이로 인해 특정 관계를 갖도록 보장되지 않아 적어도 특정 조건들 하에서 독립적이라고 간주될 수 있다. 독립적인 클록 레이트들이 가능한 조건들의 서브셋에 대해서 서로와 관계되도록 할 수 있다는 점에 주의한다. 본원에 서술된 타임스탬프 메시징 기능이 독립적인 클록 레이트들을 갖는 복수의 시간 도메인들(예를 들어, 12, 22, 32)과 함께 사용될 때 매우 유용하지만, 대안적인 실시예들은 임의의, 일부 또는 전체 복수의 시간 도메인들이 독립적인 클록 레이트들을 갖는 것을 필요로 하지 않을 수 있다.

[0021] 몇몇 실시예들에서, 시스템(10) 내의 이벤트들의 시간적 순서를 적절하게 재구성하기 위하여, 개별적인 시간 도메인들(예를 들어, 12, 22, 32) 내의 기준 카운터들(예를 들어, 44) 간의 관계들이 결정되어야 한다. 저 전력 상태들을 통해서 개별적인 시간 도메인들로서 이들 관계들을 결정하면, 개별적인 기준 카운터들이 갱신되는 상대 레이트에 영향을 미치는 클록 주파수 변화 이벤트들 및 이외 다른 팩터들은 시스템(10)의 개별적인 시간 도메인들 내에서 발생하는 다른 타임스탬프된 디버그 이벤트들의 관계를 결정하는데 유용하다. 본원에 설명된 타임스탬프 메시징 기능은 타임스탬프 정보를 NEXUS 디버그 회로(16, 36, 및 26)에 의해 발생되는 바와 같은 다른 디버그 메시지 유형들에 적용하기 위하여 사용되는 분배된 기준 카운터들 간의 관계를 결정하는데 필요한 정보를 제공한다. 이들 다른 디버그 메시지 유형들은 NEXUS 표준에 잘 알려져 있고, 프로그램 추적, 데이터 추적 및 와치포인트 메시징과 같은 메시지들을 포함한다. 일 실시예에서, 특정 시간 도메인에 의해 발생되는 이들 메시지들의 각각은 특정 시간 도메인 내의 기준 카운터의 상태와 관련된 타임스탬프 값에 따라서 충분될 수 있다. 다수의 시간 도메인들에 걸쳐서 발생되는 이벤트들의 시간적 순서화의 적절한 추적 재구성을 위하여, 특정 디버그 메시지 내에 제공되는 기준 카운터 값들은 서로에 상관되어야 한다. 본원에 서술된 타임스탬프 메시징은 시스템(10) 내의 개별적인 기준 카운터들의 관계들을 결정하는 방식을 제공함으로써 전체 좌표 능력을 제공한다.

[0022] 시간 도메인들(12, 22 및 32) 중 하나 이상의 특정 이벤트의 발생시, 하나 이상의 기준 카운터들(예를 들어, 44)의 값들은 캡쳐링되고 타임스탬프를 통해서 디버그 인터페이스(90) 외부의 디버깅 회로(93)로 전송될 수 있다. 이들 값들을 제공함으로써, 다양한 분배된 기준 카운터들(예를 들어, 44)의 관계들은 유지될 수 있고, 이로 인해, 타임스탬프로 주석되는 다른 디버그 메시지 유형들의 적절한 시간적 순서화가 수행될 수 있다. 몇몇 실시예들에 대해서, 기준 카운터들(예를 들어, 44)은 고정된 관계를 갖지 않고 서로 독립적으로 동작하기 때문에, 하나 이상의 기준 카운터들이 상태를 변화시키는 레이트는 모든 다른 기준 카운터들과 독립적이다. 기준 카운터들(예를 들어, 44)은 특정 시간 도메인(12, 22, 32)이 저전력 상태로 입력되는 동안 일시적으로 중지될 수 있거나 또는 특정 시간 도메인의 다이나믹스들이 변화될 때 더욱 빠른 또는 더욱 느린 레이트로 충분되도록 변화될 수 있다. 특정 시간 도메인(예를 들어, 32) 내에서 기준 카운터(예를 들어 44)의 값 및 클록 상태에 대한 정보를 메시징 출력(messaging out)함으로써, 복수의 시간 도메인들(12, 22, 32)에 걸쳐서 카운터 값들의 순서화를 유지할 수 있다.

[0023] 일 실시예에서, 시스템(10)은 동적으로 가변하는 레이트들로 동작하는 다수의 독립적인 기준 카운터들(예를 들어, 44)에 의해 타임스탬프되는 시스템(10)에서 이벤트들의 순서화를 용이하게 하고 디버그 인터페이스(90) 외부의 도구(예를 들어, 디버거(93))에 의해 분배된 소스들의 값들의 동기화를 성취하게 하는 방식으로 "타임스탬프 메시징" 기능을 수행한다. 타임스탬프 메시징은 다수의 시변 클록 도메인들로 분배된 시스템에서 이벤트들의 적절한 시간 순서화를 허용하는 디버그 정보의 새로운 소스이다. 타임스탬프 메시지들은 절대 또는 상대적인 타이밍 정보 중 어느 하나를 사용할 수 있다. 일부 실시예들에서, 상대적인 타이밍 정보는 메시지 길이를 더 짧게 하고 및 메시지 대역폭을 감소시킨다. 마스터 타임스탬프 동기 메시지들은 또한 주기적으로 사용되어 시간 도메인 기준 카운터들(예를 들어, 44)의 일부 또는 전부의 값들을 캡쳐링한다. 이는 디버그 도구(예를 들어, 디버거(93))가 시간 도메인(12, 22, 32) 관계들을 동기화시켜, 개별적인 시간 도메인 타임스탬프된 디버그 메시지들이 제공된 시스템(10) 이벤트들의 실제 시퀀스를 정확하게 재구성하도록 한다.

[0024] 일 실시예에서, 시간 도메인 타임스탬프 동기 메시지는 다음 시간 도메인 이벤트들을 위하여 디버그 인터페이스(90)(타임스탬프 메시징이 인에이블된다면)를 통해서 메시징될 수 있다:

[0025] 시스템 리셋으로부터 빠져나온 후 또는 타겟 도메인 타임스탬프 메시징이 인에이블될 때마다 초기 타겟 도메인 타임스탬프 동기 메시지

- [0026] 기준 클록의 변화시
- [0027] 저 전력 상태로 들어갈 때
- [0028] 저전력 상태로부터 리턴될 때
- [0029] 디버그 모드로 들어갈 때
- [0030] 디버그 모드로부터 리턴될 때
- [0031] 카운터 오버런의 발생시
- [0032] 주기적인 카운터 만료시
- [0033] 시스템(10) 내에서 하나 이상의 사용자 선택가능한 이벤트들의 발생시
- [0034] 대안적인 실시예들은 상기 리스트된 것보다 더 많은, 더 적은, 또는 상이한 시간 도메인 이벤트들을 사용할 수 있다는 점에 주의한다.
- [0035] 일 실시예에서, 마스터 타임스탬프 동기 메시지는 (제공된 타임스탬프 메시징이 인에이블되면) 다음 시간 도메인 이벤트들을 위한 디버그 인터페이스(90)를 통해서 메시징될 수 있다:
- [0036] 시스템 리셋으로부터 빠져나온 후 또는 타겟 도메인 타임스탬프 메시징이 인에이블될 때마다 초기 타겟 도메인 타임스탬프 동기 메시지
- [0037] 시스템(10) 기준 클록의 변화시
- [0038] 임의의(또는 선택된 클록들) 타겟 도메인 클록들의 변화시
- [0039] 저 전력 상태로 들어갈 때
- [0040] 저전력 상태로부터 리턴될 때
- [0041] 디버그 모드로 들어갈 때
- [0042] 디버그 모드로부터 리턴될 때
- [0043] 특정 타겟 도메인 카운터 오버런의 발생시
- [0044] 주기적인 카운터 만료시
- [0045] 내부 또는 외부에서 발생된 시스템(10) 이벤트 시
- [0046] 시스템(10) 내에서 하나 이상의 사용자 선택가능한 이벤트들의 발생시
- [0047] 대안적인 실시예는 상기 리스트된 것보다 더 많은, 더 적은, 또는 상이한 시간 도메인 이벤트들을 사용할 수 있다는 점에 주의한다.
- [0048] 시스템(10)의 도시된 형태가 디버그 목적들을 위하여 타임스탬핑을 수행하지만, 대안적인 실시예들은 임의의 원하는 목적을 위한 타임스탬핑을 수행할 수 있다. 또한, 디버그 인터페이스(90)가 JTAG 인터페이스(94) 및 NEXUS 인터페이스(92)와 같은 표준 디버그 인터페이스 회로 및 프로토콜을 사용하는 것으로서 설명되었지만, 대안적인 실시예들은 디버그 인터페이스(90)를 위한 임의의 원하는 인터페이스, 표준 또는 비표준을 사용할 수 있다.
- [0049] 지금부터 도 5를 참조하면, 도 5는 시스템(10)의 일 실시예를 따라 디버그 이벤트들을 시간-순서화하기 위한 방법을 도시한다. 흐름(200)은 타원(202)에서 시작하여 디버그 레지스터들(80)(도 1 참조)이 타임스탬프 제어 정보로 프로그램되는 단계(204)로 진행한다. 시스템(10)의 대안적인 실시예들이 시간 도메인들에 직접 타임스탬프 제어 정보를 제공할 수 있다는 점에 주의한다. 예를 들어, 타임스탬프 제어 정보는 디버그 인터페이스(90), 시스템(10)(도시되지 않음) 외부로 가는 다른 상호접속들, 시스템 버스(96), 또는 시스템(10) 내부 또는 외부의 또 다른 소스(도시되지 않음)로부터 수신될 수 있다. 시스템(10)의 몇몇 실시예들에 대해서, 레지스터들(80)은 시스템 버스(96) 및 디버그 인터페이스(90)의 하나 또는 둘 다에 의해 판독 및/또는 기록될 수 있다. 시스템(10)의 대안적인 실시예들에서, 레지스터들(80)은 시스템(10) 내의 어느 곳에도 위치될 수 있다.
- [0050] 도 5의 흐름은 단계(204)로부터 타임스탬프 제어 정보를 NEXUS 디버그 회로(16, 26, 및 36) 중 하나 이상의 회로로 전송하는 단계(206)로 계속된다. 시스템(10)의 일 실시예에 대해서, 타임스탬프 제어 신호들(60)은 이 타임스탬프 제어 정보를 NEXUS 디버그 회로(36)에 제공하기 위해 사용된다. 시스템(10)의 일 실시예에 대해서, 타

임스탬프 제어 정보는 카운터 제어 회로(42)에 제공된다. 시스템(10)의 일 실시예에서, 타임스탬프 제어 정보는 기준 카운터(44)를 인에이블하기 위한 제어 정보 및 타임스탬프 신호들(62) 상에 제공되는 정보에 영향을 미치도록 사용될 수 있는 임의의 다른 제어 정보를 포함할 수 있다.

[0051] 도 5의 흐름은 단계(206)로부터 "임의의 시간 도메인(12, 22, 32)에서 발생되는 시간 도메인 이벤트를 갖는가?"라는 질의를 하는 결정 다이아몬드(208)로 계속된다. 시간 도메인 이벤트가 임의의 시간 도메인(12, 22, 32)에서 발생되지 않으면, 이 흐름은 결정 다이아몬드(208)에서 멈추고 계속해서 시간 도메인 이벤트가 임의의 시간 도메인(12, 22, 32)에서 발생되는지 여부를 검사한다. 시간 도메인 이벤트가 임의의 시간 도메인(12, 22, 32)에서 발생되면, 이 흐름은 하나 이상의 선택된 시간 도메인 이벤트들을 토대로 한 타임스탬프 값을 하나 이상의 시간 도메인들로부터 NEXUS 모듈(70)로 전송하는 단계(210)로 계속된다.

[0052] 시간 도메인 이벤트가 그것 내에서 발생하는 시간 도메인(12, 22, 32)은 반드시 타임스탬프 값을 제공하는 시간 도메인(12, 22, 32)일 필요는 없다는 점에 주의한다. 예를 들어, 시간 도메인(12)에서 발생하는 시간 도메인 이벤트는 시간 도메인(32)을 트리거하여 타임스탬프 신호들(62)을 통해 타임스탬프를 NEXUS 모듈(70)에 제공한다. 예로서, 이 기능의 한 가지 가능한 사용은 시간 도메인(12)에서 전력 다운 시간 도메인 이벤트를 허용하여 시간 도메인(32)이 타임스탬프 신호들(62)을 통해서 타임스탬프를 제공하도록 함으로써, 기능 회로(14)가 전력 다운될 때 기능 회로(34)가 무엇을 행하는지를 결정할 수 있도록 한다. 이 기능은 광범위한 다른 목적들을 위하여 사용될 수 있다. 한 가지 실시예는 타임스탬프 제어 회로들(72, 74, 및 76) 간에 공유되는 정보에 의해 이 기능을 구현한다. 대안적인 실시예들은 기능 회로(14, 24 및 34)를 위한 다른 방법들을 제공하여, 예를 들어 기능 모듈들(14, 24 및 34) 중 2개 이상의 모듈들 간에 결합되는 하나 이상의 신호들에 의해 정보를 공유하도록 한다.

[0053] 도 5의 흐름은 단계(210)로부터 NEXUS 타임스탬프 메시지가 NEXUS 인터페이스(90)에 의해 NEXUS 모듈(70)로부터 전송되는 단계(212)로 계속된다. 일 실시예에서, 이 NEXUS 타임스탬프 메시지는 도 4에 도시된 마스터 메시지(120) 포맷을 사용한다. 대안적인 실시예들은 임의의 메시지 포맷을 사용할 수 있다. 예를 들어, 하나의 대안적인 실시예는 시퀀스의 시작을 표시하는 NEXUS TCODE 및 시퀀스의 끝을 나타내는 NEXUS TCODE를 (예를 들어, 도 2 및 도 3에 도시된 포맷들 중 하나 이상을 사용하여) 순차적으로 시간 도메인 메시지들에 제공할 수 있다. 시스템(10)의 대안적인 실시예들이 타임스탬프 정보를 제공하기 위한 임의의 원하는 포맷을 사용할 수 있다는 점에 주의한다. 도 2 내지 도 4에 도시된 메시지 포맷들은 단지 설명을 위한 것이다. 대안적인 메시지 포맷들은 NEXUS에 무관하며 TCODES를 사용하지 않을 수 있다. 대안적인 메시지 포맷들은 디버그 기능과 관련되거나 관련되지 않을 수 있다.

[0054] 도 5의 흐름은 단계(212)로부터 질문 "계속 추적하는가?"가 행해지는 결정 다이아몬드(214)로 계속된다. 추적이 계속되면, 흐름은 결정 다이아몬드(208)로 리턴하여 계속해서 시간 도메인 이벤트가 임의의 시간 도메인(12, 22, 32)에서 발생하는지를 검사한다. 추적이 계속되지 않으면, 흐름은 상기 흐름이 종료되는 END 타원(216)으로 계속된다.

[0055] 이제 도 2 및 도 3을 참조하면, 도 2는 절대 시간 도메인 기준값을 사용하는 시간 도메인 메시지(100) 포맷의 일 실시예를 도시하지만, 도 3은 상대 시간 도메인 기준 값을 사용하는 시간 도메인 메시지(110) 포맷의 일 실시예를 도시한다. 일 실시예를 위해, 시간 도메인 메시지들(100 및 110)에 대한 포맷들은 필드들(102 및 112)을 제외하면 동일할 수 있다. 일 실시예에서, 절대 시간 도메인 기준 카운트 비트 필드(102)는 거의 현재 도메인 이벤트가 발생하는 시간에 기준 카운터(44)(도 1 참조)의 실제 값을 포함할 수 있다. 일 실시예에서, 상대 시간 도메인 기준 카운트 비트 필드(112)는 (1) 거의 현재 도메인 이벤트가 발생하는 시간에서의 기준 카운트(44)의 실제 값; 및 (2) 거의 이전 도메인 이벤트가 발생하는 시간에서의 기준 카운트(44)의 이전 값 간의 차이를 포함할 수 있다. 몇몇 실시예들의 경우에서, (예를 들어, 도 1의 타임스탬프 신호들(62)에서) 더 적은 대역폭 또는 더 적은 신호들이 절대 시간 도메인 기준 카운트에 비하여 상대 시간 도메인 기준 카운트를 전송하도록 요구될 수 있다는 점에 주의한다.

[0056] 여전히 도 2 및 3을 참조하면, 시간 도메인 메시지들(100, 110)은 클록 상태 비트 필드(104, 114)를 각각 포함한다. 일 실시예에서, 클록 상태 필드(104, 114)는 클록들이 인에이블되는지 여부, 어떤 클록 주파수가 사용되고 있는지에 관한 정보, 또는 대응하는 시간 도메인(32)에서 클록들의 상태에 관한 임의의 다른 정보를 포함할 수 있다. 시간 도메인 메시지 포맷들(100, 110)은 또한 시간 도메인 식별자 비트 필드(106, 116)를 각각 포함한다. 일 실시예에서, 시간 도메인 식별자(106, 116)는 어떤 시간 도메인(12, 22, 32)이 시간 도메인 메시지에서 정보를 제공하고 있는지에 관한 정보를 포함할 수 있다. 시간 도메인 메시지 포맷들(100, 110)은 또한 TCODE 비

트 필드(108, 118)를 각각 포함할 수 있다. 일 실시예에서, TCODE 비트 필드(108, 118)는 어느 NEXUS 메시지 유형이 시간 도메인 메시지(100 및 110)에 의해 사용되는 메시지 유형인지에 관한 정보를 포함할 수 있다.

[0057] 도 4를 참조하면, 마스터 메시지(120)는 도메인 이벤트에 응답하여 타임스탬프를 제공하는 각 시간 도메인(12, 22, 32)에 대한 시간 도메인 절대 카운트(122, 124, 126)를 포함한다. 따라서, 마스터 메시지(120)는 하나 이상의 시간 도메인 절대 카운트 비트 필드들(122, 124, 126)을 포함한다. 일 실시예의 경우에, 시간 도메인(32)이 (예를 들어, 타임스탬프 신호들(62)에 의하여) NEXUS 모듈(60)로 타임스탬프를 제공하지 않는다면, 비트 필드(126)는 유용한 카운트 값을 포함하지 않거나, 마스터 메시지(120) 내에 포함되지 않는다는 것을 주의하라. 유사하게, 시간 도메인(12)이 NEXUS 모듈(70)로 타임스탬프를 제공하지 않는다면, 비트 필드(122)는 유용한 카운터 값을 포함하지 않거나 또는 마스터 메시지(120) 내에 포함되지 않을 것이다. 마찬가지로, 시간 도메인(22)이 NEXUS 모듈(70)로 타임스탬프를 제공하지 않는다면, 비트 필드(124)는 유용한 카운트 값을 포함하지 않거나, 또는 마스터 메시지(120) 내에 포함되지 않을 것이다. 하나 이상의 시간 도메인들(12, 22, 32)이 NEXUS 모듈(70)로 절대 시간 도메인 기준 카운트보다는 오히려 상대 시간 도메인 기준 카운트(예를 들어, 112)를 제공하면, 바람직한 경우, 하나 이상의 상대 시간 도메인 기준 카운터 값을(예를 들어, 112)이 예를 들어, 타임스탬프 제어 회로(72, 74, 76)(도 1 참조)에 의하여 절대 시간 도메인 기준 카운터로 변환될 수 있다는 것을 주의하라. 대안적인 실시예에서, 마스터 메시지(120)에 타임스탬프를 제공하고 있는 모든 시간 도메인들(12, 22, 32)은 상대 시간 도메인 기준 카운트(예를 들어, 112) 대신에, 절대 시간 도메인 기준 카운트(예를 들어, 102)를 사용할 수 있다.

[0058] 여전히 도 4를 참조하면, 마스터 메시지(120)는 상태 비트 필드(128)를 포함한다. 시스템(10)의 일 실시예에서, 상태 필드(128)는 하나 이상의 클록 상태 필드들(104, 114)로부터의 정보 모두 또는 일부를 인코딩되거나 인코딩되지 않은 형태로 포함할 수 있다. 시스템(10)의 일 실시예에서, 시간 도메인 절대 카운트 필드들(122, 124, 126) 중 하나에 카운트 값을 제공하는 시간 도메인들(12, 22, 32) 중 하나 이상은 또한 상태 비트 필드(128)로 임의의 원하는 상태 정보를 제공할 수 있다. 시스템(10)의 일 실시예에서, 상태 비트 필드(128)는 클록 상태 비트 필드들(104, 114)에 포함되는 것보다 더 많은 상태 정보를 포함할 수 있다. 이 상태 정보는 하나 이상의 시간 도메인들(12, 22, 및 32) 내의 회로의 임의의 원하는 상태와 관련할 수 있다.

[0059] 마스터 메시지(120)는 또한 TCODE 비트 필드(130)를 포함한다. 일 실시예에서, TCODE 비트 필드(130)는 어떤 NEXUS 메시지 유형이 마스터 메시지(120)에 의해 사용되는 메시지 유형인지에 관한 정보를 포함할 수 있다.

[0060] 도 1을 참조하면, 도시된 실시예에서, NEXUS 디버그 회로(36) 및 디버그 버스(38)의 일 실시예가 시간 도메인(32)에 대해 상세히 설명된다. 시간 도메인(12)은 또한 시간 도메인(32)과 상이한 방식으로 또는 동일한 방식으로 구현될 수 있는 NEXUS 디버그 회로(16) 및 디버그 버스(18)를 갖는다. 마찬가지로, 시간 도메인(22)은 또한 시간 도메인(32)과 상이한 방식으로 또는 동일한 방식으로 구현될 수 있는 NEXUS 디버그 회로(26) 및 디버그 버스(28)를 갖는다.

[0061] 시간 도메인(32)에 대한 도시된 실시예의 동작이 이제 설명될 것이다. 시스템(10)의 일 실시예에서, 타임스탬프 제어 회로(74)는 타임스탬프 제어 신호들(60)에 의하여 타임스탬프 제어 정보를 카운터 제어 회로(42)에 제공한다. 그 후, 카운터 제어 회로(42)는 이 제어 정보를 사용하여, 기준 카운터(44)를 제어한다. 시스템(10)의 일 실시예에서, 다른 회로(52)는 도메인 이벤트 정보 신호들(64)에 의하여 모든 도메인 이벤트들을 타임스탬프 제어(74)에 제공한다. 대안적인 실시예들에서, 타임스탬프 제어 신호들(60)은 어느 도메인 이벤트들이 선택되어서 도메인 이벤트 정보(64)에 의해 타임스탬프 제어(74)로 제공되어야 하는지에 관한 정보를 다른 회로(52)에 제공할 수 있다.

[0062] 일단, 선택된 도메인 이벤트가 발생하였다는 것을 타임스탬프 제어(74)가 결정하면, 타임스탬프 제어(74)는 타임스탬프 제어 신호들(60)에 의하여 제어 정보를 카운터 제어 회로(42)에 제공하여, 기준 카운터(44)에서의 값이 캡처링되어 타임스탬프 신호들(62)에 의하여 타임스탬프 제어(74)에 제공되어야 한다는 것을 나타낸다. 시간 도메인(12 또는 22)에서의 도메인 이벤트가 또한 기준 카운터(44)에서 값의 캡처를 트리거하기 위하여 타임스탬프 제어(74)에 의해 사용될 수 있다는 것을 주의하라. 이것은 타임스탬프 제어(72, 74, 및 76)가 정보를 공유할 수 있기 때문에 가능하다. 대안적인 실시예들은 도메인 이벤트의 발생 및 기준 카운터에서의 값의 캡처링 간의 지연을 단축하고자 할 수 있다. 시스템(10)의 일 실시예는 타임스탬프 제어 회로(74)의 기능 중 일부를 NEXUS 디버그 회로(36)로 이동하여, 도메인 이벤트의 겹출이 NEXUS 디버그 회로(36)의 외부 회로의 어떤 관여도 없이(예를 들어, 타임스탬프 제어(74) 없이), 기준 카운터(44)의 값의 캡처를 트리거함으로써 성취될 수 있다.

[0063] 기준 카운터(44)의 값이 타임스탬프 신호들(62)에 의해 제공되고, 시간 도메인 메시지(100)(도 2 참조)에서 절

대 시간 도메인 기준 카운트(102)를 제공하도록 고려될 수 있다는 것을 주의하라. 기준 카운터(44)의 값은 시간 도메인 메시지(110)의 상대 시간 도메인 기준 카운트 포맷이 사용되는 경우(도 3 참조), 타임스탬프 신호들(62)에 의하여 상대 시간 도메인 기준 카운트를 제공하기 위하여 카운터 제어 회로(42)에 의해 적절하게 수정되어야 한다. 타임스탬프 회로(40)로부터 타임스탬프 신호들(62)로 제공된 타임스탬프 정보의 포맷은 임의의 방식으로 결정될 수 있다. 예를 들어, 레지스터들(80)은 이러한 포맷을 결정할 수 있다. 대안적인 실시예들은 고정된 포맷을 사용하거나, 또는 대안적으로 임의의 원하는 방식으로 포맷을 선택할 수 있다.

[0064] 대안적인 실시예들의 경우에, 타임스탬프 제어(74)에 의해 수행되는 기능의 모두 또는 일부는 도 1에 도시된 바와 같이 집중되거나, 대안적으로 시간 도메인들(12, 22, 32) 자체 내에서 분산될 수 있다. 도 1에 도시된 시스템(10)의 일 실시예에서, 시간 도메인 식별자 회로(48)는 다른 신호들(66)에 의해 시간 도메인 식별자(도 2의 비트 필드(106) 및 도 3의 비트 필드(116) 참조)를 제공하기 위하여 사용된다. 유사하게, TCODE 발생기(50)는 다른 신호들(66)에 의하여 TCODES(도 2의 비트 필드(108), 도 3의 비트 필드(118), 및 도 4의 비트 필드(130) 참조)를 제공하기 위하여 사용된다. 유사하게, 클록 상태(46)는 다른 신호들(66)에 의하여 클록 상태 정보(도 2의 비트 필드(104), 도 3의 비트 필드(114), 및 도 4의 비트 필드(128) 참조)를 제공하기 위하여 사용된다.

[0065] 시스템(10)의 일 실시예에서, 아비터(78)는 어떤 타임스탬프 제어 회로(72, 74, 또는 76)가 시스템(10) 외부로의 전송을 위해 디버그 인터페이스(90)에 정보를 제공하게 되는지를 조정하고 결정하는데 사용될 수 있다.

[0066] 시스템(10)의 일 실시예에서, 클록 상태 필드(104, 114)는 클록들이 인에이블되는지 여부, 어떤 클록 주파수가 사용되고 있는지에 관한 정보, 또는 대응하는 시간 도메인(32)에서 클록들의 상태에 관한 임의의 다른 정보를 포함할 수 있다. 시간 도메인 메시지 포맷들(100, 110)은 또한 시간 도메인 식별자 비트 필드(106, 116)를 각각 포함한다. 일 실시예에서, 시간 도메인 식별자(106, 116)는 어떤 시간 도메인(12, 22, 32)이 시간 도메인 메시지에서 정보를 제공하고 있는지에 관한 정보를 포함할 수 있다. 시간 도메인 메시지 포맷들(100, 110)은 또한 CODE 비트 필드(108, 118)를 포함한다. 일 실시예에서, TCODE 비트 필드(108, 118)는 어떤 NEXUS 메시지 유형이 시간 도메인 메시지(100 및 110)에 의해 사용되는 메시지 유형인지에 관한 정보를 포함할 수 있다.

[0067] 상기 실시예들이 특정 전도도 유형들 또는 전위들의 극성에 대하여 설명되었을지라도, 당업자들은 다양한 변화들이 구현될 수 있다는 것을 인식한다. 예를 들어, 임의의 유형의 기능적인 회로가 NEXUS 디버그 회로로 사용될 수 있다. 다양한 반도체 프로세스들이 구현될 수 있고, 전도도 유형들 및 전위들의 극성들이 반전될 수 있다. 시스템(10)은 단일 접적 회로 상에서 시스템 온 칩(SOC)으로 구현되거나, 인쇄 회로 기판 레벨 상에서 이산 소자들로 구현될 수 있다. 시간스탬핑 메시지들의 다양한 필드들의 물리적인 포지셔닝은 도시된 것과 상이한 순서들로 배열될 수 있다. 다양한 조정 방식이 아비터(78)를 구현하기 위하여 사용될 수 있다. 임의의 유형의 저장 디바이스가 레지스터들(80)을 구현하기 위하여 사용될 수 있다. 본원에 설명된 카운터들은 다양한 유형들의 알려진 하드웨어 카운터 회로들로 구현될 수 있고, 임의의 하나의 특정 유형의 카운터에 국한되지 않는다. 버스크기들은 다양한 가능한 비트 폭들 중 어느 하나로 구현될 수 있다. 다양한 변형들의 상기 설명은 단지 설명을 위한 것이다. 광범위한 다른 변형들이 또한 가능하다.

[0068] 하나의 형태에서, 시스템 내의 이벤트를 시간 순서화하기 위한 방법이 시스템에서 제공되었다. 복수의 시간 도메인들 각각에 대응하는 제어 정보가 제공된다. 제어 정보는 복수의 시간 도메인들 각각에 대한 타임스탬프 메시지가 발생될 때를 나타낸다. 타임스탬프 메시지의 발생을 필요로 하는 시간 도메인 이벤트가 복수의 시간 도메인들 중 어느 하나에서 발생하는 때에 대한 결정이 이루어진다. 타임스탬프 메시지는 시간 도메인 이벤트가 발생했는지에 대한 결정에 응답하여 복수의 시간 도메인들 중 미리 결정된 한 도메인에 대응하여 발생된다. 하나의 형태에서, 타임스탬프 메시지들을 출력하기 위하여 복수의 시간 도메인들 모두에 대하여 동일한 포트가 사용된다. 하나의 형태에서, 제어 정보에 응답하여, 시간 도메인 이벤트가 메시지 발생 시간 도메인에서 발생되는 때에 대한 절대적 카운트 값인 메시지 발생 시간 도메인 내의 시간 카운트가 타임스탬프 내에 포함된다. 또 다른 형태에서, 제어 정보에 응답하여, 시간 도메인이 메시지 발생 시간 도메인에서 발생했던 때에 대한 가장 최근에 발생한 이전 시간 도메인 이벤트로부터 측정된 상대적 카운트 값인 메시지 발생 시간 도메인 내의 시간 카운트가 타임스탬프 메시지 내에 포함된다. 또 다른 형태에서, 제어 정보에 응답하여, 시간 도메인 이벤트가 발생했던 때에 대응하는 복수의 시간 도메인들 모두에 대한 시간 카운트가 타임스탬프 메시지 내에 포함된다. 또 다른 형태에서, 타임스탬프 메시지가 갖는 다수의 미리 결정된 포맷들 중 하나를 식별하는 포맷 식별자 필드가 타임스탬프 메시지 내에 포함된다. 타임스탬프 메시지의 발생을 필요로 하는 시간 도메인 이벤트가 복수의 시간 도메인들 중 미리 결정된 한 도메인에서 발생하는 시간을 규정하기 위하여 제어 정보가 사용된다. 하나의 형태에서, 시간 도메인 이벤트가 복수의 시간 도메인들 중 미리 결정된 한 도메인에서 발생하였는지가 그 다음에 결정된다. 하나의 형태에서, 제어 정보는 저장 디바이스 내로 프로그램된다. 또 다른 형태에서, 복수의 시간 도메인

인들 중 적어도 하나에서 시간 도메인 이벤트를 생성하는 미리 결정된 동작 조건들을 식별하는 제어 정보에 응답하여 타임스탬프 메시지가 발생된다. 또 다른 형태에서, 상기 미리 결정된 동작 조건들은 사용자 프로그램 가능한 이벤트 및 프로그램 가능한 시스템 이벤트 중 적어도 하나인 것으로 식별된다. 또 다른 형태에서, 적어도 하나 이상의 프로그램 가능한 시스템 이벤트는 전력 동작 모드로의 진입(entrance) 및 상기 전력 동작 모드로부터의 탈출(exit), 클록 소스의 변화, 클록 주기성의 변화, 하드웨어 카운터 값의 미리 결정된 변화 또는 디버그 동작 모드로의 진입 및 상기 디버그 동작 모드로부터의 탈출 중 적어도 하나를 포함한다. 상기 시스템은 하나의 형태에서, 복수의 기능적인 회로 모듈로 이루어지며, 각각의 기능적인 회로 모듈은 상이한 시간 도메인을 나타내는 클록에 의해 클록킹되며 시간 스템핑 회로를 갖는다. 시간 스템핑 회로는 미리 결정된 이벤트가 발생하는 시점을 나타내는 메시지를 제공한다. 인터페이스 모듈은 복수의 기능적인 회로 모듈들 각각에 결합되며, 상기 인터페이스 모듈은 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 나타내기 위하여 복수의 기능적인 회로 모듈들에 제어 정보를 제공한다. 인터페이스 모듈은 미리 결정된 이벤트가 제 1 시간 도메인을 포함하는 복수의 시간 도메인들 중 하나에서 발생할 때, 제 1 시간 도메인으로부터 적어도 하나의 시간스템핑 메시지를 수신한다. 인터페이스 모듈은 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 결정하는 제어 정보를 프로그램 가능한 제어 정보로서 저장하는 저장 회로를 더 포함한다. 하나의 형태에서, 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건은 전력 동작 모드로의 진입 및 상기 전력 동작 모드로부터의 탈출, 클록 소스의 변화, 클록 주기성의 변화, 하드웨어 카운터 값의 미리 결정된 변화, 디버그 동작 모드로의 진입 및 상기 디버그 동작 모드로부터의 탈출, 및 적어도 하나의 사용자 프로그램 가능한 이벤트의 발생 중 적어도 하나를 더 포함한다. 또 다른 형태에서, 시간스템핑 회로는 대응하는 기능적인 회로 모듈에서의 절대 또는 상대 시간 중 하나를 결정하는 카운터, 시간 도메인 식별자를 제공하는 시간 도메인 식별 회로, 및 대응하는 기능적인 회로 모듈에서 클록의 하나 이상의 동작 특성들을 제공하는 클록 상태 회로를 더 포함한다. 또 다른 형태에서, 시간스템핑 회로는 대응하는 메시지에 포함되는 정보의 포맷을 식별하기 위하여 각 메시지 내에 포함될 코드를 발생시키는 회로를 더 포함한다. 인터페이스 모듈은 대응하는 시간스템핑 메시지에 포함되는 정보의 포맷을 식별하기 위하여 각 시간스템핑 메시지에 포함될 코드를 발생시키는 회로를 갖는 아비터를 더 포함한다. 하나의 구현예에서, 인터페이스 모듈의 공통 인터페이스 포트는 IEEE 1588(NEXUS) 컴플라이언스(compliance)를 충족시킨다. 하나의 형태에서, 복수의 기능적인 회로 모듈들 중 적어도 하나에 의해 제공된 메시지는 적어도 알려진 시작 값에 대한 절대 값인 시간 카운트 값, 기능적인 회로 모듈들 중 하나와 관련되는 클록 신호의 상태 정보, 및 시간스템핑 메시지와 관련되는 대응하는 시간 도메인을 나타내는 식별자를 포함하는 포맷을 갖는다. 또 다른 형태에서, 메시지는 시간스템핑 메시지의 포맷이 절대 값 시간 카운트 값을 갖는지를 식별하는 필드를 더 포함하는 포맷을 갖는다. 또 다른 형태에서, 복수의 기능적인 회로 모듈들 중 적어도 하나에 의해 제공된 메시지는 적어도 최종적으로 발생하는 미리 결정된 이벤트에 대한 상대 값인 시간 카운트 값, 기능적인 회로 모듈들 중 하나와 관련되는 클록 신호의 상태 정보, 및 시간스템핑 메시지와 관련되는 대응하는 시간 도메인을 나타내는 식별자를 포함하는 포맷을 갖는다. 부가적인 형태에서, 메시지는 시간스템핑 메시지의 포맷이 상대 값 시간 카운트 값을 갖는지를 식별하는 필드를 더 포함하는 포맷을 갖는다. 또 다른 형태에서, 시간스템핑 메시지는 미리 결정된 이벤트가 발생할 때, 기능적인 회로 모듈들 각각에 관련되는 미리 결정된 상태 정보 및 기능적인 회로 모듈 각각에 대응하는 시간 카운트 값을 나타내는 포맷을 갖는다. 도시된 바와 같이, 제어 정보는 프로그램 가능하고, 인터페이스 모듈은 제어 정보를 저장하는 적어도 하나의 레지스터를 갖는다.

[0069] 또 다른 형태에서, 동일한 접속 회로 상에 복수의 기능적인 회로 모듈들을 갖는 시스템이 제공된다. 각각의 기능적인 회로 모듈은 상이한 시간 도메인을 나타내는 클록에 의해 클록킹된다. 각각의 기능적인 모듈은 타임스탬프 메시지들을 제공하기 위하여 독립적인 클록 레이트들에서 동작하는 시간스템핑 회로를 갖는다. 타임스탬프 메시지들은 각각 미리 결정된 이벤트가 발생하는 시점을 나타낸다. 인터페이스 모듈은 복수의 기능적인 회로 모듈들 각각에 결합되며, 상기 인터페이스 모듈은 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 나타내기 위하여 복수의 기능적인 회로 모듈들에 제어 정보를 제공한다. 인터페이스 모듈은 미리 결정된 이벤트가 제 1 시간 도메인을 포함하는 복수의 시간 도메인들 중 하나에서 발생할 때 제 1 시간 도메인으로부터 적어도 하나의 시간스템핑 메시지를 수신한다. 또 다른 형태에서, 시스템 내의 복수의 시간 도메인에서 발생하는 이벤트들의 시간 순서화를 재구성하는 방법이 제공된다. 다수의 시간스템핑 메시지들은 순서화된 시간 시퀀스 및 순서화되지 않은 시간 시퀀스 중 하나로 수신된다. 다수의 시간 도메인들과 관련되고 독립적인 클록 레이트들로 동작하는 복수의 시간 도메인 카운터들의 상대적 카운트 값들이 추적된다. 디버그 정보는 시간 순서화된 시퀀스로 분류되며, 상기 디버그 정보는 복수의 시간 도메인들 중 하나로부터 제공된 타임스탬프와 관련된다. 디버그 정보는 디버그 메시지를 통하여 제공된다. 디버그 메시지들은 프로그램 추적 메시지, 데이터 추적 메시지 및 감시점 메시지 중 적어도 하나로서 구현된다. 다수의 타임스탬프 메시지들은 복수의 시간 도메인들 각각에 대응하

며, 복수의 시간 도메인들 각각에 대한 타임스탬프 메시지가 발생될 때를 나타내는 제어 정보를 제공함으로써 발생된다. 타임스탬프 메시지의 발생을 필요로 하는 시간 도메인 이벤트가 복수의 시간 도메인들 중 어느 하나에서 발생하는 때에 관한 결정이 행해진다. 타임스탬프 메시지는 시간 도메인 이벤트가 발생했는지에 대한 결정에 응답하여 복수의 시간 도메인들 중 미리 결정된 한 도메인에 대응해서 발생된다.

[0070] 본원에 도시된 요소들이 도시된 요소들을 결합시키는 도시되지 않은 중개 요소들을 가질 수 있다. 본원에서 사용된 바와 같은 용어 "결합된"은 반드시 직접적이지는 않고, 반드시 기계적이지는 않을지라도, 접합되거나 또는 링크되는 것으로서 규정된다.

[0071] 상기 명세서에서, 본 발명은 특정 실시예들에 대해 설명되었다. 그러나, 당업자들은 이하의 청구항에서 설명된 바와 같은 본 발명의 범위를 벗어나지 않고 다양한 변경들 및 변화들이 행해질 수 있다는 것을 이해한다. 따라서, 명세서 및 도면들은 제한적이라기보다는 오히려 예시적인 것으로 간주되어야 하며, 모든 이러한 변경들이 본 발명의 범위 내에 포함되게 된다.

[0072] 이점들, 장점들, 및 문제점들에 대한 해결책들이 특정 실시예들에 대하여 상술되었다. 그러나, 이점들, 장점들, 문제점들에 대한 해결책들, 임의의 이점, 장점, 또는 해결책이 발생하거나 명백해지도록 할 수 있는 임의의 요소(들)는 어느 하나 또는 모든 청구항들의 중요하고, 필요로 되거나 본질적인 특징 또는 요소로서 해석되지 않아야 한다. 본원에 사용된 바와 같은 용어 "포함한다", "포함하는", 또는 이의 임의의 다른 변형은 요소들의 리스트를 포함하는 프로세스, 방법, 조항, 또는 장치가 단지 이러한 요소들을 포함하는 것이 아니라, 이와 같은 프로세스, 방법, 조항, 또는 장치에 명백하게 목록화되거나 고유한 다른 요소들을 포함할 수 있도록 비-배제적인 포함을 커버하게 된다.

부가 텍스트

[0073] 1. 시스템에서 이벤트들을 시간 순서화하기 위한 방법에 있어서,

[0074] 복수의 시간 도메인들의 각각에 대응하는 제어 정보를 제공하는 단계로서, 상기 제어 정보는 상기 복수의 시간 도메인들의 각각을 위한 타임스탬프 메시지가 발생될 때를 표시하는, 상기 제공 단계;

[0075] 타임스탬프 메시지의 발생을 필요로 하는 시간 도메인 이벤트가 상기 복수의 시간 도메인들 중 임의의 하나의 시간 도메인에서 발생하는 때를 결정하는 단계; 및,

[0076] 상기 시간 도메인 이벤트가 발생하였음을 결정하는 것에 응답하여 상기 복수의 시간 도메인들 중 미리 결정된 도메인에 대응하는 타임스탬프 메시지를 발생시키는 단계를 포함하는 이벤트들을 시간 순서화하기 위한 방법.

[0077] 2. 제 1 항에 있어서, 상기 제어 정보에 응답하여, 상기 시간 도메인 이벤트가 상기 메시지 발생 시간 도메인에서 발생될 때의 절대적 카운트 값인 메시지 발생 시간 도메인에서 시간 카운트를 상기 타임스탬프 메시지 내에서 포함하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.

[0078] 3. 제 1 항에 있어서, 상기 제어 정보에 응답하여, 상기 시간 도메인이 상기 메시지 발생 시간 도메인에서 발생될 때의 가장 최근에 발생하는 이전 시간 도메인 이벤트로부터 측정된 상대적 카운트 값인 메시지 발생 시간 도메인에서 시간 카운트를 상기 타임스탬프 메시지 내에서 포함하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.

[0079] 4. 제 1 항에 있어서, 상기 제어 정보에 응답하여, 상기 시간 도메인 이벤트가 발생한 때에 대응하는 상기 복수의 시간 도메인들 모두에 대한 시간 카운트를 상기 타임스탬프 메시지 내에서 포함하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.

[0080] 5. 제 1 항에 있어서, 상기 타임스탬프 메시지가 갖는 다수의 미리 결정된 포맷들 중 한 포맷을 식별하는 포맷 식별자 필드들을 상기 타임스탬프 메시지 내에서 포함하는 단계를 더 포함하는 이벤트들을 시간 순서화하기 위한 방법.

[0081] 6. 제 1 항에 있어서,

[0082] 타임스탬프 메시지의 발생을 필요로 하는 시간 도메인 이벤트가 상기 복수의 시간 도메인들 중 미리 결정된 도메인에서 발생하는 때를 규정하기 위하여 상기 제어 정보를 사용하는 단계; 및,

[0083] 그 후, 상기 시간 도메인 이벤트가 상기 복수의 시간 도메인들 중 미리 결정된 도메인에서 발생하는 때를 결정하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.

- [0085] 7. 제 1 항에 있어서, 상기 제어 정보를 저장 디바이스로 프로그램하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.
- [0086] 8. 제 1 항에 있어서, 상기 복수의 시간 도메인들 중 적어도 하나에서 시간 도메인 이벤트를 생성하는 미리 결정된 동작 조건들을 식별하는 상기 제어 정보에 응답하여 상기 타임스탬프 메시지를 발생시키는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.
- [0087] 9. 제 8 항에 있어서, 상기 미리 결정된 동작 조건들을 사용자 프로그램가능한 이벤트 및 프로그램가능한 시스템 이벤트 중 적어도 하나가 되도록 식별하는 단계를 더 포함하는, 이벤트들을 시간 순서화하기 위한 방법.
- [0088] 10. 제 9 에 있어서, 상기 적어도 하나의 프로그램가능한 시스템 이벤트는 전력 동작 모드로 진입 또는 탈출, 클록 소스의 변화, 클록 주기성의 변화, 하드웨어 카운터 값의 미리 결정된 변화, 디버그 동작 모드로 진입 또는 탈출 중 적어도 하나를 포함하는 이벤트들을 시간 순서화하기 위한 방법.
- [0089] 11. 이벤트들을 시간 순서화하기 위한 시스템으로서,
- [0090] 복수의 기능 회로 모듈들로서, 각 기능 회로 모듈은 상이한 시간 도메인을 표시하는 클록에 의해 클록킹되고 타임스탬핑 회로를 가지며, 상기 타임스탬핑 회로는 미리 결정된 이벤트가 발생될 때의 시점을 나타내는 메시지를 제공하는, 복수의 기능 회로 모듈들: 및,
- [0091] 상기 복수의 기능 회로 모듈들 각각에 결합되는 인터페이스 모듈로서, 상기 인터페이스 모듈은 상기 복수의 기능 회로 모듈에 제어 정보를 제공하여 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 나타내며, 상기 인터페이스 모듈은 상기 미리 결정된 이벤트가 상기 제1 시간 도메인을 포함하는 복수의 시간 도메인들 중 하나에서 발생될 때 제1 시간 도메인으로부터 적어도 하나의 타임스탬핑 메시지를 수신하는, 인터페이스 모듈을 포함하는 이벤트들을 시간 순서화하기 위한 시스템.
- [0092] 12. 제 11 에 있어서, 상기 인터페이스 모듈은:
- [0093] 상기 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 결정하는 프로그램가능 제어 정보로서 상기 제어 정보를 저장하는 저장 회로를 더 포함하는 이벤트들을 시간 순서화하기 위한 시스템.
- [0094] 13. 제 12 에 있어서, 상기 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건은 전력 동작 모드로 진입 또는 탈출, 클록 소스의 변화, 클록 주기성의 변화, 하드웨어 카운터 값의 미리 결정된 변화, 디버그 동작 모드로 진입 또는 탈출 및 적어도 하나의 사용자 프로그램가능 이벤트의 발생 중 적어도 하나를 포함하는 이벤트들을 시간 순서화하기 위한 시스템.
- [0095] 14. 제 11 에 있어서, 상기 타임스탬핑 회로는:
- [0096] 대응하는 기능 회로 모듈에서 절대 또는 상대 시간 중 어느 하나를 결정하는 카운터;
- [0097] 시간 도메인 식별자를 제공하는 시간 도메인 식별 회로; 및,
- [0098] 상기 대응하는 기능 회로 모듈에서 클록의 하나 이상의 동작 특성들을 제공하는 클록 상태 회로를 더 포함하는 이벤트들을 시간 순서화하기 위한 시스템.
- [0099] 15. 제 14 항에 있어서, 상기 타임스탬핑 회로는 대응하는 메시지에 포함되는 정보 포맷을 식별하기 위하여 각 메시지에 포함될 코드를 발생시키기 위한 회로를 더 포함하는, 이벤트들을 시간 순서화하기 위한 시스템.
- [0100] 16. 제 14 항에 있어서, 상기 인터페이스 모듈은 대응하는 타임스탬핑 메시지에 포함되는 정보의 포맷을 식별하기 위하여 각 타임스탬핑 메시지에 포함될 코드를 발생시키는 회로를 갖는 아비터를 더 포함하는 이벤트들을 시간 순서화하기 위한 시스템.
- [0101] 17. 제 11 항에 있어서, 상기 복수의 기능 회로 모듈들 중 적어도 한 모듈에 의해 제공되는 메시지는 적어도, 알려진 시작값을 참조하는 절대값인 시간 카운트 값, 상기 기능 회로 모듈들 중 하나의 모듈과 관련된 클록 신호의 상태 정보 및 상기 타임스탬핑 메시지와 관련된 대응하는 시간 도메인을 나타내는 식별자를 포함하는 포맷을 갖는, 이벤트들을 시간 순서화하기 위한 시스템.
- [0102] 18. 제 17 항에 있어서, 상기 메시지는 타임스탬핑 메시지의 포맷이 절대값 시간 카운트 값을 갖는지를 식별하는 필드를 더 포함하는 포맷을 갖는, 이벤트들을 시간 순서화하기 위한 시스템.
- [0103] 19. 제 11 항에 있어서, 상기 복수의 기능 회로 모듈들 중 적어도 하나의 모듈에 의해 제공되는 메시지는 적어

도, 마지막 발생하는 미리 결정된 이벤트를 기준으로 하는 상대값인 시간 카운트 값, 상기 기능 회로 모듈들 중 하나의 모듈과 관련된 클록 신호의 상태 정보 및 상기 타임스탬핑 메시지와 관련된 대응하는 시간 도메인을 나타내는 식별자를 포함하는 포맷을 갖는, 이벤트들을 시간 순서화하기 위한 시스템.

[0104] 20. 제 19 항에 있어서, 상기 메시지는 상기 타임스탬핑 메시지의 포맷이 상대 값 시간 카운트 값을 갖는지를 식별하는 필드를 더 포함하는 포맷을 갖는, 이벤트들을 시간 순서화하기 위한 시스템.

[0105] 21. 제 11 항에 있어서, 상기 타임스탬핑 메시지는 상기 기능 회로 모듈들의 각각에 대응하는 시간 카운트 값 및 상기 미리 결정된 이벤트가 발생될 때 상기 기능 회로 모듈들의 각각과 관련된 미리 결정된 상태 정보를 포함하는 포맷을 갖는, 이벤트들을 시간 순서화하기 위한 시스템.

[0106] 22. 제 11 항에 있어서, 상기 제어 정보는 프로그램가능한 이벤트들을 시간 순서화하기 위한 시스템.

[0107] 23. 제 11 항에 있어서, 상기 인터페이스 모듈은 상기 제어 정보를 저장하기 위한 적어도 하나의 레지스터를 더 포함하는 포맷을 갖는, 이벤트들을 시간 순서화하기 위한 시스템.

[0108] 24. 제 11 항에 있어서, 상기 인터페이스 모듈은 공통 인터페이스 포트에서 모든 시간 도메인들로부터 타임스탬핑 메시지들을 제공하는 이벤트들을 시간 순서화하기 위한 시스템.

[0109] 25. 제 24 항에 있어서, 상기 인터페이스 모듈의 공통 인터페이스 포트는 IEEE ISTO 5001(NEXUS) 컴플라이언스에 부합하는 이벤트들을 시간 순서화하기 위한 시스템.

[0110] 26. 이벤트들을 시간 순서화하기 위한 시스템에 있어서,

[0111] 복수의 기능 회로 모듈 수단으로서, 각 수단은 상이한 시간 도메인을 나타내는 클록에 의해 클록킹되고 타임스탬핑 회로 수단을 가지며, 상기 타임스탬핑 회로 수단은 미리 결정된 이벤트가 발생될 때의 시점을 나타내는 메시지를 제공하는, 복수의 기능 회로 모듈 수단; 및,

[0112] 상기 복수의 기능 회로 모듈 수단 각각에 결합되는 인터페이스 모듈 수단으로서, 상기 인터페이스 모듈 수단은 제어 정보를 상기 복수의 기능 회로 모듈에 제공하여 상기 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 나타내며, 상기 인터페이스 모듈 수단은 상기 미리 결정된 이벤트가 제 1 시간 도메인을 포함하는 복수의 시간 도메인들 중 한 도메인에서 발생될 때 상기 제 1 시간 도메인으로부터 적어도 하나의 타임스탬핑 메시지를 수신하는, 인터페이스 모듈 수단을 포함하는 이벤트들을 시간 순서화하기 위한 시스템.

[0113] 27. 제 26 항에 있어서, 모든 시간 도메인들로부터 상기 타임스탬핑 메시지들은 공통 인터페이스 포트 수단에서 인터페이스 모듈 수단에 의해 제공되는 이벤트들을 시간 순서화하기 위한 시스템.

[0114] 28. 시스템에 있어서, 동일한 접속 회로 상의 복수의 기능 회로 모듈들로서, 각 기능 회로 모듈은 상이한 시간 도메인을 나타내는 클록에 의해 클록킹되며, 각 기능 모듈은 타임스탬프 메시지들을 제공하는 독립적인 클록 레이트들에서 동작하는 타임스탬핑 회로를 갖는, 복수의 기능 회로 모듈들을 포함하는 시스템.

[0115] 29. 제 28 항에 있어서, 상기 타임스탬프 메시지를 각각은 미리 결정된 이벤트가 발생될 때의 시점을 나타내는, 시스템.

[0116] 30. 제 29 항에 있어서,

[0117] 상기 복수의 기능 회로 모듈들의 각각에 결합되는 인터페이스 모듈로서, 상기 인터페이스 모듈은 제어 정보를 상기 복수의 기능 회로 모듈들에 제공하여 상기 미리 결정된 이벤트를 트리거하는 적어도 하나의 동작 조건을 나타내도록 하며, 상기 인터페이스 모듈은 상기 미리 결정된 이벤트가 제 1 시간 도메인을 포함하는 복수의 시간 도메인들 중 하나의 도메인에서 발생할 때 제 1 시간 도메인으로부터 적어도 하나의 타임스탬핑 메시지를 수신하는, 시스템.

[0118] 31. 시스템에서 복수의 시간 도메인에서 발생하는 이벤트들의 시간 순서를 재구성하는 방법으로서,

[0119] 순서화된 시간 시퀀스 및 순서화되지 않은 시간 시퀀스 중 하나에서 다수의 타임스탬핑 메시지들을 수신하는 단계;

[0120] 복수의 시간 도메인들과 관련되고 독립적인 클록 레이트들로 동작하는 복수의 시간 도메인의 상대적 카운트 값을 들을 추적하는 단계; 및

[0121] 시간 순서화된 시퀀스에서 디버그 정보를 분류하는 단계로서, 상기 디버그 정보는 상기 복수의 시간 도메인들

중 한 도메인으로부터 제공되는 타임스탬프와 관련되는, 상기 분류 단계를 포함하는, 재구성 방법.

[0122] 32. 제 31 항에 있어서, 디버그 메시지를 통해서 상기 디버그 정보를 제공하는 단계를 더 포함하는, 재구성 방법.

[0123] 33. 제 32 항에 있어서, 프로그램 추적 메시지, 데이터 추적 메시지, 및 와치포인트 메시지 중 적어도 하나로서 상기 디버그 메시지들을 구현하는 단계를 더 포함하는, 재구성 방법.

[0124] 34. 제 31 항에 있어서, 복수의 시간 도메인들의 각각을 위한 타임스탬프 메시지가 발생될 때를 나타내고, 복수의 시간 도메인들의 각각에 대응하는 제어 정보를 제공하고

[0125] 타임스탬프 메시지의 발생을 필요로 하는 시간 도메인 이벤트가 복수의 시간 도메인들 중 임의의 한 도메인에서 발생될 때를 결정하며; 및

[0126] 상기 시간 도메인 이벤트가 발생되었다는 결정에 응답하여 복수의 시간 도메인들 중 미리 결정된 도메인에 대응하는 타임스탬프 메시지를 발생시킴으로써, 상기 다수의 타임스탬프 메시지들을 발생시키는 단계를 더 포함하는, 재구성 방법.

도면의 간단한 설명

[0005] 도 1은 본 발명의 일 실시예를 따른 시스템의 블록도.

[0006] 도 2는 본 발명의 일 실시예를 따른 시간 도메인 메시지(100)의 블록도.

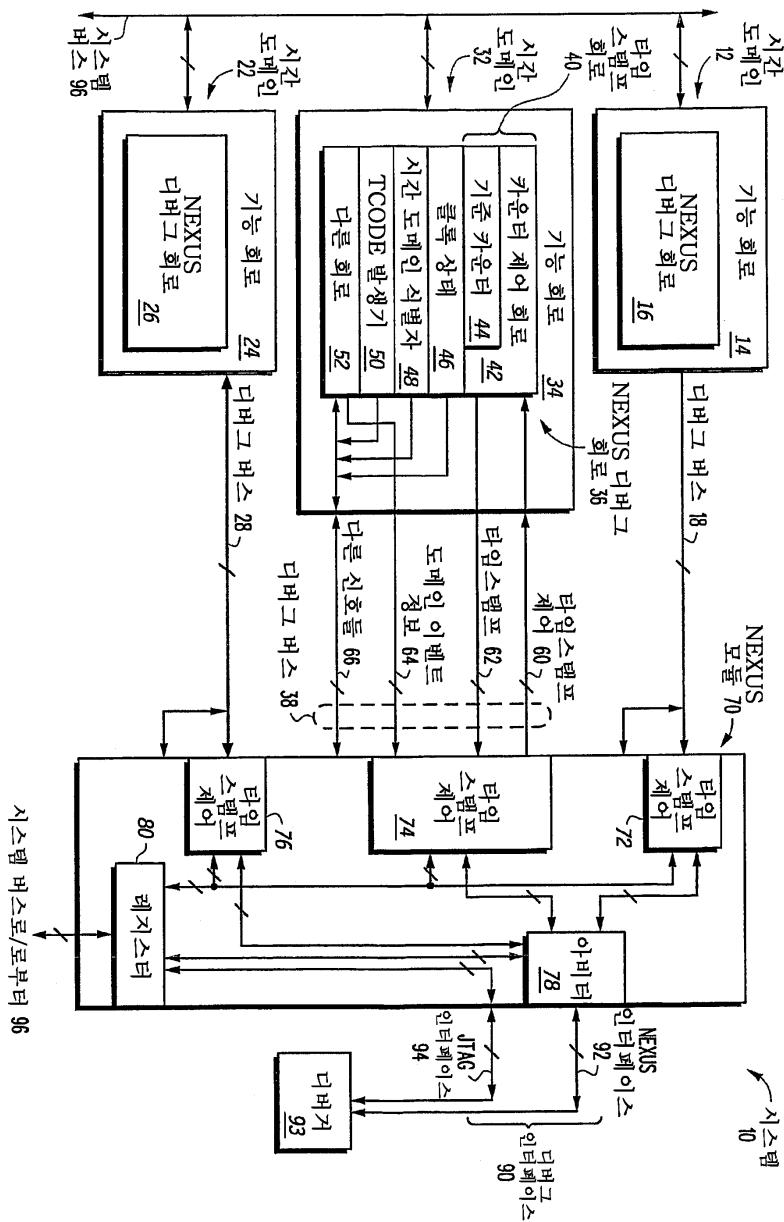
[0007] 도 3은 본 발명의 일 실시예를 따른 시간 도메인 메시지(110)의 블록도.

[0008] 도 4는 본 발명의 일 실시예를 따른 마스터 메시지(120)의 블록도.

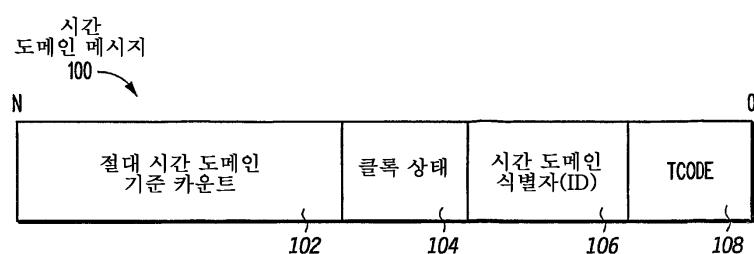
[0009] 도 5는 본 발명의 일 실시예를 따른 마스터 메시지(120)의 블록도.

도면

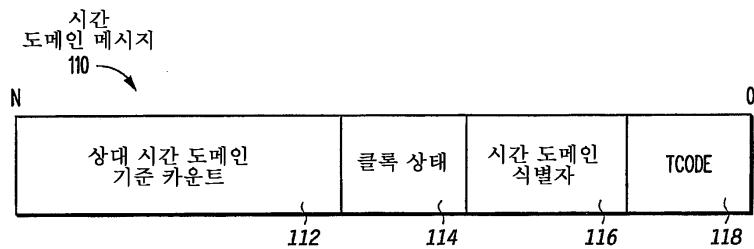
도면1



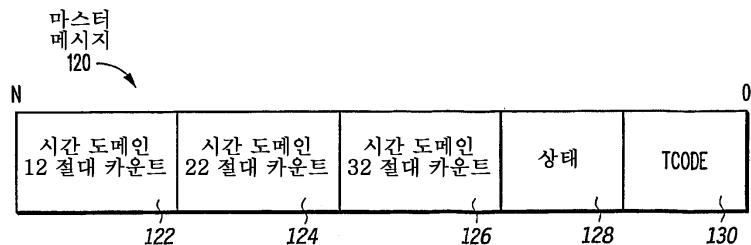
도면2



도면3



도면4



도면5

